FORMALIZED MATHEMATICS
Vol. 26, No. 3, Pages 223–229, 2018
DOI: 10.2478/forma-2018-0020

sciendo
https://www.sciendo.com/

Binary Representation of Natural Numbers

Hiroyuki Okazaki[1]
Shinshu University
Nagano, Japan

Summary. Binary representation of integers [5], [3] and arithmetic operations on them have already been introduced in Mizar Mathematical Library [8, 7, 6, 4]. However, these articles formalize the notion of integers as mapped into a certain length tuple of boolean values.

In this article we formalize, by means of Mizar system [2], [1], the binary representation of natural numbers which maps $\mathbb{N}$ into bitstreams.

MSC: 68W01 68T99 03B35

Keywords: algorithms

MML identifier: BINARI_6, version: 8.1.08 5.53.1335

1. Preliminaries

Let us consider a natural number x. Now we state the propositions:

(1) There exists a natural number m such that $x < 2^m$.

(2) If $x \neq 0$, then there exists a natural number n such that $2^n \leqslant x < 2^{n+1}$.
PROOF: Define $\mathcal{Q}$[natural number] $\equiv x < 2^{\$_1}$. There exists a natural number m such that $\mathcal{Q}[m]$. Consider k being a natural number such that $\mathcal{Q}[k]$ and for every natural number n such that $\mathcal{Q}[n]$ holds $k \leqslant n$. Reconsider $k_1 = k - 1$ as a natural number. $2^{k_1} \leqslant x$. □

(3) Let us consider a natural number x, and natural numbers n_1, n_2. If $2^{n_1} \leqslant x < 2^{n_1+1}$ and $2^{n_2} \leqslant x < 2^{n_2+1}$, then $n_1 = n_2$.

[1]This study was supported in part by JSPS KAKENHI Grant Numbers JP17K00182. The author would also like to express gratitude to Prof. Yasunari Shidama for his support and encouragement.

(4) $\langle 0 \rangle = \langle \underbrace{0, \ldots, 0}_{1} \rangle$.

(5) Let us consider natural numbers n_1, n_2. Then $\langle \underbrace{0, \ldots, 0}_{n_1} \rangle \frown \langle \underbrace{0, \ldots, 0}_{n_2} \rangle = \langle \underbrace{0, \ldots, 0}_{n_1+n_2} \rangle$.

2. Homomorphism from the Natural Numbers to the Bitstreams

Let x be a natural number. The functor LenBinSeq(x) yielding a non zero natural number is defined by

(Def. 1) (i) $it = 1$, **if** $x = 0$,

(ii) there exists a natural number n such that $2^n \leqslant x < 2^{n+1}$ and $it = n + 1$, **otherwise**.

Let us consider a natural number x. Now we state the propositions:

(6) $x < 2^{\mathrm{LenBinSeq}(x)}$.

(7) $x = \mathrm{AbsVal}(\mathrm{LenBinSeq}(x)\text{-BinarySequence}(x))$. The theorem is a consequence of (6).

(8) Let us consider a natural number n, and an $(n + 1)$-tuple x of *Boolean*. If $x(n + 1) = 1$, then $2^n \leqslant \mathrm{AbsVal}(x) < 2^{n+1}$.

(9) There exists a function F from *Boolean** into $\mathbb{N}$ such that for every element x of *Boolean**, there exists a $(\mathrm{len}\, x)$-tuple x_0 of *Boolean* such that $x = x_0$ and $F(x) = \mathrm{AbsVal}(x_0)$.

Proof: Define $\mathcal{P}[\text{element of } \mathit{Boolean}^*, \text{object}] \equiv$ there exists a $(\mathrm{len}\,\$_1)$-tuple x_0 of *Boolean* such that $\$_1 = x_0$ and $\$_2 = \mathrm{AbsVal}(x_0)$. For every element x of *Boolean**, there exists an element y of $\mathbb{N}$ such that $\mathcal{P}[x, y]$. Consider f being a function from *Boolean** into $\mathbb{N}$ such that for every element x of *Boolean**, $\mathcal{P}[x, f(x)]$. $\square$

The functor Nat2BinLen yielding a function from $\mathbb{N}$ into *Boolean** is defined by

(Def. 2) for every element x of $\mathbb{N}$, $it(x) = \mathrm{LenBinSeq}(x)\text{-BinarySequence}(x)$.

