Two-base Numeration Systems and Their Applications: Data Compression, Error Correction Codes, Cryptography

Prof. Anatoly V. Anisimov
ava@unicyb.kiev.ua

Kyiv, “Creative Computer” seminar
April 18, 2017
Numbers

- Numbers have many faces:
  - $\mathbb{N} = \{1, 2, 3, \ldots, 2014, \ldots, 2017, \ldots\}$
  - Unary: $1, 11, 111, \ldots$
  - Decimal: $1, 2, 3, \ldots, 2014, \ldots, 2017, \ldots$
  - Binary: $1, 10, 11, \ldots, 1111101111, \ldots$
  - Fibonacci:
    \[ x = \sum_{i=1}^{n} a_i F_i, \quad a_i \in \{0, 1\} \]
We get accustomed to traditional number representations: *Hindu-Arabic systems (the 5th century)*.

The impact of computer technologies leads to the interest of other ways of integer representations:

- *computer architectures*: binary, ternary, ...
- *coding*: Fibonacci-like systems, prefix representations, ...
- *cryptography, big data*: arithmetic of very large numbers, ...
One-base Numeration Systems

\[ U = u_0, u_1, \ldots ; \]
\[ 1 = u_0 < u_1 < \ldots ; \]

Representation of \( N \)

\[
\text{find } \max u_n \leq N \\
\text{for } i = n \text{ to } 0 \text{ by } -1 \\
\quad d_i \leftarrow \lfloor N/u_i \rfloor \\
\quad N \leftarrow N - d_i u_i \\
\text{end}
\]

\[ N = \sum_{i=0}^{n} d_i u_i, \quad \sum_{i=0}^{j} d_i u_i < u_{j+1} \quad (0 \leq j < n) \]

A. S. Fraenkel, The Use and Usefulness of Numeration Systems, Information and Computation 81, pp. 46-61, 1989
One-base Numeration Systems

They all are still one-base numeration systems!

Two-base Numeration Systems

\[ U = u_1, u_2, \ldots \quad V = v_1, v_2, \ldots \]

\[ x = au_n + bv_n \]

\((U, V)\)-transform

\(U\) is the main base, \(V\) is the auxiliary base
A) **Base:** \( U = u_1, u_2, \ldots; V = v_1, v_2, \ldots \). \( U \) is not upper bounded. All \( u_i, v_i \) are positive integers.

B) **Representation:** \( x = au_n + bv_n, \quad a > 0, b \geq 0 \) 
   (sometimes we consider \( b < 0 \))

C) **Recursion:** Representation \((x) = \text{Representation } (a)u_n + \text{Representation } (b)v_n + \text{Representation } (b)v_n\)

Apply B) recursively to \( a \) and \( b \).

How to notate this?
Linear Forms

- All integers \( N = \{1, 2, 3, \ldots\} \)
- Bases: \( U = u_1, u_2, u_3, \ldots, \) \( V = v_1, v_2, v_3, \ldots, \)

\[
x \perp y \iff GCD(x, y) = 1 \quad U \perp V \iff \forall i \quad u_i \perp v_i
\]

Definitions

- Linear form: \( x = au_n + bv_n \)
- Rank: \( n \)
- Positively defined (p. d.): \( a > 0, b \geq 0 \)
- Maximal rank: for a given \( x \) \( n \to \text{max} \)
- Canonical linear forms:
  
  - For a fixed rank \( n \) :  
    
    - Right canonical: \( a \) is maximal (\( b \) is minimal)
    
    - Left canonical: \( a \) is minimal (\( b \) is maximal)
  
  - Transformation into canonical forms:  
    \( x = (a - kv_n)u_n + (b + ku_n)v_n \)

- For a fixed rank \( n \) the canonical form (left or right) is unique.
Positively Defined Linear Forms of Maximal Rank

\[ x = au_n + bv_n, \quad a > 0, b \geq 0, \quad n \rightarrow \max \]

- **Motivation:**
  - We are trying to maximally impose “good” properties of \( U \) and \( V \) to arbitrary numbers.
How to obtain a variety of useful $U$ and $V$?

