Numeral systems & data structures

Jyrki Katajainen\textsuperscript{1)}

Amr Elmasry\textsuperscript{2)} and Claus Jensen\textsuperscript{3)}

\textsuperscript{1)} University of Copenhagen
\textsuperscript{2)} Max-Planck-Institut für Informatik
\textsuperscript{3)} The Royal Library
Can you see the problem?

The decimal numeral system was introduced to the west through Muhammad ibn Mūsā al-Khwārizmī’s book [al-Khwārizmī 825].

\[
\begin{array}{cccccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}
\begin{array}{cccccccccc}
9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9
\end{array}
\begin{array}{cccccccccc}
+ & & & & & & & & & & 1
\end{array}
\begin{array}{cccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
\]

The binary numeral system has the same problem.
Why is this a problem?

An addition of two bits can be a heavy operation!
Goal

Let \( d \) be a positive integer.

\[ \text{rep}(d): \langle d_0, d_1, \ldots, d_{r-1} \rangle \quad (d_0 \text{ is the least significant digit}) \]

\[ \text{value}(d): \sum_{i=0}^{r-1} d_i \times w_i \quad \text{(in our case } w_i = 2^i) \]

Develop a numeral system for which

- \( \max \{d_i \mid i \in \{0, 1, \ldots, r - 1\} \} \) is as small as possible for all \( d \);
- an increment at any position \( i \) \( (\text{increment}(d, i)) \) generates as few digit changes as possible in the worst case; and
- a decrement at any position \( i \) \( (\text{decrement}(d, i)) \) generates as few digit changes as possible in the worst case.
Strictly-regular system

**Digits:** \( d_i \in \{0, 1, 2\} \)

**Strict regularity:** The sequence from the least-significant to the most-significant digit is of the form \( (1^+ | 01^*2)^* (\varepsilon | 01^+) \).

**Extreme digits:** 0 and 2.

Which of the following representations are strictly regular?

a) 1111111
b) 11011211101
c) 1201
d) 1110101
Increment example

Notation: Digit $d_i$ to be increased is displayed in red. $d_a$ is the first extreme digit after $d_i$, $k$ is a positive integer, $\alpha$ denotes any combination of $1^+$ and $01^2$ blocks, and $\omega$ any combination of $1^+$ and $01^2$ blocks followed by at most one $01^+$ block.

Initial configuration: $\alpha 01^*1^*21^k \omega$

Action: $d_i \leftarrow 2$; $d_a \leftarrow d_a - 2$; $d_{a+1} \leftarrow d_{a+1} + 1$

Final configuration: $\alpha 01^*21^*021^{k-1} \omega$
General algorithm

Subroutine \texttt{fix-carry}(d, i): Assert that $d_i \geq 2$. Perform $d_i \leftarrow d_i - 2$ and $d_{i+1} \leftarrow d_{i+1} + 1$.

Algorithm \texttt{increment}(d, i):

1: \texttt{++}$d_i$
2: Let $d_b$ be the first extreme digit before $d_i$, $d_b \in \{0, 2, \text{undefined}\}$
3: Let $d_a$ be the first extreme digit after $d_i$, $d_a \in \{0, 2, \text{undefined}\}$
4: if $d_i = 3$ or ($d_i = 2$ and $d_b \neq 0$)
5: \texttt{fix-carry}(d, i)
6: else if $d_a = 2$
7: \texttt{fix-carry}(d, a)
One of our results

Theorem: Given a strictly-regular representation of $d$, $\text{increment}(d, i)$ and $\text{decrement}(d, i)$ incur at most three digit changes.

Proof: By a case analysis. For $\text{increment}(d, i)$ we must consider 19 cases and for $\text{decrement}(d, i)$ 15 cases. □
Some related systems

Regular system: A regular sequence is of the form \( (0 \mid 1 \mid 01^*2)^* \). Allows increments at any position with \( O(1) \) digit changes [Clancy & Knuth 1977].

Extended regular system: \( d_i \in \{0, 1, 2, 3\} \). Every 3 is preceded by at least one \( \{0, 1\} \) before the next 3 or running out of digits, and every 0 is preceded by at least one \( \{2, 3\} \) before the next 0 or running out of digits. Allows increments and decrements at any position with \( O(1) \) digit changes [Clancy & Knuth 1977; Kaplan & Tarjan 1996].
Full repertoire of operations

**increment**\((d, i)\): Assert that \(i \in \{0, 1, \ldots, r\}\). Perform \(++d_i\) resulting in \(d'\), i.e. \(\text{value}(d') = \text{value}(d) + w_i\). Make \(d'\) valid without changing its value.

**decrement**\((d, i)\): Assert that \(i \in \{0, 1, \ldots, r - 1\}\). Perform \(--d_i\) resulting in \(d'\), i.e. \(\text{value}(d') = \text{value}(d) - w_i\). Make \(d'\) valid without changing its value.

**cut**\((d, i)\): Cut \(\text{rep}(d)\) into two valid sequences having the same value as the numbers corresponding to \(\langle d_0, d_1, \ldots, d_{i-1} \rangle\) and \(\langle d_i, d_{i+1}, \ldots, d_{r-1} \rangle\).

**concatenate**\((d, d')\): Concatenate \(\text{rep}(d)\) and \(\text{rep}(d')\) into one valid sequence that has the same value as \(\langle d_0, d_1, \ldots, d_{r-1}, d'_0, d'_1, \ldots, d'_{r'-1} \rangle\).

**add**\((d, d')\): Construct a valid sequence \(d''\) such that \(\text{value}(d'') = \text{value}(d) + \text{value}(d')\).
Conclusions

- We got the strictly-regular system from God; it was not invented by us.
- It is still open whether the system can be extended for ternary numbers ($w_i = 3^i$).
- Also, it is open whether there exists an equally economical system that allows increments and decrements in $O(1)$ worst-case time (we talked about the number of digit changes, not actual running time).
Further reading