Now we state the propositions:

(10) Let us consider an element x of $\mathbb{N}$, and a $(\mathrm{LenBinSeq}(x))$-tuple y of *Boolean*. If $(\mathrm{Nat2BinLen})(x) = y$, then $\mathrm{AbsVal}(y) = x$. The theorem is a consequence of (7).

(11) $\mathrm{rng}\,\mathrm{Nat2BinLen} = \{x, \text{ where } x \text{ is an element of } \mathit{Boolean}^* : x(\mathrm{len}\, x) = 1\} \cup \{\langle 0 \rangle\}$.

Proof: For every object z, $z \in \operatorname{rng}$ Nat2BinLen iff $z \in \{x$, where x is an element of $Boolean^*$: $x(\operatorname{len} x) = 1\} \cup \{\langle 0 \rangle\}$. □

(12) Nat2BinLen is one-to-one.

Let x, y be elements of $Boolean^*$. Assume $\operatorname{len} x \neq 0$ and $\operatorname{len} y \neq 0$. The functor MaxLen$(x, y)$ yielding a non zero natural number is defined by the term

(Def. 3) $\max(\operatorname{len} x, \operatorname{len} y)$.

Let K be a natural number and x be an element of $Boolean^*$. The functor ExtBit(x, K) yielding a K-tuple of $Boolean$ is defined by the term

(Def. 4) $\begin{cases} x \frown \underbrace{\langle 0, \ldots, 0 \rangle}_{K -' \operatorname{len} x}, & \textbf{if } \operatorname{len} x \leqslant K, \\ x \restriction K, & \textbf{otherwise}. \end{cases}$

Now we state the propositions:

(13) Let us consider a natural number K, and an element x of $Boolean^*$. Suppose $\operatorname{len} x \leqslant K$. Then ExtBit$(x, K+1) =$ ExtBit$(x, K) \frown \langle 0 \rangle$.

(14) Let us consider a non zero natural number K, and an element x of $Boolean^*$. If $\operatorname{len} x = K$, then ExtBit$(x, K) = x$.

(15) Let us consider a non zero natural number K, K-tuples x, y of $Boolean$, and $(K+1)$-tuples x_1, y_1 of $Boolean$. Suppose $x_1 = x \frown \langle 0 \rangle$ and $y_1 = y \frown \langle 0 \rangle$. Then x_1 and y_1 are summable.

(16) Let us consider a non zero natural number K, and a K-tuple y of $Boolean$. Suppose $y = \underbrace{\langle 0, \ldots, 0 \rangle}_{K}$. Let us consider a non zero natural number n. If $n \leqslant K$, then $y_{/n} = 0$.

(17) Let us consider a non zero natural number K, and K-tuples x, y of $Boolean$. Then carry$(x, y) =$ carry(y, x).

(18) Let us consider a non zero natural number K, and K-tuples x, y of $Boolean$. Suppose $y = \underbrace{\langle 0, \ldots, 0 \rangle}_{K}$. Let us consider a non zero natural number n. Suppose $n \leqslant K$. Then

  (i) $(\text{carry}(x, y))_{/n} = 0$, and

  (ii) $(\text{carry}(y, x))_{/n} = 0$.

Proof: Define $\mathcal{P}$[natural number] $\equiv$ if $1 \leqslant \$_1 \leqslant K$, then $(\text{carry}(x, y))_{/\$_1} = 0$. For every non zero natural number i such that $\mathcal{P}[i]$ holds $\mathcal{P}[i+1]$. For every non zero natural number k, $\mathcal{P}[k]$. □

Let us consider a non zero natural number K and K-tuples x, y of $Boolean$. Now we state the propositions:

(19) $x + y = y + x$. The theorem is a consequence of (17).

(20) If $y = \langle \underbrace{0, \ldots, 0}_{K} \rangle$, then $x + y = x$ and $y + x = x$.

