FORMALIZED MATHEMATICS
Volume 6, Number 2, 1997
University of Białystok

2's Complement Circuit

Katsumi Wasaki
National College of Technology
Nagano

Pauline N. Kawamoto
Shinshu University
Nagano

Summary. This article introduces various Boolean operators which are used in discussing the properties and stability of a 2's complement circuit. We present the definitions and related theorems for the following logical operators which include negative input/output: 'and2a', 'or2a', 'xor2a' and 'nand2a', 'nor2a', etc. We formalize the concept of a 2's complement circuit, define the structures of complementors/incrementors for binary operations, and prove the stability of the circuit.

MML Identifier: `TWOSCOMP`.

The terminology and notation used here are introduced in the following articles: [13], [15], [12], [1], [17], [5], [6], [16], [2], [4], [11], [14], [10], [8], [9], [7], and [3].

1. Boolean Operators

Let x be a set. Then $\langle x \rangle$ is a finite sequence with length 1. Let y be a set. Then $\langle x, y \rangle$ is a finite sequence with length 2. Let z be a set. Then $\langle x, y, z \rangle$ is a finite sequence with length 3.

Let n, m be natural numbers, let p be a finite sequence with length n, and let q be a finite sequence with length m. Then $p \frown q$ is a finite sequence with length $n + m$.

Let S be an unsplit non void non empty many sorted signature, let A be a Boolean circuit of S, let s be a state of A, and let v be a vertex of S. Then $s(v)$ is an element of *Boolean*.

Next we state two propositions:

(1) For every function f and for all sets x_1, x_2 such that $x_1 \in \mathrm{dom}\, f$ and $x_2 \in \mathrm{dom}\, f$ holds $f \cdot \langle x_1, x_2 \rangle = \langle f(x_1), f(x_2) \rangle$.

ISSN 1426–2630

(2) For every function f and for all sets x_1, x_2, x_3 such that $x_1 \in \mathrm{dom}\, f$ and $x_2 \in \mathrm{dom}\, f$ and $x_3 \in \mathrm{dom}\, f$ holds $f \cdot \langle x_1, x_2, x_3 \rangle = \langle f(x_1), f(x_2), f(x_3) \rangle$.

The function and_2 from *Boolean*^{2} into *Boolean* is defined by:

(Def. 1) For all elements x, y of *Boolean* holds $\mathrm{and}_2(\langle x, y \rangle) = x \wedge y$.

The function and_{2a} from *Boolean*^{2} into *Boolean* is defined by:

(Def. 2) For all elements x, y of *Boolean* holds $(\mathrm{and}_{2a})(\langle x, y \rangle) = \neg x \wedge y$.

The function and_{2b} from *Boolean*^{2} into *Boolean* is defined as follows:

(Def. 3) For all elements x, y of *Boolean* holds $(\mathrm{and}_{2b})(\langle x, y \rangle) = \neg x \wedge \neg y$.

The function nand_2 from *Boolean*^{2} into *Boolean* is defined by:

(Def. 4) For all elements x, y of *Boolean* holds $\mathrm{nand}_2(\langle x, y \rangle) = \neg(x \wedge y)$.

The function nand_{2a} from *Boolean*^{2} into *Boolean* is defined as follows:

(Def. 5) For all elements x, y of *Boolean* holds $(\mathrm{nand}_{2a})(\langle x, y \rangle) = \neg(\neg x \wedge y)$.

The function nand_{2b} from *Boolean*^{2} into *Boolean* is defined as follows:

(Def. 6) For all elements x, y of *Boolean* holds $(\mathrm{nand}_{2b})(\langle x, y \rangle) = \neg(\neg x \wedge \neg y)$.

The function or_2 from *Boolean*^{2} into *Boolean* is defined by:

(Def. 7) For all elements x, y of *Boolean* holds $\mathrm{or}_2(\langle x, y \rangle) = x \vee y$.

The function or_{2a} from *Boolean*^{2} into *Boolean* is defined as follows:

(Def. 8) For all elements x, y of *Boolean* holds $(\mathrm{or}_{2a})(\langle x, y \rangle) = \neg x \vee y$.

The function or_{2b} from *Boolean*^{2} into *Boolean* is defined as follows:

(Def. 9) For all elements x, y of *Boolean* holds $(\mathrm{or}_{2b})(\langle x, y \rangle) = \neg x \vee \neg y$.

The function nor_2 from *Boolean*^{2} into *Boolean* is defined by:

(Def. 10) For all elements x, y of *Boolean* holds $\mathrm{nor}_2(\langle x, y \rangle) = \neg(x \vee y)$.

The function nor_{2a} from *Boolean*^{2} into *Boolean* is defined by:

(Def. 11) For all elements x, y of *Boolean* holds $(\mathrm{nor}_{2a})(\langle x, y \rangle) = \neg(\neg x \vee y)$.

