Advances in Alternative Non-Adjacent Form Representations

Gildas Avoine, Jean Monnerat, and Thomas Peyrin

EPFL

Lausanne, Switzerland

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Preliminaries

Integer Representations

- Binary representation $n = \sum a_i 2^i$ where $a_i \in \{0, 1\}$
  e.g. $(13)_{10} = (001101)_2 = (1101)_2$.

  Unicity: The most significant bit is not 0.

- Ternary representation $n = \sum a_i 2^i$ where $a_i \in \{0, 1, \bar{1}\}$
  e.g. $(13)_{10} = (100\bar{1}\bar{1})_2 = (1\bar{1}000\bar{1}\bar{1})_2 = (10\bar{1}01)_2$.

  Unicity: For any two adjacent digits, at least one is zero and the most significant digit is not 0 [Reitwiesner, 1960].

Foundations

- $\{0, 1, \bar{1}\}$ can be generalized to $\{0, 1, x\}$. Improvement of [Muir and Stinson, 2003]
- The canonical representation of an integer using $\{0, 1, x\}$ is defined as in the case $\{0, 1, \bar{1}\}$: For any two adjacent digits, at least one is zero and the most significant digit is not 0.
- Such a representation is called the $\{0, 1, x\}$-Non-Adjacent Form (NAF), if it exists.
- Which sets $D = \{0, 1, x\}$ where $x \in \mathbb{Z}$ are such that every positive integer has a D-NAF?
- Such a set $\{0, 1, x\}$ is called a Non-Adjacent Digit Set (NADS).

Known NADS

- $\{0, 1, \bar{1}\}$
- $\{0, 1, 3\}$
- $\{0, 1, -5\}$, $\{0, 1, -13\}$, $\{0, 1, -17\}$, $\{0, 1, -25\}$, etc.

Known NADS

- $\{0, 1, \bar{1}\}$
- $\{0, 1, 3\} \rightarrow$ In the following, we will consider x negative
- $\{0, 1, -5\}$, $\{0, 1, -13\}$, $\{0, 1, -17\}$, $\{0, 1, -25\}$, etc.

Infinite Families

Example of infinite family of NADS [Muir and Stinson, 2003]:

- Let x be a negative integer such that $x \equiv 3 \pmod 4$ and $x = 7 - 2^t$, $t \geq 3$, $\{0, 1, x\}$ is a NADS iff t is odd
  e.g. -1, -25, -121, etc.

Example of infinite family of NON-NADS [Muir and Stinson, 2003]:

- Let x be a negative integer, if $\frac{3-x}{4} = 11 \cdot 2^i$ with $i \geq 0$, then $\{0, 1, x\}$ is a not a NADS (so called NON-NADS)
  e.g. -41, -85, -173, etc.

NADS

How to determine whether or not a set $D = \{0, 1, x\}$ is a NADS?

Definition

D is a NADS iff every positive integer has a D-NAF.

Theorem (Muir and Stinson)

If every positive integer in $[0, \lfloor -x/3 \rfloor]$ has a D-NAF, then D is a NADS.

Theorem (Muir and Stinson)

If every positive integer in $[0, \lfloor -x/3 \rfloor]$ and equal to 3 modulo 4 has a D-NAF, then D is a NADS.

NAF

How to determine whether or not an integer n has a D-NAF?

Theorem

A positive integer n has a D-NAF iff, $f_D(n)$ has a D-NAF, where

$$f_D(n) = \frac{n}{4} \quad \text{if } n \equiv 0 \pmod 4$$

$$f_D(n) = \frac{n-1}{4} \quad \text{if } n \equiv 1 \pmod 4$$

$$f_D(n) = \frac{n}{2} \quad \text{if } n \equiv 2 \pmod 4$$

$$f_D(n) = \frac{n-x}{4} \quad \text{if } n \equiv 3 \pmod 4$$

Graph of n

$$G_n : \quad n \longrightarrow f_D(n) \longrightarrow f_D^2(n) \longrightarrow f_D^3(n) \longrightarrow \ldots \longrightarrow 0$$

$$\begin{array}{ccccccc} & & & & & f_D^4(n) & \\ & & & & \swarrow & & \nwarrow \\ G_n : & n \longrightarrow & f_D(n) \longrightarrow & f_D^2(n) & & \longrightarrow & f_D^3(n) \end{array}$$