Given $U$, $V$ and $x$: how to obtain the positively defined linear form of maximal rank $n$, $x = au_n + bv_n$? $n \rightarrow \text{max}$

What are the properties of maximal linear forms?

(relations among $x$, $a$, $b$, $n$, $u_n$, $v_n$).

How to notate the recursive decomposition of a given number into linear forms?

Applications: What kind of problems in CS and IT can be solved using linear forms of two bases?
The World of Sequences

- \([1] = 1, 1, 1, \ldots\)
- \([a] = a, a, a, \ldots\)
- Powers of \(M\): \(1, M, M^2, \ldots\)
- Arithmetic progression: \(a_n = a_{n-1} + q\)
- Geometric progression: \(a_n = a_{n-1}q\)
The World of Sequences

- Fibonacci numbers: 0, 1, 1, 2, 3, 5, 8, 13, ..., 
  \[ F_{n+1} = F_{n-1} + F_n \]
- Lucas numbers: 2, 1, 3, 4, 7, 11, ..., 
  \[ L_{n+1} = L_{n-1} + L_n \]
- Pell numbers, approximations to \( \sqrt{2} = 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \ldots}} } \)
- Recurrent sequences, like 
  \[ P_n = aP_{n-1}^k + bP_{n-2}^m, \quad k \text{ and } m \text{ are fixed} \]
- SLOANE base of sequences (> 200000 sqs).
  [https://oeis.org](https://oeis.org)
How to Obtain the Variety of $U$ And $V$?

- $U$ is given, $V$ is shifted $U$:
  \[ v_n = u_{n+1}, \quad x = au_n + bu_{n+1} \]

- Continued fractions:
  \[
  \frac{P_n}{Q_n} = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \ldots}} \quad \rightarrow \varphi
  \]

- The most interesting case is quadratic irrationalities:
  \[
  \sqrt{k} = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \ldots}} \quad \frac{P_n}{Q_n} \rightarrow \sqrt{k}
  \]
Linear Fibonacci Forms

- U = 1, 1, 2, 3, 5, ...,  \( U = V \),  \( u_n = F_n \),  \( v_n = F_{n+1} \)

\[ x = aF_n + bF_{n+1} \]

197 = 23 \( F_6 \) + \( F_7 \) = 10 \( F_6 \) + 9 \( F_7 \)

15411 = 123 \( F_{11} \) + 31 \( F_{12} \)
Properties of Linear Forms

*Theorem 1.*

If \( U \perp V, v_n > u_n, \exists \lim \frac{v_{n+1}}{u_n} \), \( x = au_n + bv_n, n \rightarrow \max \)

then there exists a constant \( c \) such that \( a + b < c \sqrt{x} \)

(\( a \) and \( b \) have at least twice shorter bit length compared to \( x \)).
Integers \rightarrow \text{trees: } x = au_n + bv_n, \ n \rightarrow \text{max}

\[ T(x) = x \]

\[ u_n \quad v_n \]

\[ T(a) \quad T(b) \]

\[ \text{Height} = h \]

\[ \text{Theorem.} \]

If conditions of Theorem 1 hold then \( h = O(\log \log x) \)
Linear Fibonacci Trees of Integers

- \( x = aF_n + bF_{n+1} \)

- \( T(x) = F_n \)

- \( T(a) \)

- \( T(b) \)
Two–base Numeration System

- We fix $U = u_1, u_2, ...; \ V = v_1, v_2, ....$
- Definition:
  - a pair $(U, V)$ is a numeration system:
    - if for any $x \in \mathbb{N}$ there exists a positively defined representation of maximal rank
      $$x = au_n + bv_n$$
- Different forms of representations:
  $\text{Repr} \ (x) = \text{Repr} \ (\text{Repr} \ (a), \text{Repr} \ (b), n)$
Examples of Two-base Numeration Systems