The function nor_{2b} from *Boolean*^{2} into *Boolean* is defined as follows:

(Def. 12) For all elements x, y of *Boolean* holds $(\mathrm{nor}_{2b})(\langle x, y \rangle) = \neg(\neg x \vee \neg y)$.

The function xor_2 from *Boolean*^{2} into *Boolean* is defined by:

(Def. 13) For all elements x, y of *Boolean* holds $\mathrm{xor}_2(\langle x, y \rangle) = x \oplus y$.

The function xor_{2a} from *Boolean*^{2} into *Boolean* is defined as follows:

(Def. 14) For all elements x, y of *Boolean* holds $(\mathrm{xor}_{2a})(\langle x, y \rangle) = \neg x \oplus y$.

The function xor_{2b} from *Boolean*^{2} into *Boolean* is defined as follows:

(Def. 15) For all elements x, y of *Boolean* holds $(\mathrm{xor}_{2b})(\langle x, y \rangle) = \neg x \oplus \neg y$.

We now state a number of propositions:

(3) For all elements x, y of *Boolean* holds $\mathrm{and}_2(\langle x, y \rangle) = x \wedge y$ and $(\mathrm{and}_{2a})(\langle x, y \rangle) = \neg x \wedge y$ and $(\mathrm{and}_{2b})(\langle x, y \rangle) = \neg x \wedge \neg y$.

(4) For all elements x, y of *Boolean* holds $\mathrm{nand}_2(\langle x, y \rangle) = \neg(x \wedge y)$ and $(\mathrm{nand}_{2a})(\langle x, y \rangle) = \neg(\neg x \wedge y)$ and $(\mathrm{nand}_{2b})(\langle x, y \rangle) = \neg(\neg x \wedge \neg y)$.

(5) For all elements x, y of *Boolean* holds $\mathrm{or}_2(\langle x, y \rangle) = x \vee y$ and $(\mathrm{or}_{2a})(\langle x, y \rangle) = \neg x \vee y$ and $(\mathrm{or}_{2b})(\langle x, y \rangle) = \neg x \vee \neg y$.

(6) For all elements x, y of *Boolean* holds $\text{nor}_2(\langle x, y\rangle) = \neg(x \vee y)$ and $(\text{nor}_{2a})(\langle x, y\rangle) = \neg(\neg x \vee y)$ and $(\text{nor}_{2b})(\langle x, y\rangle) = \neg(\neg x \vee \neg y)$.

(7) For all elements x, y of *Boolean* holds $\text{xor}_2(\langle x, y\rangle) = x \oplus y$ and $(\text{xor}_{2a})(\langle x, y\rangle) = \neg x \oplus y$ and $(\text{xor}_{2b})(\langle x, y\rangle) = \neg x \oplus \neg y$.

(8) For all elements x, y of *Boolean* holds $\text{and}_2(\langle x, y\rangle) = (\text{nor}_{2b})(\langle x, y\rangle)$ and $(\text{and}_{2a})(\langle x, y\rangle) = (\text{nor}_{2a})(\langle y, x\rangle)$ and $(\text{and}_{2b})(\langle x, y\rangle) = \text{nor}_2(\langle x, y\rangle)$.

(9) For all elements x, y of *Boolean* holds $\text{or}_2(\langle x, y\rangle) = (\text{nand}_{2b})(\langle x, y\rangle)$ and $(\text{or}_{2a})(\langle x, y\rangle) = (\text{nand}_{2a})(\langle y, x\rangle)$ and $(\text{or}_{2b})(\langle x, y\rangle) = \text{nand}_2(\langle x, y\rangle)$.

(10) For all elements x, y of *Boolean* holds $(\text{xor}_{2b})(\langle x, y\rangle) = \text{xor}_2(\langle x, y\rangle)$.

(11)(i) $\text{and}_2(\langle 0, 0\rangle) = 0$,
(ii) $\text{and}_2(\langle 0, 1\rangle) = 0$,
(iii) $\text{and}_2(\langle 1, 0\rangle) = 0$,
(iv) $\text{and}_2(\langle 1, 1\rangle) = 1$,
(v) $(\text{and}_{2a})(\langle 0, 0\rangle) = 0$,
(vi) $(\text{and}_{2a})(\langle 0, 1\rangle) = 1$,
(vii) $(\text{and}_{2a})(\langle 1, 0\rangle) = 0$,
(viii) $(\text{and}_{2a})(\langle 1, 1\rangle) = 0$,
(ix) $(\text{and}_{2b})(\langle 0, 0\rangle) = 1$,
(x) $(\text{and}_{2b})(\langle 0, 1\rangle) = 0$,
(xi) $(\text{and}_{2b})(\langle 1, 0\rangle) = 0$, and
(xii) $(\text{and}_{2b})(\langle 1, 1\rangle) = 0$.