PROOF: For every natural number i such that $i \in \operatorname{Seg} K$ holds $(x+y)(i) = x(i)$. □

(21) Let us consider a non zero natural number K, and K-tuples x, y of *Boolean*. If $x(\operatorname{len} x) = 1$ and $y(\operatorname{len} y) = 1$, then x and y are not summable.

Let us consider a non zero natural number K and K-tuples x, y of *Boolean*. Now we state the propositions:

(22) If x and y are summable, then y and x are summable. The theorem is a consequence of (17).

(23) If x and y are summable and ($x(\operatorname{len} x) = 1$ or $y(\operatorname{len} y) = 1$), then $(x + y)(\operatorname{len}(x + y)) = 1$. The theorem is a consequence of (19) and (22).

(24) Let us consider a non zero natural number K, K-tuples x, y of *Boolean*, and $(K + 1)$-tuples x_1, y_1 of *Boolean*. Suppose x and y are not summable and $x_1 = x \frown \langle 0 \rangle$ and $y_1 = y \frown \langle 0 \rangle$. Then $(x_1 + y_1)(\operatorname{len}(x_1 + y_1)) = 1$.

PROOF: Set $K_1 = K + 1$. Reconsider $S = \operatorname{carry}(x, y) \frown \langle 1 \rangle$ as a K_1-tuple of *Boolean*. $S_{/1} = \mathit{false}$. For every natural number i such that $1 \leqslant i < K_1$ holds $S_{/i+1} = (x_{1/i} \wedge y_{1/i} \vee x_{1/i} \wedge S_{/i}) \vee y_{1/i} \wedge S_{/i}$. □

Let x, y be elements of *Boolean**. The functor $x + y$ yielding an element of *Boolean** is defined by the term

$$\text{(Def. 5)} \quad \begin{cases} y, \textbf{if } \operatorname{len} x = 0, \\ x, \textbf{if } \operatorname{len} y = 0, \\ \operatorname{ExtBit}(x, \operatorname{MaxLen}(x, y)) + \operatorname{ExtBit}(y, \operatorname{MaxLen}(x, y)), \\ \quad \textbf{if } \operatorname{ExtBit}(x, \operatorname{MaxLen}(x, y)) \text{ and } \operatorname{ExtBit}(y, \operatorname{MaxLen}(x, y)) \\ \quad \text{are summable and } \operatorname{len} x \neq 0 \text{ and } \operatorname{len} y \neq 0, \\ \operatorname{ExtBit}(x, \operatorname{MaxLen}(x, y) + 1) + \operatorname{ExtBit}(y, \operatorname{MaxLen}(x, y) + 1), \\ \quad \textbf{otherwise}. \end{cases}$$

Let F be a function from $\mathbb{N}$ into *Boolean** and x be an element of $\mathbb{N}$. Let us note that the functor $F(x)$ yields an element of *Boolean**. Now we state the propositions:

(25) Let us consider an element x of *Boolean**. If $x \in \operatorname{rng} \operatorname{Nat2BinLen}$, then $1 \leqslant \operatorname{len} x$.

(26) Let us consider elements x, y of *Boolean**. Suppose $x, y \in \operatorname{rng} \operatorname{Nat2BinLen}$. Then $x + y \in \operatorname{rng} \operatorname{Nat2BinLen}$. The theorem is a consequence of (11), (25), (4), (18), (16), (20), (14), (21), (23), (13), and (24).

(27) Let us consider a non zero natural number n, an n-tuple x of *Boolean*, natural numbers m, l, and an l-tuple y of *Boolean*. Suppose $y = x \frown \langle \underbrace{0, \ldots, 0}_{m} \rangle$. Then $\operatorname{AbsVal}(y) = \operatorname{AbsVal}(x)$.