The Ordinary Classic Representation by Powers of a Radix M

- $U = 1, M, M^2, \ldots; \quad V = [I] = 1, 1, 1, \ldots$
- Decomposition of integers into maximal right canonical linear forms $x = a_n M^n + x_1 \cdot 1, \quad n \text{ maximal}, \quad a_n \text{ is maximal possible}$

$$x_1 = a_m M^m + x_2 \cdot 1, \ldots, x = a_n M^n + a_{n-1} M^{n-1} + \ldots + a_0, \quad 0 \leq a_i < M.$$ 

$2014 = 2 \cdot 10^3 + 14^1 = 2 \cdot 10^3 + 1 \cdot 10 + 4^1 = 2 \cdot 10^3 + 1 \cdot 10 + 4 \cdot 10^0$
Examples of Two–base Numeration Systems

**Fibonacci Representation**

(Zeckendorf Representation)

\[ U = 1, 2, 3, 5, \ldots, F_n, \ldots; \quad V = [1] = 1, 1, 1, \ldots; \]

decomposition into left canonical maximal linear forms yields the representation:

\[ x = \sum_{i=1}^{k} a_i F_i, \quad a_i \in \{0, 1\}, \quad x = (a_k a_{k-1} \ldots a_1), \quad a_k = 1 \]

\[
2014 = F(17) + F(14) + F(9) + F(5) + F(2) = 1597 + 377 + 34 + 5 + 1 = (10010000100010010)_{F}\]

\[17 \ldots 14 \ldots 9 \ldots 5 \ldots 2 \ldots 1\]

There could be no two adjacent ones in Fibonacci representation. The Fibonacci code:

\[ Fib(x) = a_1 a_2 \ldots a_{k-1} 11\]
Examples of Two-base Numeration Systems

Additive representation as a sum of prime numbers:

\[ U = 2, 3, 5, 7, \ldots \text{ - prime numbers, } V = [1] = 1, 1, 1, \ldots \]

\[ x = p_{i_1} + p_{i_2} + \ldots + p_{i_k}, \quad p_{i_j} > p_{i_{j+1}}, \quad j = 1, 2, \ldots \]

\[ 2p_i > p_{i+1} > p_i \]

Continued fractions

\[ \frac{P_n}{Q_n} \rightarrow \alpha \quad U = P_1, P_2, \ldots \quad V = Q_1, Q_2, \ldots \]

\[ \alpha = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \ldots}} \]

\[ U = P_1, P_2, \ldots, \quad V = P_2, P_3, \ldots; \quad P_i = a_iP_{i-1} + P_{i-2}, \quad Q_i = a_iQ_{i-1} + Q_{i-2}; \]

Variants of linear decompositions

\[ x = aQ_i + bQ_{i+1}, \quad x = aP_i + bP_{i+1}, \quad x = aP_i + bQ_i \]
Classic well-known representation of numbers is a very particular (junior) “one dimensional” case of two-base representation.

Our traditional arithmetic and computer architectures stem from the very restricted “one dimensional” root.

Possibly two-base numeration systems is the way to new arithmetic, computational algorithms and computer architectures.
We consider the cases:

1. (2,3)-representation: $U = 1, 2, 2^2, \ldots, V = 3, 3, \ldots$

2. Linear recurrences: $U = P_1 = P_2 = 1$, $P_{i+1} = a_{i+1}P_i + P_{i-1}$, $i \geq 2$
   \[ V = P_3, P_4, \ldots, \quad u_n = P_n, \quad v_n = P_{n+1} \]
   \[ x = aP_n + bP_{n+1} \]

3. Continued fractions:
   \[ \alpha = [a_0, a_1, \ldots] \quad \alpha = \frac{P_n}{Q_n} \quad x = aQ_n - bP_n \]
2 is the main radix, 3 is the additional radix

\( U = 2, 2^2, 2^3, \ldots; \ V = 3, 3, 3, \ldots. \)

\( N_{2,3}: x \in N_{2,3} \iff \gcd(x, 2) = \gcd(x, 3) = 1. \)