(12)(i) $\text{or}_2(\langle 0, 0\rangle) = 0$,
(ii) $\text{or}_2(\langle 0, 1\rangle) = 1$,
(iii) $\text{or}_2(\langle 1, 0\rangle) = 1$,
(iv) $\text{or}_2(\langle 1, 1\rangle) = 1$,
(v) $(\text{or}_{2a})(\langle 0, 0\rangle) = 1$,
(vi) $(\text{or}_{2a})(\langle 0, 1\rangle) = 1$,
(vii) $(\text{or}_{2a})(\langle 1, 0\rangle) = 0$,
(viii) $(\text{or}_{2a})(\langle 1, 1\rangle) = 1$,
(ix) $(\text{or}_{2b})(\langle 0, 0\rangle) = 1$,
(x) $(\text{or}_{2b})(\langle 0, 1\rangle) = 1$,
(xi) $(\text{or}_{2b})(\langle 1, 0\rangle) = 1$, and
(xii) $(\text{or}_{2b})(\langle 1, 1\rangle) = 0$.

(13) $\text{xor}_2(\langle 0, 0\rangle) = 0$ and $\text{xor}_2(\langle 0, 1\rangle) = 1$ and $\text{xor}_2(\langle 1, 0\rangle) = 1$ and $\text{xor}_2(\langle 1, 1\rangle) = 0$ and $(\text{xor}_{2a})(\langle 0, 0\rangle) = 1$ and $(\text{xor}_{2a})(\langle 0, 1\rangle) = 0$ and $(\text{xor}_{2a})(\langle 1, 0\rangle) = 0$ and $(\text{xor}_{2a})(\langle 1, 1\rangle) = 1$.

The function and_3 from $\textit{Boolean}^3$ into *Boolean* is defined as follows:

(Def. 16) For all elements x, y, z of *Boolean* holds $\text{and}_3(\langle x, y, z\rangle) = x \wedge y \wedge z$.

The function and_{3a} from $\textit{Boolean}^3$ into *Boolean* is defined by:

(Def. 17) For all elements x, y, z of *Boolean* holds $(\text{and}_{3a})(\langle x, y, z\rangle) = \neg x \wedge y \wedge z$.

The function and_{3b} from $\textit{Boolean}^3$ into *Boolean* is defined by:

(Def. 18) For all elements x, y, z of *Boolean* holds $(\text{and}_{3b})(\langle x, y, z\rangle) = \neg x \wedge \neg y \wedge z$.

The function and_{3c} from $\textit{Boolean}^3$ into *Boolean* is defined by:

(Def. 19) For all elements x, y, z of *Boolean* holds $(\mathrm{and}_{3c})(\langle x, y, z\rangle) = \neg x \wedge \neg y \wedge \neg z$.

The function nand_3 from $\mathit{Boolean}^3$ into *Boolean* is defined by:

(Def. 20) For all elements x, y, z of *Boolean* holds $\mathrm{nand}_3(\langle x, y, z\rangle) = \neg(x \wedge y \wedge z)$.

The function nand_{3a} from $\mathit{Boolean}^3$ into *Boolean* is defined as follows:

(Def. 21) For all elements x, y, z of *Boolean* holds $(\mathrm{nand}_{3a})(\langle x, y, z\rangle) = \neg(\neg x \wedge y \wedge z)$.

The function nand_{3b} from $\mathit{Boolean}^3$ into *Boolean* is defined as follows:

(Def. 22) For all elements x, y, z of *Boolean* holds $(\mathrm{nand}_{3b})(\langle x, y, z\rangle) = \neg(\neg x \wedge \neg y \wedge z)$.

The function nand_{3c} from $\mathit{Boolean}^3$ into *Boolean* is defined by:

(Def. 23) For all elements x, y, z of *Boolean* holds $(\mathrm{nand}_{3c})(\langle x, y, z\rangle) = \neg(\neg x \wedge \neg y \wedge \neg z)$.

The function or_3 from $\mathit{Boolean}^3$ into *Boolean* is defined by:

(Def. 24) For all elements x, y, z of *Boolean* holds $\mathrm{or}_3(\langle x, y, z\rangle) = x \vee y \vee z$.

The function or_{3a} from $\mathit{Boolean}^3$ into *Boolean* is defined as follows:

(Def. 25) For all elements x, y, z of *Boolean* holds $(\mathrm{or}_{3a})(\langle x, y, z\rangle) = \neg x \vee y \vee z$.

The function or_{3b} from $\mathit{Boolean}^3$ into *Boolean* is defined as follows:

(Def. 26) For all elements x, y, z of *Boolean* holds $(\mathrm{or}_{3b})(\langle x, y, z\rangle) = \neg x \vee \neg y \vee z$.

