JOURNAL OF FORMALIZED MATHEMATICS
Volume 7, Released 1995, Published 2003
Inst. of Computer Science, Univ. of Białystok

Full Adder Circuit. Part I[1]

Grzegorz Bancerek
Institute of Mathematics
Polish Academy of Sciences

Yatsuka Nakamura
Shinshu University
Nagano

Summary. We continue the formalisation of circuits started by Piotr Rudnicki, Andrzej Trybulec, Pauline Kawamoto, and the second author in [12], [13], [11], [14]. The first step in proving properties of full *n*-bit adder circuit, i.e. 1-bit adder, is presented. We employ the notation of combining circuits introduced in [10].

MML Identifier: FACIRC_1.

WWW: http://mizar.org/JFM/Vol7/facirc_1.html

The articles [17], [16], [21], [20], [1], [18], [22], [4], [5], [3], [9], [7], [8], [23], [15], [2], [6], [19], [13], [14], and [10] provide the notation and terminology for this paper.

1. Combining of Many Sorted Signatures

Let I_1 be a set. We say that I_1 is pair if and only if:

(Def. 1) There exist sets x, y such that $I_1 = \langle x, y \rangle$.

Let us note that every set which is pair is also non empty.
Let x, y be sets. Note that $\langle x, y \rangle$ is pair.
One can verify that there exists a set which is pair and there exists a set which is non pair.
Let us observe that every natural number is non pair.
Let I_1 be a set. We say that I_1 has a pair if and only if:

(Def. 2) There exists a pair set x such that $x \in I_1$.

We introduce I_1 has no pairs as an antonym of I_1 has a pair.

Observe that every set which is empty has also no pairs. Let x be a non pair set. Observe that $\{x\}$ has no pairs. Let y be a non pair set. Note that $\{x,y\}$ has no pairs. Let z be a non pair set. One can verify that $\{x,y,z\}$ has no pairs.

Let us observe that there exists a non empty set which has no pairs.

Let X, Y be sets with no pairs. One can check that $X \cup Y$ has no pairs.

Let X be a set with no pairs and let Y be a set. One can check the following observations:

* $X \setminus Y$ has no pairs,
* $X \cap Y$ has no pairs, and
* $Y \cap X$ has no pairs.

[1]This work was written while the first author visited Shinshu University, July–August 1994.

Let x be a pair set. Observe that $\{x\}$ is relation-like. Let y be a pair set. One can verify that $\{x,y\}$ is relation-like. Let z be a pair set. Note that $\{x,y,z\}$ is relation-like.

One can verify that every set which is relation-like and has no pairs is also empty.

Let I_1 be a function. We say that I_1 is nonpair yielding if and only if:

(Def. 3) For every set x such that $x \in \operatorname{dom} I_1$ holds $I_1(x)$ is non pair.

Let x be a non pair set. Note that $\langle x \rangle$ is nonpair yielding. Let y be a non pair set. One can verify that $\langle x,y \rangle$ is nonpair yielding. Let z be a non pair set. Observe that $\langle x,y,z \rangle$ is nonpair yielding.

Next we state the proposition

(1) For every function f such that f is nonpair yielding holds $\operatorname{rng} f$ has no pairs.

Let n be a natural number. Observe that there exists a finite sequence with length n which is one-to-one and nonpair yielding.

Let us observe that there exists a finite sequence which is one-to-one and nonpair yielding.

Let f be a nonpair yielding function. Observe that $\operatorname{rng} f$ has no pairs.

One can prove the following propositions:

(2) Let S_1, S_2 be non empty many sorted signatures. Suppose $S_1 \approx S_2$ and InnerVertices(S_1) is a binary relation and InnerVertices(S_2) is a binary relation. Then InnerVertices$(S_1 {+}\cdot S_2)$ is a binary relation.

(3) Let S_1, S_2 be unsplit non empty many sorted signatures with arity held in gates. Suppose InnerVertices(S_1) is a binary relation and InnerVertices(S_2) is a binary relation. Then InnerVertices$(S_1 {+}\cdot S_2)$ is a binary relation.