For any \( x \in N_{2,3}, x = 2^n + 3^k x_1, x_1 \in N_{2,3}, \)

\[
n = \begin{cases} 
[\log_2 x] - 1 & \text{or} \\
[\log_2 x] 
\end{cases}
\]

\( x = 2^{n_0} 3^{k_0} x_1, x_1 = 2^{n_1} + 3^{k_1} x_2, \ldots, x_t = 2^{n_t} + 3^{k_t} , x_{t+1} = 1 \)

\( x_1 = 2^{n_1} + 3^{k_1} (2^{n_2} + 3^{k_2} (\ldots (2^{n_t} + 3^{k_t})\ldots)), x_i \in N_{2,3} \)
Examples of Two-base Numeration Systems

(2,3)-Representation

- $11 = 23 + 3$, $17 = 23 + 32$, $19 = 24 + 3$,
- $31 = 24 + 3(2+3)$

- $20092009 = 2^{24} + 3(2^{20} + 3(2^{13} + 3^2(2^{10} + 3^2(2^3 + 3^2))))$
- $20122012 = 2^2(2^{22} + 3^2(2^{16} + 3(2^{13} + 3(2^7 + 3(2^4 + 3^2(2 + 3)))))))$
(2,3)-Codes

\[ x = 2^{n_1} + 3^{k_1} (2^{n_2} + 3^{k_2} ...(2^{n_t} + 3^{k_t})...) \]

\[ x \leftrightarrow (n_1, k_1) (n_2, k_2) ...(n_t, k_t) \]

\[ \Delta \text{ - approach, } \Delta_i = n_i - n_{i+1} - k_i, \]

\[ x \leftrightarrow (\Delta_1, k_1)(\Delta_2, k_2) ...(\Delta_t, k_t) \]

\[ \Delta_i = n_i - \bar{n}_{i+1} - k_i, \quad \bar{n}_{i+1} = \lfloor \log_2 x_{i+1} \rfloor \]

Blocks: \((\Delta_i, k_i)\)

A block \((\Delta_i, k_i)\) is called:

- maximal if \( n_i = \lfloor \log_2 x_i \rfloor \),
- minimal if \( n_i = \lfloor \log_2 x_i \rfloor - 1 \)
(2,3)–Codes

Properties

\( \tau_{\text{max}}(x) = \) number of \textit{max} blocks
\( \tau_{\text{min}}(x) = \) number of \textit{min} blocks
\( \tau(x) = \) number of all blocks
\[
\sigma(x) = \sum_{i=1}^{t} k_i
\]

Properties:

\( \Delta_i \geq 0, \quad k_i (\log_2 3 - 1) - \log_2 3 < \Delta_i \)
\( \tau_{\text{max}}(x) + (2 - \log_2 3) \tau_{\text{min}}(x) + (\log_2 3) \sigma(x) \leq \log_2 x \)
\( \tau_{\text{max}}(x) < \frac{\log_2 x}{1 + \log_2 3} \)
\( P_r(k_i = s) = \frac{2}{3^s} \)
\( E \left[ \frac{\tau(x)}{\log_2 x} \right] \leq \frac{1}{1,5 + \log_2 3} \approx 0.32 \)
(2,3)-Codes

Properties

Average value of $\frac{\tau_{\max}(x)}{\tau(x)}$  

Average value of $\frac{\sigma(x)}{\tau(x)}$
(2,3)–Codes

Number x
(2,3) – representation
(delta, k) approach

There are some relations between delta and k (inequalities)

robust binary prefix error correction (2,3)–codes
Efficiency of the (2,3,3)–error correction code

Time complexity

Correction efficiency

seconds vs. number of errors

% corrected vs. number of errors
Some experiments with (2,3) – Compression

A

Texts → RAR-compression → Compressed Data

B

Texts → (2,3) - Compression → Compressed Data

C

Texts → (2,3) - Compression → RAR-compression → Compressed Data

B % - A % ≈ 7 % ≈ 8%;
A % - C % ≈ 1 % ≈ 2%;
B allows search in compressed data;
C is better than RAR.
## (2,3)-Codes. Data Compression. Results