The function or_{3c} from $\mathit{Boolean}^3$ into *Boolean* is defined as follows:

(Def. 27) For all elements x, y, z of *Boolean* holds $(\mathrm{or}_{3c})(\langle x, y, z\rangle) = \neg x \vee \neg y \vee \neg z$.

The function nor_3 from $\mathit{Boolean}^3$ into *Boolean* is defined by:

(Def. 28) For all elements x, y, z of *Boolean* holds $\mathrm{nor}_3(\langle x, y, z\rangle) = \neg(x \vee y \vee z)$.

The function nor_{3a} from $\mathit{Boolean}^3$ into *Boolean* is defined as follows:

(Def. 29) For all elements x, y, z of *Boolean* holds $(\mathrm{nor}_{3a})(\langle x, y, z\rangle) = \neg(\neg x \vee y \vee z)$.

The function nor_{3b} from $\mathit{Boolean}^3$ into *Boolean* is defined by:

(Def. 30) For all elements x, y, z of *Boolean* holds $(\mathrm{nor}_{3b})(\langle x, y, z\rangle) = \neg(\neg x \vee \neg y \vee z)$.

The function nor_{3c} from $\mathit{Boolean}^3$ into *Boolean* is defined by:

(Def. 31) For all elements x, y, z of *Boolean* holds $(\mathrm{nor}_{3c})(\langle x, y, z\rangle) = \neg(\neg x \vee \neg y \vee \neg z)$.

The function xor_3 from $\mathit{Boolean}^3$ into *Boolean* is defined by:

(Def. 32) For all elements x, y, z of *Boolean* holds $\mathrm{xor}_3(\langle x, y, z\rangle) = x \oplus y \oplus z$.

Next we state a number of propositions:

(14) For all elements x, y, z of *Boolean* holds $\mathrm{and}_3(\langle x, y, z\rangle) = x \wedge y \wedge z$ and $(\mathrm{and}_{3a})(\langle x, y, z\rangle) = \neg x \wedge y \wedge z$ and $(\mathrm{and}_{3b})(\langle x, y, z\rangle) = \neg x \wedge \neg y \wedge z$ and $(\mathrm{and}_{3c})(\langle x, y, z\rangle) = \neg x \wedge \neg y \wedge \neg z$.

(15) Let x, y, z be elements of *Boolean*. Then $\mathrm{nand}_3(\langle x, y, z\rangle) = \neg(x \wedge y \wedge z)$ and $(\mathrm{nand}_{3a})(\langle x, y, z\rangle) = \neg(\neg x \wedge y \wedge z)$ and $(\mathrm{nand}_{3b})(\langle x, y, z\rangle) = \neg(\neg x \wedge \neg y \wedge z)$ and $(\mathrm{nand}_{3c})(\langle x, y, z\rangle) = \neg(\neg x \wedge \neg y \wedge \neg z)$.

(16) For all elements x, y, z of *Boolean* holds $\mathrm{or}_3(\langle x,y,z\rangle) = x \lor y \lor z$ and $(\mathrm{or}_{3a})(\langle x,y,z\rangle) = \neg x \lor y \lor z$ and $(\mathrm{or}_{3b})(\langle x,y,z\rangle) = \neg x \lor \neg y \lor z$ and $(\mathrm{or}_{3c})(\langle x, y,z\rangle) = \neg x \lor \neg y \lor \neg z$.

(17) Let x, y, z be elements of *Boolean*. Then $\mathrm{nor}_3(\langle x,y,z\rangle) = \neg(x \lor y \lor z)$ and $(\mathrm{nor}_{3a})(\langle x,y,z\rangle) = \neg(\neg x \lor y \lor z)$ and $(\mathrm{nor}_{3b})(\langle x,y,z\rangle) = \neg(\neg x \lor \neg y \lor z)$ and $(\mathrm{nor}_{3c})(\langle x,y,z\rangle) = \neg(\neg x \lor \neg y \lor \neg z)$.

(18) For all elements x, y, z of *Boolean* holds $\mathrm{xor}_3(\langle x,y,z\rangle) = x \oplus y \oplus z$.

(19) For all elements x, y, z of *Boolean* holds $\mathrm{and}_3(\langle x,y,z\rangle) = (\mathrm{nor}_{3c})(\langle x, y,z\rangle)$ and $(\mathrm{and}_{3a})(\langle x,y,z\rangle) = (\mathrm{nor}_{3b})(\langle z,y,x\rangle)$ and $(\mathrm{and}_{3b})(\langle x,y,z\rangle) = (\mathrm{nor}_{3a})(\langle z,y,x\rangle)$ and $(\mathrm{and}_{3c})(\langle x,y,z\rangle) = \mathrm{nor}_3(\langle x,y,z\rangle)$.