(4) For all non empty many sorted signatures S_1, S_2 such that $S_1 \approx S_2$ and InnerVertices(S_2) misses InputVertices(S_1) holds InputVertices$(S_1) \subseteq$ InputVertices$(S_1 {+}\cdot S_2)$ and InputVertices$(S_1 {+}\cdot S_2) =$ InputVertices$(S_1) \cup ($InputVertices$(S_2) \setminus$ InnerVertices$(S_1))$.

(5) For all sets X, R such that X has no pairs and R is a binary relation holds X misses R.

(6) Let S_1, S_2 be unsplit non empty many sorted signatures with arity held in gates. Suppose InputVertices(S_1) has no pairs and InnerVertices(S_2) is a binary relation. Then InputVertices$(S_1) \subseteq$ InputVertices$(S_1 {+}\cdot S_2)$ and InputVertices$(S_1 {+}\cdot S_2) =$ InputVertices$(S_1) \cup$ (InputVertices$(S_2) \setminus$ InnerVertices$(S_1))$.

(7) Let S_1, S_2 be unsplit non empty many sorted signatures with arity held in gates. Suppose InputVertices(S_1) has no pairs and InnerVertices(S_1) is a binary relation and InputVertices(S_2) has no pairs and InnerVertices(S_2) is a binary relation. Then InputVertices$(S_1 {+}\cdot S_2) =$ InputVertices$(S_1) \cup$ InputVertices(S_2).

(8) For all non empty many sorted signatures S_1, S_2 such that $S_1 \approx S_2$ and InputVertices(S_1) has no pairs and InputVertices(S_2) has no pairs holds InputVertices$(S_1 {+}\cdot S_2)$ has no pairs.

(9) Let S_1, S_2 be unsplit non empty many sorted signatures with arity held in gates. If InputVertices(S_1) has no pairs and InputVertices(S_2) has no pairs, then InputVertices$(S_1 {+}\cdot S_2)$ has no pairs.

2. Combining of Circuits

In this article we present several logical schemes. The scheme *2AryBooleEx* deals with a binary functor $\mathcal{F}$ yielding an element of *Boolean*, and states that:

There exists a function f from *Boolean*^{2} into *Boolean* such that for all elements x, y of *Boolean* holds $f(\langle x,y \rangle) = \mathcal{F}(x,y)$

for all values of the parameter.

The scheme *2AryBooleUniq* deals with a binary functor $\mathcal{F}$ yielding an element of *Boolean*, and states that:

Let f_1, f_2 be functions from *Boolean*^{2} into *Boolean*. Suppose for all elements x, y of *Boolean* holds $f_1(\langle x,y\rangle) = \mathcal{F}(x,y)$ and for all elements x, y of *Boolean* holds $f_2(\langle x, y\rangle) = \mathcal{F}(x,y)$. Then $f_1 = f_2$

for all values of the parameter.

The scheme *2AryBooleDef* deals with a binary functor $\mathcal{F}$ yielding an element of *Boolean*, and states that:

(i) There exists a function f from *Boolean*^{2} into *Boolean* such that for all elements x, y of *Boolean* holds $f(\langle x,y\rangle) = \mathcal{F}(x,y)$, and

(ii) for all functions f_1, f_2 from *Boolean*^{2} into *Boolean* such that for all elements x, y of *Boolean* holds $f_1(\langle x,y\rangle) = \mathcal{F}(x,y)$ and for all elements x, y of *Boolean* holds $f_2(\langle x,y\rangle) = \mathcal{F}(x,y)$ holds $f_1 = f_2$

for all values of the parameter.

The scheme *3AryBooleEx* deals with a ternary functor $\mathcal{F}$ yielding an element of *Boolean*, and states that:

There exists a function f from *Boolean*^{3} into *Boolean* such that for all elements x, y, z of *Boolean* holds $f(\langle x,y,z\rangle) = \mathcal{F}(x,y,z)$

for all values of the parameter.