<table>
<thead>
<tr>
<th>Text</th>
<th>Size(.txt)</th>
<th>Size(.rar)</th>
<th>Size(.out)</th>
<th>Size(.out,.rar)</th>
<th>% (rar/txt)</th>
<th>% (out/txt)</th>
<th>% (out.rar/txt)</th>
<th>%/(out.rar/txt) - %/(out/txt)</th>
<th>% (out.rar)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jane Eyre</td>
<td>166925</td>
<td>56458</td>
<td>69054</td>
<td>54268</td>
<td>33,822</td>
<td>41,368</td>
<td>32,51</td>
<td>1,311</td>
<td>96,121</td>
</tr>
<tr>
<td>The Citadel</td>
<td>212655</td>
<td>65626</td>
<td>82089</td>
<td>64434</td>
<td>30,860</td>
<td>38,601</td>
<td>30,299</td>
<td>0,56</td>
<td>98,183</td>
</tr>
<tr>
<td>RobinHood</td>
<td>19343</td>
<td>6909</td>
<td>7191</td>
<td>6010</td>
<td>35,718</td>
<td>37,176</td>
<td>31,07</td>
<td>4,647</td>
<td>86,987</td>
</tr>
<tr>
<td>(3.2)</td>
<td></td>
<td></td>
<td></td>
<td>16459</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mockingjay</td>
<td>397966</td>
<td>144012</td>
<td>6</td>
<td>133561</td>
<td>36,187</td>
<td>41,359</td>
<td>33,56</td>
<td>2,626</td>
<td>92,742</td>
</tr>
<tr>
<td>The coral island</td>
<td>122657</td>
<td>42543</td>
<td>50838</td>
<td>39839</td>
<td>34,684</td>
<td>41,447</td>
<td>32,48</td>
<td>2,204</td>
<td>93,644</td>
</tr>
<tr>
<td>Studies of tree</td>
<td>288312</td>
<td>87359</td>
<td>10897</td>
<td>83211</td>
<td>30,3</td>
<td>37,796</td>
<td>28,861</td>
<td>1,438</td>
<td>95,251</td>
</tr>
<tr>
<td>The war of the words</td>
<td>150476</td>
<td>49274</td>
<td>57607</td>
<td>45555</td>
<td>32,745</td>
<td>38,283</td>
<td>30,273</td>
<td>2,471</td>
<td>92,452</td>
</tr>
<tr>
<td>Charly and chocolate factory</td>
<td>157068</td>
<td>51539</td>
<td>63220</td>
<td>49937</td>
<td>32,813</td>
<td>40,25</td>
<td>31,793</td>
<td>1,019</td>
<td>96,891</td>
</tr>
<tr>
<td>East of Eden</td>
<td>164537</td>
<td>56519</td>
<td>68798</td>
<td>53953</td>
<td>34,35</td>
<td>41,813</td>
<td>32,790</td>
<td>1,55</td>
<td>95,459</td>
</tr>
<tr>
<td>(2) Catching Fire</td>
<td>549345</td>
<td>183903</td>
<td>23276</td>
<td></td>
<td>33,476</td>
<td>42,371</td>
<td>32,416</td>
<td>1,060</td>
<td>96,831</td>
</tr>
<tr>
<td>Death on the Nile</td>
<td>152823</td>
<td>49683</td>
<td>58867</td>
<td>46910</td>
<td>32,51</td>
<td>38,519</td>
<td>30,695</td>
<td>1,814</td>
<td>94,418</td>
</tr>
<tr>
<td>Twilight</td>
<td>770101</td>
<td>226920</td>
<td>31511</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td>Page 1</td>
<td>Page 2</td>
<td>Page 3</td>
<td>Page 4</td>
<td>Page 5</td>
<td>Page 6</td>
<td>Page 7</td>
<td>Page 8</td>
<td>Page 9</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>Harry Potter &amp; The chamber of secret</td>
<td>504396</td>
<td>170491</td>
<td>201096</td>
<td>161571</td>
<td>33,801</td>
<td>39,868</td>
<td>32,032</td>
<td>1,768</td>
<td>94,768</td>
</tr>
<tr>
<td>A child's history of English</td>
<td>944563</td>
<td>302768</td>
<td>364968</td>
<td>283356</td>
<td>32,053</td>
<td>38,638</td>
<td>29,998</td>
<td>2,055</td>
<td>93,588</td>
</tr>
<tr>
<td>(3.1) Mockingjay</td>
<td>165858</td>
<td>63237</td>
<td>67890</td>
<td>56351</td>
<td>38,127</td>
<td>40,932</td>
<td>33,975</td>
<td>4,151</td>
<td>89,110</td>
</tr>
<tr>
<td>Pride And Prejudice</td>
<td>228627</td>
<td>71059</td>
<td>86411</td>
<td>67199</td>
<td>31,08</td>
<td>37,795</td>
<td>29,392</td>
<td>1,688</td>
<td>94,567</td>
</tr>
<tr>
<td>(1) The Hunger Games</td>
<td>523003</td>
<td>175810</td>
<td>216627</td>
<td>167864</td>
<td>33,615</td>
<td>41,419</td>
<td>32,096</td>
<td>1,519</td>
<td>95,48</td>
</tr>
<tr>
<td>The History of London</td>
<td>422825</td>
<td>144740</td>
<td>164092</td>
<td>131214</td>
<td>34,231</td>
<td>38,808</td>
<td>69674</td>
<td>3,198</td>
<td>90,654</td>
</tr>
<tr>
<td>2001 A space oddisey</td>
<td>140711</td>
<td>47110</td>
<td>53883</td>
<td>43142</td>
<td>33,479</td>
<td>38,293</td>
<td>30,66</td>
<td>2,819</td>
<td>91,577</td>
</tr>
<tr>
<td>Benneton</td>
<td>146013</td>
<td>47833</td>
<td>52989</td>
<td>42546</td>
<td>32,759</td>
<td>36,290</td>
<td>29,138</td>
<td>3,62</td>
<td>88,946</td>
</tr>
<tr>
<td>Cry Freedom</td>
<td>154359</td>
<td>50470</td>
<td>60543</td>
<td>47950</td>
<td>32,696</td>
<td>39,222</td>
<td>31,063</td>
<td>1,632</td>
<td>95,006</td>
</tr>
</tbody>
</table>
DATA COMPRESSION. FRESH RESULTS