(20) For all elements x, y, z of *Boolean* holds $\mathrm{or}_3(\langle x,y,z\rangle) = (\mathrm{nand}_{3c})(\langle x, y,z\rangle)$ and $(\mathrm{or}_{3a})(\langle x,y,z\rangle) = (\mathrm{nand}_{3b})(\langle z,y,x\rangle)$ and $(\mathrm{or}_{3b})(\langle x,y,z\rangle) = (\mathrm{nand}_{3a})(\langle z,y,x\rangle)$ and $(\mathrm{or}_{3c})(\langle x,y,z\rangle) = \mathrm{nand}_3(\langle x,y,z\rangle)$.

(21) $\mathrm{and}_3(\langle 0,0,0\rangle) = 0$ and $\mathrm{and}_3(\langle 0,0,1\rangle) = 0$ and $\mathrm{and}_3(\langle 0,1,0\rangle) = 0$ and $\mathrm{and}_3(\langle 0,1,1\rangle) = 0$ and $\mathrm{and}_3(\langle 1,0,0\rangle) = 0$ and $\mathrm{and}_3(\langle 1,0,1\rangle) = 0$ and $\mathrm{and}_3(\langle 1,1,0\rangle) = 0$ and $\mathrm{and}_3(\langle 1,1,1\rangle) = 1$.

(22) $(\mathrm{and}_{3a})(\langle 0,0,0\rangle) = 0$ and $(\mathrm{and}_{3a})(\langle 0,0,1\rangle) = 0$ and $(\mathrm{and}_{3a})(\langle 0,1,0\rangle) = 0$ and $(\mathrm{and}_{3a})(\langle 0,1,1\rangle) = 1$ and $(\mathrm{and}_{3a})(\langle 1,0,0\rangle) = 0$ and $(\mathrm{and}_{3a})(\langle 1,0, 1\rangle) = 0$ and $(\mathrm{and}_{3a})(\langle 1,1,0\rangle) = 0$ and $(\mathrm{and}_{3a})(\langle 1,1,1\rangle) = 0$.

(23) $(\mathrm{and}_{3b})(\langle 0,0,0\rangle) = 0$ and $(\mathrm{and}_{3b})(\langle 0,0,1\rangle) = 1$ and $(\mathrm{and}_{3b})(\langle 0,1,0\rangle) = 0$ and $(\mathrm{and}_{3b})(\langle 0,1,1\rangle) = 0$ and $(\mathrm{and}_{3b})(\langle 1,0,0\rangle) = 0$ and $(\mathrm{and}_{3b})(\langle 1,0, 1\rangle) = 0$ and $(\mathrm{and}_{3b})(\langle 1,1,0\rangle) = 0$ and $(\mathrm{and}_{3b})(\langle 1,1,1\rangle) = 0$.

(24) $(\mathrm{and}_{3c})(\langle 0,0,0\rangle) = 1$ and $(\mathrm{and}_{3c})(\langle 0,0,1\rangle) = 0$ and $(\mathrm{and}_{3c})(\langle 0,1,0\rangle) = 0$ and $(\mathrm{and}_{3c})(\langle 0,1,1\rangle) = 0$ and $(\mathrm{and}_{3c})(\langle 1,0,0\rangle) = 0$ and $(\mathrm{and}_{3c})(\langle 1,0, 1\rangle) = 0$ and $(\mathrm{and}_{3c})(\langle 1,1,0\rangle) = 0$ and $(\mathrm{and}_{3c})(\langle 1,1,1\rangle) = 0$.

(25) $\mathrm{or}_3(\langle 0,0,0\rangle) = 0$ and $\mathrm{or}_3(\langle 0,0,1\rangle) = 1$ and $\mathrm{or}_3(\langle 0,1,0\rangle) = 1$ and $\mathrm{or}_3(\langle 0, 1,1\rangle) = 1$ and $\mathrm{or}_3(\langle 1,0,0\rangle) = 1$ and $\mathrm{or}_3(\langle 1,0,1\rangle) = 1$ and $\mathrm{or}_3(\langle 1,1,0\rangle) = 1$ and $\mathrm{or}_3(\langle 1,1,1\rangle) = 1$.

(26) $(\mathrm{or}_{3a})(\langle 0,0,0\rangle) = 1$ and $(\mathrm{or}_{3a})(\langle 0,0,1\rangle) = 1$ and $(\mathrm{or}_{3a})(\langle 0,1,0\rangle) = 1$ and $(\mathrm{or}_{3a})(\langle 0,1,1\rangle) = 1$ and $(\mathrm{or}_{3a})(\langle 1,0,0\rangle) = 0$ and $(\mathrm{or}_{3a})(\langle 1,0,1\rangle) = 1$ and $(\mathrm{or}_{3a})(\langle 1,1,0\rangle) = 1$ and $(\mathrm{or}_{3a})(\langle 1,1,1\rangle) = 1$.