The scheme *3AryBooleUniq* deals with a ternary functor $\mathcal{F}$ yielding an element of *Boolean*, and states that:

Let f_1, f_2 be functions from *Boolean*^{3} into *Boolean*. Suppose for all elements x, y, z of *Boolean* holds $f_1(\langle x,y,z\rangle) = \mathcal{F}(x,y,z)$ and for all elements x, y, z of *Boolean* holds $f_2(\langle x,y,z\rangle) = \mathcal{F}(x,y,z)$. Then $f_1 = f_2$

for all values of the parameter.

The scheme *3AryBooleDef* deals with a ternary functor $\mathcal{F}$ yielding an element of *Boolean*, and states that:

(i) There exists a function f from *Boolean*^{3} into *Boolean* such that for all elements x, y, z of *Boolean* holds $f(\langle x,y,z\rangle) = \mathcal{F}(x,y,z)$, and

(ii) for all functions f_1, f_2 from *Boolean*^{3} into *Boolean* such that for all elements x, y, z of *Boolean* holds $f_1(\langle x,y,z\rangle) = \mathcal{F}(x,y,z)$ and for all elements x, y, z of *Boolean* holds $f_2(\langle x,y,z\rangle) = \mathcal{F}(x,y,z)$ holds $f_1 = f_2$

for all values of the parameter.

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

(Def. 4) For all elements x, y of *Boolean* holds $\mathrm{xor}(\langle x,y\rangle) = x \oplus y$.

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

(Def. 5) For all elements x, y of *Boolean* holds $\mathrm{or}(\langle x,y\rangle) = x \lor y$.

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

(Def. 6) For all elements x, y of *Boolean* holds $\&(\langle x,y\rangle) = x \land y$.

The function or_3 from *Boolean*^{3} into *Boolean* is defined as follows:

(Def. 7) For all elements x, y, z of *Boolean* holds $\mathrm{or}_3(\langle x,y,z\rangle) = x \lor y \lor z$.

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 $n+m$.

3. Signatures with One Operation

One can prove the following proposition

(10) Let S be a circuit-like non void non empty many sorted signature, A be a non-empty circuit of S, s be a state of A, and g be a gate of S. Then $(\mathrm{Following}(s))(\text{the result sort of } g) = (\mathrm{Den}(g, A))(s \cdot \mathrm{Arity}(g))$.

Let S be a non void circuit-like non empty many sorted signature, let A be a non-empty circuit of S, let s be a state of A, and let n be a natural number. The functor $\mathrm{Following}(s,n)$ yields a state of A and is defined by the condition (Def. 8).

(Def. 8) There exists a function f from $\mathbb{N}$ into $\prod$(the sorts of A) such that $\mathrm{Following}(s,n) = f(n)$ and $f(0) = s$ and for every natural number n holds $f(n+1) = \mathrm{Following}(f(n))$.

One can prove the following propositions:

(11) Let S be a circuit-like non void non empty many sorted signature, A be a non-empty circuit of S, and s be a state of A. Then $\mathrm{Following}(s,0) = s$.

(12) Let S be a circuit-like non void non empty many sorted signature, A be a non-empty circuit of S, s be a state of A, and n be a natural number. Then $\mathrm{Following}(s,n+1) = \mathrm{Following}(\mathrm{Following}(s,n))$.

(13) Let S be a circuit-like non void non empty many sorted signature, A be a non-empty circuit of S, s be a state of A, and n, m be natural numbers. Then $\mathrm{Following}(s,n+m) = \mathrm{Following}(\mathrm{Following}(s,n),m)$.

(14) Let S be a non void circuit-like non empty many sorted signature, A be a non-empty circuit of S, and s be a state of A. Then $\mathrm{Following}(s,1) = \mathrm{Following}(s)$.

(15) Let S be a non void circuit-like non empty many sorted signature, A be a non-empty circuit of S, and s be a state of A. Then $\mathrm{Following}(s,2) = \mathrm{Following}(\mathrm{Following}(s))$.

(16) Let S be a circuit-like non void non empty many sorted signature, A be a non-empty circuit of S, s be a state of A, and n be a natural number. Then $\mathrm{Following}(s,n+1) = \mathrm{Following}(\mathrm{Following}(s),n)$.

Let S be a non void circuit-like non empty many sorted signature, let A be a non-empty circuit of S, let s be a state of A, and let x be a set. We say that s is stable at x if and only if:

(Def. 9) For every natural number n holds $(\mathrm{Following}(s,n))(x) = s(x)$.