Either $f_D(n)$ reaches 0 or $f_D(n)$ loops because:

- $f_D(n) \leq \frac{-x}{3}$ when n is in the search domain
- 0 is the only fixpoint of f_D

Graph of n

$$G_n : \quad n \longrightarrow f_D(n) \longrightarrow f_D^2(n) \longrightarrow f_D^3(n) \longrightarrow \dots \longrightarrow 0$$

$$\begin{tikzcd}
 & & & & f_D^4(n) \arrow[dl] & \\
G_n : & n \arrow[r] & f_D(n) \arrow[r] & f_D^2(n) \arrow[rr] & & f_D^3(n) \arrow[ul]
\end{tikzcd}$$

Either $f_D(n)$ reaches 0 or $f_D(n)$ loops because:

- $f_D(n) \leq \frac{-x}{3}$ when n is in the search domain
- 0 is the only fixpoint of f_D

A positive integer n has a D-NAF iff G_n does not contain cycle.

Theoretical Results

Contents

- Search domain
- Generators of infinite families of NON-NADS
- Worst NON-NADS

Search Domain

Theorem

If every positive integer in $[0, \lfloor -x/3 \rfloor]$ has a D-NAF, then D is a NADS.

Search Domain

Theorem

If $3 \nmid x$ and every positive integer in $[0, \lfloor -x/3 \rfloor]$ has a D-NAF, then D is a NADS.

Search Domain

Theorem

If $3 \nmid x$ *and every positive integer in* $[0, \lfloor -x/6 \rfloor]$ *has a D-NAF, then D is a NADS.*

Search Domain

Theorem

If $3 \nmid x$ and every positive integer in $[0, \lfloor -x/6 \rfloor]$ has a D-NAF, then D is a NADS.

Theorem

If $3 \nmid x$ and $7 \nmid x$ and every positive integer in $[0, \lfloor -x/12 \rfloor] \cup [\lfloor -x/7 \rfloor, \lfloor -x/6 \rfloor]$ has a D-NAF, then D is a NADS.

Generators of NON-NADS

- n has a D-NAF if and only if G_n does not contain any cycle.
- If it exists n such that G_n contains a cycle, D is not a NADS.
- Instead of looking for NADS, we look for NON-NADS, obtaining (theoretically) the NADS by completion.
- We consider a cycle of a given form and deduce the x's for which it exists an n which lies in this cycle.

Generators of NON-NADS

- We choose the length t of the cycle and solve

$$f_D^t(n) = n.$$

- Define $f_0(n) = \frac{n}{4}$, $f_1(n) = \frac{n-1}{4}$, $f_2(n) = \frac{n}{2}$, and $f_3(n) = \frac{n-x}{4}$.
- We choose the form of the cycle and solve

$$f_D^t(n) = f_{i_t} \circ f_{i_{t-1}} \circ \dots f_{i_1}(n) = n,$$

for some chosen $i_k \in \{0, 1, 2, 3\}$ for $k = 1, 2 \dots, t$.

- Such a cycle is denoted as $i_1|i_2|\dots|i_t$.

2-cycles

- We have 3 possible cycles of length 2, namely $3|0$, $3|1$ and $3|2$.
- They lead to the equations $\frac{n-x}{16} = n$, $\frac{n-x-4}{16} = n$ and $\frac{n-x}{8} = n$.
- Since $n \equiv 3 \pmod 4$, we can set $n = 4k - 1$.