(27) $(\mathrm{or}_{3b})(\langle 0,0,0\rangle) = 1$ and $(\mathrm{or}_{3b})(\langle 0,0,1\rangle) = 1$ and $(\mathrm{or}_{3b})(\langle 0,1,0\rangle) = 1$ and $(\mathrm{or}_{3b})(\langle 0,1,1\rangle) = 1$ and $(\mathrm{or}_{3b})(\langle 1,0,0\rangle) = 1$ and $(\mathrm{or}_{3b})(\langle 1,0,1\rangle) = 1$ and $(\mathrm{or}_{3b})(\langle 1,1,0\rangle) = 0$ and $(\mathrm{or}_{3b})(\langle 1,1,1\rangle) = 1$.

(28) $(\mathrm{or}_{3c})(\langle 0,0,0\rangle) = 1$ and $(\mathrm{or}_{3c})(\langle 0,0,1\rangle) = 1$ and $(\mathrm{or}_{3c})(\langle 0,1,0\rangle) = 1$ and $(\mathrm{or}_{3c})(\langle 0,1,1\rangle) = 1$ and $(\mathrm{or}_{3c})(\langle 1,0,0\rangle) = 1$ and $(\mathrm{or}_{3c})(\langle 1,0,1\rangle) = 1$ and $(\mathrm{or}_{3c})(\langle 1,1,0\rangle) = 1$ and $(\mathrm{or}_{3c})(\langle 1,1,1\rangle) = 0$.

(29) $\mathrm{xor}_3(\langle 0,0,0\rangle) = 0$ and $\mathrm{xor}_3(\langle 0,0,1\rangle) = 1$ and $\mathrm{xor}_3(\langle 0,1,0\rangle) = 1$ and $\mathrm{xor}_3(\langle 0,1,1\rangle) = 0$ and $\mathrm{xor}_3(\langle 1,0,0\rangle) = 1$ and $\mathrm{xor}_3(\langle 1,0,1\rangle) = 0$ and $\mathrm{xor}_3(\langle 1,1,0\rangle) = 0$ and $\mathrm{xor}_3(\langle 1,1,1\rangle) = 1$.

2. 2's Complement Circuit Properties

Let x, b be sets. The functor $\mathrm{CompStr}(x, b)$ yields an unsplit non void strict non empty many sorted signature with arity held in gates and Boolean denotation held in gates and is defined by:

(Def. 33) $\mathrm{CompStr}(x, b) = \mathrm{1GateCircStr}(\langle x, b\rangle, \mathrm{xor}_{2a})$.

Let x, b be sets. The functor $\mathrm{CompCirc}(x, b)$ yields a strict Boolean circuit of $\mathrm{CompStr}(x, b)$ with denotation held in gates and is defined as follows:

(Def. 34) $\mathrm{CompCirc}(x, b) = \mathrm{1GateCircuit}(x, b, \mathrm{xor}_{2a})$.

Let x, b be sets. The functor $\mathrm{CompOutput}(x, b)$ yielding an element of $\mathrm{InnerVertices}(\mathrm{CompStr}(x, b))$ is defined by:

(Def. 35) $\mathrm{CompOutput}(x, b) = \langle\langle x, b\rangle, \mathrm{xor}_{2a}\rangle$.

Let x, b be sets. The functor $\mathrm{IncrementStr}(x, b)$ yielding an unsplit non void strict non empty many sorted signature with arity held in gates and Boolean denotation held in gates is defined by:

(Def. 36) $\mathrm{IncrementStr}(x, b) = \mathrm{1GateCircStr}(\langle x, b\rangle, \mathrm{and}_{2a})$.

Let x, b be sets. The functor $\mathrm{IncrementCirc}(x, b)$ yields a strict Boolean circuit of $\mathrm{IncrementStr}(x, b)$ with denotation held in gates and is defined as follows:

(Def. 37) $\mathrm{IncrementCirc}(x, b) = \mathrm{1GateCircuit}(x, b, \mathrm{and}_{2a})$.

Let x, b be sets. The functor $\mathrm{IncrementOutput}(x, b)$ yields an element of $\mathrm{InnerVertices}(\mathrm{IncrementStr}(x, b))$ and is defined by:

(Def. 38) $\mathrm{IncrementOutput}(x, b) = \langle\langle x, b\rangle, \mathrm{and}_{2a}\rangle$.

Let x, b be sets. The functor $\mathrm{BitCompStr}(x, b)$ yielding an unsplit non void strict non empty many sorted signature with arity held in gates and Boolean denotation held in gates is defined as follows:

(Def. 39) $\mathrm{BitCompStr}(x, b) = \mathrm{CompStr}(x, b) +\cdot \mathrm{IncrementStr}(x, b)$.