The following propositions are true:

(17) Let S be a non void circuit-like non empty many sorted signature, A be a non-empty circuit of S, s be a state of A, and x be a set. If s is stable at x, then for every natural number n holds $\mathrm{Following}(s,n)$ is stable at x.

(18) Let S be a non void circuit-like non empty many sorted signature, A be a non-empty circuit of S, s be a state of A, and x be a set. If $x \in \mathrm{InputVertices}(S)$, then s is stable at x.

(19) Let S be a non void circuit-like non empty many sorted signature, A be a non-empty circuit of S, s be a state of A, and g be a gate of S. Suppose that for every set x such that $x \in \mathrm{rng}\,\mathrm{Arity}(g)$ holds s is stable at x. Then $\mathrm{Following}(s)$ is stable at the result sort of g.

4. UNSPLIT CONDITION

Next we state a number of propositions:

(20) Let S_1, S_2 be non empty many sorted signatures and v be a vertex of S_1. Then $v \in$ the carrier of $S_1 +\cdot S_2$ and $v \in$ the carrier of $S_2 +\cdot S_1$.

(21) Let S_1, S_2 be unsplit non empty many sorted signatures with arity held in gates and x be a set. If $x \in \mathrm{InnerVertices}(S_1)$, then $x \in \mathrm{InnerVertices}(S_1 +\cdot S_2)$ and $x \in \mathrm{InnerVertices}(S_2 +\cdot S_1)$.

(22) For all non empty many sorted signatures S_1, S_2 and for every set x such that $x \in \mathrm{InnerVertices}(S_2)$ holds $x \in \mathrm{InnerVertices}(S_1 +\cdot S_2)$.

(23) For all unsplit non empty many sorted signatures S_1, S_2 with arity held in gates holds $S_1 + \cdot S_2 = S_2 + \cdot S_1$.

(24) Let S_1, S_2 be unsplit non void non empty many sorted signatures with arity held in gates and Boolean denotation held in gates, A_1 be a Boolean circuit of S_1 with denotation held in gates, and A_2 be a Boolean circuit of S_2 with denotation held in gates. Then $A_1 + \cdot A_2 = A_2 + \cdot A_1$.

(25) Let S_1, S_2, S_3 be unsplit non void non empty many sorted signatures with arity held in gates and Boolean denotation held in gates, A_1 be a Boolean circuit of S_1, A_2 be a Boolean circuit of S_2, and A_3 be a Boolean circuit of S_3. Then $(A_1 + \cdot A_2) + \cdot A_3 = A_1 + \cdot (A_2 + \cdot A_3)$.

(26) Let S_1, S_2 be unsplit non void non empty many sorted signatures with arity held in gates and Boolean denotation held in gates, A_1 be a Boolean non-empty circuit of S_1 with denotation held in gates, A_2 be a Boolean non-empty circuit of S_2 with denotation held in gates, and s be a state of $A_1 + \cdot A_2$. Then $s \restriction$the carrier of S_1 is a state of A_1 and $s \restriction$the carrier of S_2 is a state of A_2.

(27) For all unsplit non empty many sorted signatures S_1, S_2 with arity held in gates holds $\mathrm{InnerVertices}(S_1 + \cdot S_2) = \mathrm{InnerVertices}(S_1) \cup \mathrm{InnerVertices}(S_2)$.

(28) Let S_1, S_2 be unsplit non void non empty many sorted signatures with arity held in gates and Boolean denotation held in gates. Suppose $\mathrm{InnerVertices}(S_2)$ misses $\mathrm{InputVertices}(S_1)$. Let A_1 be a Boolean circuit of S_1 with denotation held in gates, A_2 be a Boolean circuit of S_2 with denotation held in gates, s be a state of $A_1 + \cdot A_2$, and s_1 be a state of A_1. If $s_1 = s \restriction$the carrier of S_1, then $\mathrm{Following}(s) \restriction$the carrier of $S_1 = \mathrm{Following}(s_1)$.