Proof: Define $\mathcal{P}$[natural number] $\equiv$ for every natural number l for every l-tuple y of *Boolean* such that $y = x \frown \langle \underbrace{0, \ldots, 0}_{\$_1} \rangle$ holds $\mathrm{AbsVal}(y) = \mathrm{AbsVal}(x)$. For every natural number m such that $\mathcal{P}[m]$ holds $\mathcal{P}[m+1]$. $\mathcal{P}[0]$. For every natural number m, $\mathcal{P}[m]$. $\square$

(28) Let us consider a natural number n, an element x of $\mathbb{N}$, and an n-tuple y of *Boolean*. Suppose $y = (\mathrm{Nat2BinLen})(x)$. Then

(i) $n = \mathrm{LenBinSeq}(x)$, and

(ii) $\mathrm{AbsVal}(y) = x$, and

(iii) $(\mathrm{Nat2BinLen})(\mathrm{AbsVal}(y)) = y$.

The theorem is a consequence of (6).

(29) Let us consider elements x, y of $\mathbb{N}$. Then $(\mathrm{Nat2BinLen})(x + y) = (\mathrm{Nat2BinLen})(x) + (\mathrm{Nat2BinLen})(y)$. The theorem is a consequence of (7), (27), (26), (28), (13), and (15).

(30) Let us consider elements x, y of *Boolean**. If x, $y \in \mathrm{rng}\,\mathrm{Nat2BinLen}$, then $x + y = y + x$. The theorem is a consequence of (29).

(31) Let us consider elements x, y, z of *Boolean**. If x, y, $z \in \mathrm{rng}\,\mathrm{Nat2BinLen}$, then $(x + y) + z = x + (y + z)$. The theorem is a consequence of (29).

3. Homomorphism from the Bitstreams to the Natural Numbers

Let x be an element of *Boolean**. The functor $\mathrm{ExtAbsVal}(x)$ yielding a natural number is defined by

(Def. 6) there exists a natural number n and there exists an n-tuple y of *Boolean* such that $y = x$ and *it* $= \mathrm{AbsVal}(y)$.

Now we state the proposition:

(32) There exists a function F from *Boolean** into $\mathbb{N}$ such that for every element x of *Boolean**, $F(x) = \mathrm{ExtAbsVal}(x)$.

Proof: Define $\mathcal{P}$[element of *Boolean**, object] $\equiv \$_2 = \mathrm{ExtAbsVal}(\$_1)$. For every element x of *Boolean**, there exists an element y of $\mathbb{N}$ such that $\mathcal{P}[x, y]$. Consider f being a function from *Boolean** into $\mathbb{N}$ such that for every element x of *Boolean**, $\mathcal{P}[x, f(x)]$. $\square$

The functor BinLen2Nat yielding a function from *Boolean** into $\mathbb{N}$ is defined by

(Def. 7) for every element x of *Boolean**, *it*$(x) = \mathrm{ExtAbsVal}(x)$.

Let F be a function from $Boolean^*$ into $\mathbb{N}$ and x be an element of $Boolean^*$. Let us observe that the functor $F(x)$ yields an element of $\mathbb{N}$. Observe that BinLen2Nat is onto.

Now we state the propositions:

(33) Let us consider an element x of $Boolean^*$, and a natural number K. Suppose $\operatorname{len} x \neq 0$ and $\operatorname{len} x \leqslant K$. Then $\operatorname{ExtAbsVal}(x) = \operatorname{AbsVal}(\operatorname{ExtBit}(x, K))$. The theorem is a consequence of (27).

(34) Let us consider elements x, y of $Boolean^*$. Then $(\text{BinLen2Nat})(x + y) = (\text{BinLen2Nat})(x) + (\text{BinLen2Nat})(y)$. The theorem is a consequence of (33), (13), and (15).

The functor EqBinLen2Nat yielding an equivalence relation of $Boolean^*$ is defined by

(Def. 8) for every objects x, y, $\langle x, y\rangle \in it$ iff $x, y \in Boolean^*$ and $(\text{BinLen2Nat})(x) = (\text{BinLen2Nat})(y)$.

The functor QuBinLen2Nat yielding a function from Classes EqBinLen2Nat into $\mathbb{N}$ is defined by

(Def. 9) for every element A of Classes EqBinLen2Nat, there exists an object x such that $x \in A$ and $it(A) = (\text{BinLen2Nat})(x)$.

Let us observe that QuBinLen2Nat is one-to-one and onto.