Let x, b be sets. The functor $\mathrm{BitCompCirc}(x, b)$ yielding a strict Boolean circuit of $\mathrm{BitCompStr}(x, b)$ with denotation held in gates is defined by:

(Def. 40) $\mathrm{BitCompCirc}(x, b) = \mathrm{CompCirc}(x, b) +\cdot \mathrm{IncrementCirc}(x, b)$.

One can prove the following propositions:

(30) For all non pair sets x, b holds $\mathrm{InnerVertices}(\mathrm{CompStr}(x, b))$ is a binary relation.

(31) For all non pair sets x, b holds $x \in$ the carrier of $\mathrm{CompStr}(x, b)$ and $b \in$ the carrier of $\mathrm{CompStr}(x, b)$ and $\langle\langle x, b\rangle, \mathrm{xor}_{2a}\rangle \in$ the carrier of $\mathrm{CompStr}(x, b)$.

(32) For all non pair sets x, b holds the carrier of $\mathrm{CompStr}(x, b) = \{x, b\} \cup \{\langle\langle x, b\rangle, \mathrm{xor}_{2a}\rangle\}$.

(33) For all non pair sets x, b holds $\mathrm{InnerVertices}(\mathrm{CompStr}(x, b)) = \{\langle\langle x, b\rangle, \mathrm{xor}_{2a}\rangle\}$.

(34) For all non pair sets x, b holds $\langle\!\langle \langle x, b\rangle, \text{xor}_{2a} \rangle\!\rangle \in$ InnerVertices(CompStr(x, b)).

(35) For all non pair sets x, b holds InputVertices(CompStr(x, b)) $= \{x, b\}$.

(36) For all non pair sets x, b holds $x \in$ InputVertices(CompStr(x, b)) and $b \in$ InputVertices(CompStr(x, b)).

(37) For all non pair sets x, b holds InputVertices(CompStr(x, b)) has no pairs.

(38) For all non pair sets x, b holds InnerVertices(IncrementStr(x, b)) is a binary relation.

(39) For all non pair sets x, b holds $x \in$ the carrier of IncrementStr(x, b) and $b \in$ the carrier of IncrementStr(x, b) and $\langle\!\langle \langle x, b\rangle, \text{and}_{2a} \rangle\!\rangle \in$ the carrier of IncrementStr(x, b).

(40) For all non pair sets x, b holds the carrier of IncrementStr(x, b) $= \{x, b\} \cup \{\langle\!\langle \langle x, b\rangle, \text{and}_{2a} \rangle\!\rangle\}$.

(41) For all non pair sets x, b holds InnerVertices(IncrementStr(x, b)) $= \{\langle\!\langle \langle x, b\rangle, \text{and}_{2a} \rangle\!\rangle\}$.

(42) For all non pair sets x, b holds $\langle\!\langle \langle x, b\rangle, \text{and}_{2a} \rangle\!\rangle \in$ InnerVertices(IncrementStr(x, b)).

(43) For all non pair sets x, b holds InputVertices(IncrementStr(x, b)) $= \{x, b\}$.

(44) For all non pair sets x, b holds $x \in$ InputVertices(IncrementStr(x, b)) and $b \in$ InputVertices(IncrementStr(x, b)).

(45) For all non pair sets x, b holds InputVertices(IncrementStr(x, b)) has no pairs.

(46) For all non pair sets x, b holds InnerVertices(BitCompStr(x, b)) is a binary relation.

(47) Let x, b be non pair sets. Then
- (i) $x \in$ the carrier of BitCompStr(x, b),
- (ii) $b \in$ the carrier of BitCompStr(x, b),
- (iii) $\langle\!\langle \langle x, b\rangle, \text{xor}_{2a} \rangle\!\rangle \in$ the carrier of BitCompStr(x, b), and
- (iv) $\langle\!\langle \langle x, b\rangle, \text{and}_{2a} \rangle\!\rangle \in$ the carrier of BitCompStr(x, b).

(48) For all non pair sets x, b holds the carrier of BitCompStr(x, b) $= \{x, b\} \cup \{\langle\!\langle \langle x, b\rangle, \text{xor}_{2a} \rangle\!\rangle, \langle\!\langle \langle x, b\rangle, \text{and}_{2a} \rangle\!\rangle\}$.

(49) For all non pair sets x, b holds InnerVertices(BitCompStr(x, b)) $= \{\langle\!\langle \langle x, b\rangle, \text{xor}_{2a} \rangle\!\rangle, \langle\!\langle \langle x, b\rangle, \text{and}_{2a} \rangle\!\rangle\}$.

(50) For all non pair sets x, b holds $\langle\!\langle \langle x, b\rangle, \text{xor}_{2a} \rangle\!\rangle \in$ InnerVertices(BitCompStr(x, b)) and $\langle\!\langle \langle x, b\rangle, \text{and}_{2a} \rangle\!\rangle \in$ InnerVertices(BitCompStr(x, b)).