(29) Let S_1, S_2 be unsplit non void non empty many sorted signatures with arity held in gates and Boolean denotation held in gates. Suppose $\mathrm{InnerVertices}(S_1)$ misses $\mathrm{InputVertices}(S_2)$. Let A_1 be a Boolean circuit of S_1 with denotation held in gates, A_2 be a Boolean circuit of S_2 with denotation held in gates, s be a state of $A_1 + \cdot A_2$, and s_2 be a state of A_2. If $s_2 = s \restriction$the carrier of S_2, then $\mathrm{Following}(s) \restriction$the carrier of $S_2 = \mathrm{Following}(s_2)$.

(30) Let S_1, S_2 be unsplit non void non empty many sorted signatures with arity held in gates and Boolean denotation held in gates. Suppose $\mathrm{InnerVertices}(S_2)$ misses $\mathrm{InputVertices}(S_1)$. Let A_1 be a Boolean circuit of S_1 with denotation held in gates, A_2 be a Boolean circuit of S_2 with denotation held in gates, s be a state of $A_1 + \cdot A_2$, and s_1 be a state of A_1. Suppose $s_1 = s \restriction$the carrier of S_1. Let n be a natural number. Then $\mathrm{Following}(s, n) \restriction$the carrier of $S_1 = \mathrm{Following}(s_1, n)$.

(31) Let S_1, S_2 be unsplit non void non empty many sorted signatures with arity held in gates and Boolean denotation held in gates. Suppose $\mathrm{InnerVertices}(S_1)$ misses $\mathrm{InputVertices}(S_2)$. Let A_1 be a Boolean circuit of S_1 with denotation held in gates, A_2 be a Boolean circuit of S_2 with denotation held in gates, s be a state of $A_1 + \cdot A_2$, and s_2 be a state of A_2. Suppose $s_2 = s \restriction$the carrier of S_2. Let n be a natural number. Then $\mathrm{Following}(s, n) \restriction$the carrier of $S_2 = \mathrm{Following}(s_2, n)$.

(32) Let S_1, S_2 be unsplit non void non empty many sorted signatures with arity held in gates and Boolean denotation held in gates. Suppose $\mathrm{InnerVertices}(S_2)$ misses $\mathrm{InputVertices}(S_1)$. Let A_1 be a Boolean circuit of S_1 with denotation held in gates, A_2 be a Boolean circuit of S_2 with denotation held in gates, s be a state of $A_1 + \cdot A_2$, and s_1 be a state of A_1. Suppose $s_1 = s \restriction$the carrier of S_1. Let v be a set. Suppose $v \in$ the carrier of S_1. Let n be a natural number. Then $(\mathrm{Following}(s, n))(v) = (\mathrm{Following}(s_1, n))(v)$.

(33) Let S_1, S_2 be unsplit non void non empty many sorted signatures with arity held in gates and Boolean denotation held in gates. Suppose $\mathrm{InnerVertices}(S_1)$ misses $\mathrm{InputVertices}(S_2)$. Let A_1 be a Boolean circuit of S_1 with denotation held in gates, A_2 be a Boolean circuit of S_2 with denotation held in gates, s be a state of $A_1 + \cdot A_2$, and s_2 be a state of A_2. Suppose $s_2 = s \restriction$the carrier of S_2. Let v be a set. Suppose $v \in$ the carrier of S_2. Let n be a natural number. Then $(\mathrm{Following}(s, n))(v) = (\mathrm{Following}(s_2, n))(v)$.

Let S be a non void non empty many sorted signature with denotation held in gates and let g be a gate of S. One can verify that $g_{\mathbf{2}}$ is function-like and relation-like.

The following four propositions are true:

(34) Let S be a circuit-like non void non empty many sorted signature with denotation held in gates and A be a non-empty circuit of S. Suppose A has denotation held in gates. Let s be a state of A and g be a gate of S. Then $(\mathrm{Following}(s))(\text{the result sort of } g) = g_{\mathbf{2}}(s \cdot \mathrm{Arity}(g))$.

(35) Let S be an unsplit non void non empty many sorted signature with arity held in gates and Boolean denotation held in gates, A be a Boolean non-empty circuit of S with denotation held in gates, s be a state of A, p be a finite sequence, and f be a function. If $\langle p, f\rangle \in$ the operation symbols of S, then $(\mathrm{Following}(s))(\langle p, f\rangle) = f(s \cdot p)$.