Theorem

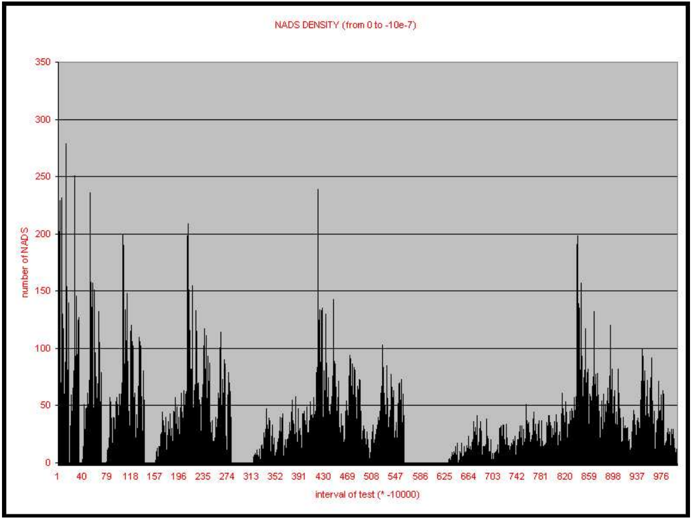
If $x = -60k + 15$, $x = -60k + 11$ or $x = -28k + 7$ with $k \in \mathbb{N}$, then $\{0, 1, x\}$ is a NON-NADS.

- We apply our method to a cycle of length t of the form $3|3|3|\ldots|3|0$.
- We solve $f_0 \circ f_3^{t-1}(n) = n$ for $t \geq 2$

Theorem

Let $t \geq 2$ and $k > 0$ be two integers and $x = -(4k-1)(2^{2t-1}-1)$. Then $\{0,1,x\}$ is a NON-NADS.

NADS Density



Definition

Let x be a negative integer such that $x \equiv 3 \pmod 4$. $\{0, 1, x\}$ is a worst NON-NADS if for all $n \leq -\frac{x}{3}$ with $n \equiv 3 \pmod 4$, n has not $\{0, 1, x\}$-NAF.

Theorem

Let x be a negative integer such that $x \equiv 3 \pmod 4$. $\{0, 1, x\}$ is a worst NON-NADS if and only if there exists $i \geq 2$ such that $(4m_i - 1) < -x < (3 \cdot 2^i)$, where

$$m_i := \begin{cases} 2 \cdot \frac{2^i - 1}{3} & \text{for } i \text{ even} \\ \\ \frac{2^{i+1} - 1}{3} & \text{for } i \text{ odd} \end{cases}$$

Algorithmic Aspects

Improvements

- Improvements of the search domain

Improvements

- Improvements of the search domain
- Generators of NON-NADS as a sieve
  (with an optimal cycle length t_{max})

Improvements

- Improvements of the search domain
- Generators of NON-NADS as a sieve (with an optimal cycle length t_{max})
- Worst NON-NADS

Improvements

- Improvements of the search domain
- Generators of NON-NADS as a sieve (with an optimal cycle length t_{max})
- Worst NON-NADS
- Memoization techniques

Memoization

- Memoization consists of remembering function calls and the corresponding outputs.
- The goal is to avoid to call a function several times with the same arguments.

Is-NADS?

Is-NADS?(x)

$N \leftarrow 3$

while $N \leq \frac{-x}{3}$

do
- $n \leftarrow N$
- $S \leftarrow \varnothing$
- **while** $n \neq 0$
- **do**
  - **if** $n \in S$
    - **then return** false
  - $S \leftarrow S \cup \{n\}$
  - $n \leftarrow f_D(n)$
- $N \leftarrow N + 4$

return true

Is-NADS?

```
Is-NADS?(x)
N ← 3
while N ≤ −x/3
  do { n ← N
       S ← ∅
       while n ≠ 0
         do { if n ∈ S
                then return false
              S ← S ∪ {n}
              n ← f_D(n)
            }
       N ← N + 4
     }
return true
```

Is-NADS?

Is-NADS?(x)

$N \leftarrow 3$

while $N \leq \frac{-x}{3}$

$$\textbf{do} \begin{cases} n \leftarrow N \\ S \leftarrow \varnothing \\ \textbf{while } n \neq 0 \\ \qquad \textbf{do} \begin{cases} \textbf{if } n \in S \\ \quad \textbf{then return } \text{false} \\ S \leftarrow S \cup \{n\} \\ n \leftarrow f_D(n) \end{cases} \\ N \leftarrow N + 4 \end{cases}$$

return true

Is-NADS?