(51) For all non pair sets x, b holds InputVertices(BitCompStr(x, b)) $= \{x, b\}$.

(52) For all non pair sets x, b holds $x \in$ InputVertices(BitCompStr(x, b)) and $b \in$ InputVertices(BitCompStr(x, b)).

(53) For all non pair sets x, b holds InputVertices(BitCompStr(x, b)) has no pairs.

(54) For all non pair sets x, b and for every state s of CompCirc(x, b) holds (Following(s))(CompOutput(x, b)) $=$ $(\mathrm{xor}_{2a})(\langle s(x), s(b)\rangle)$ and (Following(s))$(x) = s(x)$ and (Following(s))$(b) = s(b)$.

(55) Let x, b be non pair sets, s be a state of CompCirc(x, b), and a_1, a_2 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(b)$, then (Following(s))(CompOutput(x, b)) $= \neg a_1 \oplus a_2$ and (Following(s))$(x) = a_1$ and (Following(s))$(b) = a_2$.

(56) For all non pair sets x, b and for every state s of BitCompCirc(x, b) holds (Following(s))(CompOutput(x, b)) $=$ $(\mathrm{xor}_{2a})(\langle s(x), s(b)\rangle)$ and (Following(s))$(x) = s(x)$ and (Following(s))$(b) = s(b)$.

(57) Let x, b be non pair sets, s be a state of BitCompCirc(x, b), and a_1, a_2 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(b)$, then (Following(s))(CompOutput(x, b)) $= \neg a_1 \oplus a_2$ and (Following(s))$(x) = a_1$ and (Following(s))$(b) = a_2$.

(58) For all non pair sets x, b and for every state s of IncrementCirc(x, b) holds (Following(s))(IncrementOutput(x, b)) $= (\mathrm{and}_{2a})(\langle s(x), s(b)\rangle)$ and (Following(s))$(x) = s(x)$ and (Following(s))$(b) = s(b)$.

(59) Let x, b be non pair sets, s be a state of IncrementCirc(x, b), and a_1, a_2 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(b)$, then (Following(s))(IncrementOutput(x, b)) $= \neg a_1 \wedge a_2$ and (Following(s))$(x) = a_1$ and (Following(s))$(b) = a_2$.

(60) For all non pair sets x, b and for every state s of BitCompCirc(x, b) holds (Following(s))(IncrementOutput(x, b)) $= (\mathrm{and}_{2a})(\langle s(x), s(b)\rangle)$ and (Following(s))$(x) = s(x)$ and (Following(s))$(b) = s(b)$.

(61) Let x, b be non pair sets, s be a state of BitCompCirc(x, b), and a_1, a_2 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(b)$, then (Following(s))(IncrementOutput(x, b)) $= \neg a_1 \wedge a_2$ and (Following(s))$(x) = a_1$ and (Following(s))$(b) = a_2$.

(62) Let x, b be non pair sets and s be a state of BitCompCirc(x, b). Then (Following(s))(CompOutput(x, b)) $= (\mathrm{xor}_{2a})(\langle s(x), s(b)\rangle)$ and (Following(s))(IncrementOutput(x, b)) $= (\mathrm{and}_{2a})(\langle s(x), s(b)\rangle)$ and (Following(s))$(x) = s(x)$ and (Following(s))$(b) = s(b)$.

(63) Let x, b be non pair sets, s be a state of BitCompCirc(x, b), and a_1, a_2 be elements of *Boolean*. Suppose $a_1 = s(x)$ and $a_2 = s(b)$. Then (Following(s))(CompOutput(x, b)) $= \neg a_1 \oplus a_2$ and (Following(s))(IncrementOutput(x, b)) $= \neg a_1 \wedge a_2$ and (Following(s))$(x) = a_1$ and (Following(s))$(b) = a_2$.

(64) For all non pair sets x, b and for every state s of BitCompCirc(x, b) holds Following(s) is stable.

References

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.

[2] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[3] Grzegorz Bancerek and Yatsuka Nakamura. Full adder circuit. Part I. *Formalized Mathematics*, 5(**3**):367–380, 1996.
[4] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(**3**):529–536, 1990.
[5] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[6] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[7] Yatsuka Nakamura and Grzegorz Bancerek. Combining of circuits. *Formalized Mathematics*, 5(**2**):283–295, 1996.
[8] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Introduction to circuits, I. *Formalized Mathematics*, 5(**2**):227–232, 1996.
[9] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Introduction to circuits, II. *Formalized Mathematics*, 5(**2**):273–278, 1996.
[10] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, II. *Formalized Mathematics*, 5(**2**):215–220, 1996.
[11] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[12] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[13] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[14] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[15] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[16] Edmund Woronowicz. Many–argument relations. *Formalized Mathematics*, 1(**4**):733–737, 1990.
[17] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received October 25, 1996