Now we state the proposition:

(35) Let us consider an element x of $Boolean^*$.
Then $(\text{QuBinLen2Nat})([x]_{\text{EqBinLen2Nat}}) = (\text{BinLen2Nat})(x)$.

Let A, B be elements of Classes EqBinLen2Nat. The functor $A+B$ yielding an element of Classes EqBinLen2Nat is defined by

(Def. 10) there exist elements x, y of $Boolean^*$ such that $x \in A$ and $y \in B$ and $it = [x + y]_{\text{EqBinLen2Nat}}$.

Now we state the proposition:

(36) Let us consider elements A, B of Classes EqBinLen2Nat, and elements x, y of $Boolean^*$. If $x \in A$ and $y \in B$, then $A + B = [x + y]_{\text{EqBinLen2Nat}}$. The theorem is a consequence of (34).

Let us consider elements A, B of Classes EqBinLen2Nat. Now we state the propositions:

(37) $(\text{QuBinLen2Nat})(A+B) = (\text{QuBinLen2Nat})(A)+(\text{QuBinLen2Nat})(B)$. The theorem is a consequence of (36), (35), and (34).

(38) $A + B = B + A$. The theorem is a consequence of (36), (35), and (34).

(39) Let us consider elements A, B, C of Classes EqBinLen2Nat. Then $(A + B) + C = A + (B + C)$. The theorem is a consequence of (36), (35), and (34).

(40) Let us consider a natural number n, and elements z, z_1 of $Boolean^*$. Suppose $z = \varepsilon_{Boolean}$ and $z_1 = \langle\underbrace{0,\ldots,0}_{n}\rangle$.

Then $[z]_{\mathrm{EqBinLen2Nat}} = [z_1]_{\mathrm{EqBinLen2Nat}}$.

(41) Let us consider elements A, Z of Classes EqBinLen2Nat, a natural number n, and an element z of $Boolean^*$. Suppose $Z = [z]_{\mathrm{EqBinLen2Nat}}$ and $z = \langle\underbrace{0,\ldots,0}_{n}\rangle$. Then

(i) $A + Z = A$, and

(ii) $Z + A = A$.

The theorem is a consequence of (40), (36), and (38).

References

[1] Grzegorz Bancerek, Czesław Byliński, Adam Grabowski, Artur Korniłowicz, Roman Matuszewski, Adam Naumowicz, Karol Pąk, and Josef Urban. Mizar: State-of-the-art and beyond. In Manfred Kerber, Jacques Carette, Cezary Kaliszyk, Florian Rabe, and Volker Sorge, editors, *Intelligent Computer Mathematics*, volume 9150 of *Lecture Notes in Computer Science*, pages 261–279. Springer International Publishing, 2015. ISBN 978-3-319-20614-1. doi:10.1007/978-3-319-20615-8_17.

[2] Grzegorz Bancerek, Czesław Byliński, Adam Grabowski, Artur Korniłowicz, Roman Matuszewski, Adam Naumowicz, and Karol Pąk. The role of the Mizar Mathematical Library for interactive proof development in Mizar. *Journal of Automated Reasoning*, 61(1):9–32, 2018. doi:10.1007/s10817-017-9440-6.

[3] Donald E. Knuth. *The Art of Computer Programming, Volume 1: Fundamental Algorithms, Third Edition*. Addison-Wesley, 1997.

[4] Hisayoshi Kunimune and Yatsuka Nakamura. A representation of integers by binary arithmetics and addition of integers. *Formalized Mathematics*, 11(**2**):175–178, 2003.

[5] Gottfried Wilhelm Leibniz. *Explication de l'Arithmétique Binaire*, volume 7. C. Gerhardt, Die Mathematische Schriften edition, 223 pages, 1879.

[6] Robert Milewski. Binary arithmetics. Binary sequences. *Formalized Mathematics*, 7(**1**):23–26, 1998.

[7] Yasuho Mizuhara and Takaya Nishiyama. Binary arithmetics, addition and subtraction of integers. *Formalized Mathematics*, 5(**1**):27–29, 1996.

[8] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.

Accepted September 29, 2018