Is-NADS?(x)

$N \leftarrow 3$

while $N \leq \frac{-x}{3}$

do $\begin{cases} n \leftarrow N \\ S \leftarrow \varnothing \\ \textbf{while } n \neq 0 \\ \quad \textbf{do } \begin{cases} \textbf{if } n \in S \\ \quad \textbf{then return } \text{false} \\ S \leftarrow S \cup \{n\} \\ n \leftarrow f_D(n) \end{cases} \\ N \leftarrow N + 4 \end{cases}$

return true

Evaluation of Is-NADS?(-25)

Example

Evaluation of Is-NADS?(-25)

G_3

$$f_D(3) \\ \downarrow \\ f_D(1) \\ \downarrow \\ 0$$

Example

Evaluation of Is-NADS?(-25)

$$
\begin{array}{cc}
G_3 & G_7 \\
 & \\
f_D(3) & f_D(7) \\
\downarrow & \downarrow \\
f_D(1) & f_D(2) \\
\downarrow & \downarrow \\
0 & f_D(1) \\
 & \downarrow \\
 & 0
\end{array}
$$

Example

Evaluation of Is-NADS?(-25)

$$
\begin{array}{ccc}
G_3 & G_7 & G_{11} \\
 & & \\
f_D(3) & f_D(7) & f_D(11) \\
\downarrow & \downarrow & \downarrow \\
f_D(1) & f_D(2) & f_D(3) \\
\downarrow & \downarrow & \downarrow \\
0 & f_D(1) & f_D(1) \\
 & \downarrow & \downarrow \\
 & 0 & 0
\end{array}
$$

Example

Evaluation of Is-NADS?(-25)

G_3	G_7	G_{11}
$f_D(3)$	$f_D(7)$	$f_D(11)$
$\downarrow$	$\downarrow$	$\downarrow$
$f_D(1)$	$f_D(2)$	$f_D(3)$
$\downarrow$	$\downarrow$	$\downarrow$
0	$f_D(1)$	$f_D(1)$
	$\downarrow$	$\downarrow$
	0	0

Example

Evaluation of Is-NADS?(-25)

$$
\begin{array}{ccc}
G_3 & G_7 & G_{11} \\
 & & \\
f_D(3) & f_D(7) & f_D(11) \\
\downarrow & \downarrow & \downarrow \\
f_D(1) & f_D(2) & f_D(3) \\
\downarrow & \downarrow & \downarrow \\
0 & f_D(1) & f_D(1) \\
 & \downarrow & \downarrow \\
 & 0 & 0
\end{array}
$$

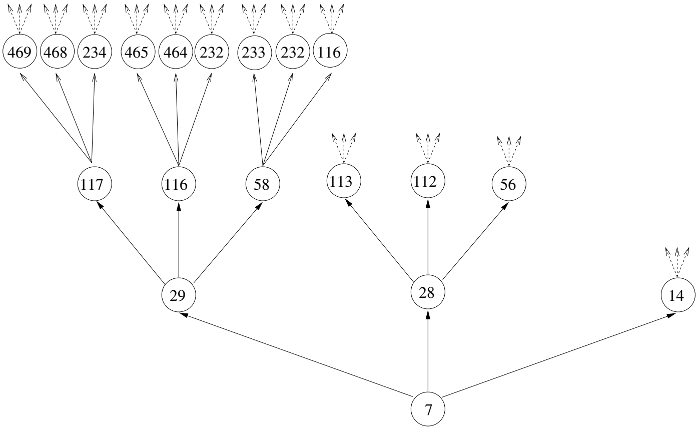
Memoization Technique

- Memoization is a straighforward technique (it can be applied because x is fixed at the begining of the evaluation of Is-NADS?(x)).
- A much more interesting idea is to use memoization over several executions of Is-NADS?.
- $f_D(n)$ depends on x
- Memoization only when $n \not\equiv 3 \pmod 4$.
- For that we define equivalence classes.

Memoization Technique

- Memoization is a straighforward technique (it can be applied because x is fixed at the begining of the evaluation of Is-NADS?(x)).
- A much more interesting idea is to use memoization over several executions of Is-NADS?.
- $f_D(n)$ depends on x but only when $n \equiv 3 \pmod 4$.
- Memoization only when $n \not\equiv 3 \pmod 4$.
- For that we define equivalence classes.