(2,3) –codes $\rightarrow$ $(\delta, k)$-codes

Multidelimeter codes $\rightarrow$ $(\delta, k)$-codes
Two-base Numeration Systems. Encoding Trees by Integers

\[ x = au_n + bv_n \]

- Integers $\mapsto$ binary ordered trees.
- The inverse problem: the task is to encode trees by integers.
Cantor’s pairing function:

\[(a, b) \leftrightarrow \left(\frac{(a + b)(a + b + 1)}{2}\right) + b\]

Encoding pairs by maximal linear forms (much easier):

\[(a, b) \leftrightarrow y = (a + b)P_n + bP_{n+1} , \ a + b < P_{n+1}\]

Decoding:

\[y = cP_n + dP_{n+1}\]

maximal left-canonical

\[b = d , \ a = c - d\]

Can be extended to \(m\)-tuples.
We can do:

**Integers** $\rightarrow$ linear trees by means of linear forms

Inverse problem: **binary trees** $\rightarrow$ **integers**

find $U$ and $V$ with the property:

$x$ such that $T_{u,v}(x) = T$
Such encoding can be extended to encode labeled trees.

Arbitrary ordered trees can be simulated by binary ordered trees ("left child, right sibling").
Encoding Trees

Problems where tree encoding can give a solution:

- tree isomorphism and related problems.
- tree compression.
- tree generation algorithms.
Components are structured and interconnected.

Together they form a single, complex system.
Nondeterministic Encryption. Deterministic Decryption

Observation of codewords gives no information about $x$.

Strong cryptography.