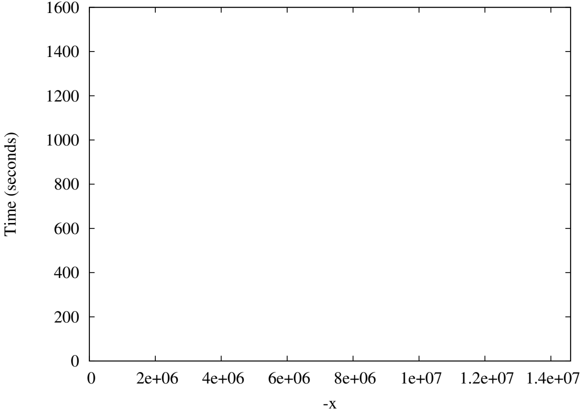
Equivalence Class of 7



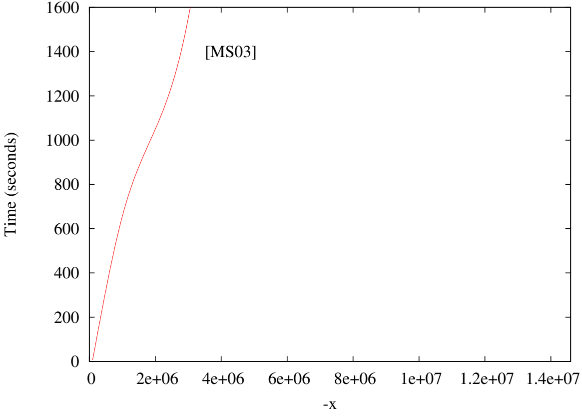
Improvements

- Improvement of the search domain
- Generators of NON-NADS as a sieve
- Worst NON-NADS
- Memoization techniques

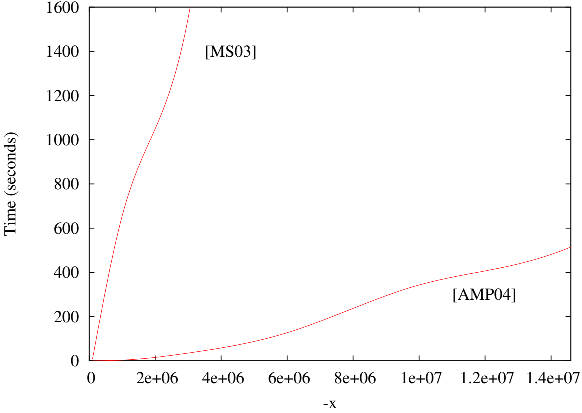
Results


1600
1400
1200
1000
800
600
400
200
0
Time (seconds)
0
2e+06
4e+06
6e+06
8e+06
1e+07
1.2e+07
1.4e+07
-x

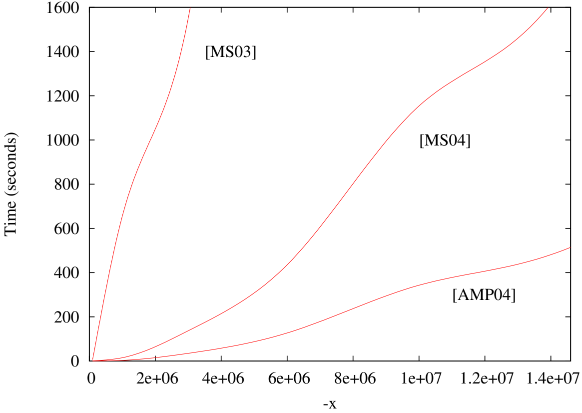
Results


1600
1400
1200
1000
800
600
400
200
0
Time (seconds)
[MS03]
0
2e+06
4e+06
6e+06
8e+06
1e+07
1.2e+07
1.4e+07
-x

Results


1600
1400
1200
1000
800
600
400
200
0
Time (seconds)
[MS03]
[AMP04]
0
2e+06
4e+06
6e+06
8e+06
1e+07
1.2e+07
1.4e+07
-x

Results


1600
1400
1200
1000
800
600
400
200
0
Time (seconds)
0
2e+06
4e+06
6e+06
8e+06
1e+07
1.2e+07
1.4e+07
-x
[MS03]
[MS04]
[AMP04]

Conclusion

Conclusion

- Reduction of the search domain.
- Generator of infinite families of NON-NADS.
- Improvement of the Muir and Stinson algorithm