(36) Let S be an unsplit non void non empty many sorted signature with arity held in gates and Boolean denotation held in gates, A be a Boolean non-empty circuit of S with denotation held in gates, s be a state of A, p be a finite sequence, and f be a function. Suppose $\langle p, f\rangle \in$ the operation symbols of S and for every set x such that $x \in \mathrm{rng}\, p$ holds s is stable at x. Then $\mathrm{Following}(s)$ is stable at $\langle p, f\rangle$.

(37) For every unsplit non empty many sorted signature S holds $\mathrm{InnerVertices}(S) =$ the operation symbols of S.

5. One Gate Circuits

Next we state a number of propositions:

(38) For every set f and for every finite sequence p holds $\mathrm{InnerVertices}(\mathrm{1GateCircStr}(p, f))$ is a binary relation.

(39) For every set f and for every nonpair yielding finite sequence p holds $\mathrm{InputVertices}(\mathrm{1GateCircStr}(p, f))$ has no pairs.

(40) For every set f and for all sets x, y holds $\mathrm{InputVertices}(\mathrm{1GateCircStr}(\langle x,y\rangle, f)) = \{x,y\}$.

(41) For every set f and for all non pair sets x, y holds $\mathrm{InputVertices}(\mathrm{1GateCircStr}(\langle x,y\rangle, f))$ has no pairs.

(42) For every set f and for all sets x, y, z holds $\mathrm{InputVertices}(\mathrm{1GateCircStr}(\langle x,y,z\rangle, f)) = \{x,y,z\}$.

(43) Let x, y, f be sets. Then $x \in$ the carrier of $\mathrm{1GateCircStr}(\langle x,y\rangle, f)$ and $y \in$ the carrier of $\mathrm{1GateCircStr}(\langle x,y\rangle, f)$ and $\langle\langle x,y\rangle, f\rangle \in$ the carrier of $\mathrm{1GateCircStr}(\langle x,y\rangle, f)$.

(44) Let x, y, z, f be sets. Then $x \in$ the carrier of $\mathrm{1GateCircStr}(\langle x,y,z\rangle, f)$ and $y \in$ the carrier of $\mathrm{1GateCircStr}(\langle x,y,z\rangle, f)$ and $z \in$ the carrier of $\mathrm{1GateCircStr}(\langle x,y,z\rangle, f)$.

(45) Let f, x be sets and p be a finite sequence. Then $x \in$ the carrier of $\mathrm{1GateCircStr}(p, f, x)$ and for every set y such that $y \in \mathrm{rng}\, p$ holds $y \in$ the carrier of $\mathrm{1GateCircStr}(p, f, x)$.

(46) For all sets f, x and for every finite sequence p holds $\mathrm{1GateCircStr}(p, f, x)$ is circuit-like and has arity held in gates.

(47) For every finite sequence p and for every set f holds $\langle p, f\rangle \in \mathrm{InnerVertices}(\mathrm{1GateCircStr}(p, f))$.

Let x, y be sets and let f be a function from $\mathit{Boolean}^2$ into $\mathit{Boolean}$. The functor $\mathrm{1GateCircuit}(x,y,f)$ yields a Boolean strict circuit of $\mathrm{1GateCircStr}(\langle x,y\rangle, f)$ with denotation held in gates and is defined by:

(Def. 10) $\mathrm{1GateCircuit}(x,y,f) = \mathrm{1GateCircuit}(\langle x,y\rangle, f)$.

We use the following convention: x, y, z, c denote sets and f denotes a function from $Boolean^2$ into *Boolean*.

Next we state four propositions:

(48) Let X be a finite non empty set, f be a function from X^2 into X, and s be a state of 1GateCircuit($\langle x,y \rangle, f$). Then (Following(s))($\langle\langle x,y\rangle, f\rangle$) $= f(\langle s(x), s(y)\rangle)$ and (Following(s))(x) $= s(x)$ and (Following(s))(y) $= s(y)$.

(49) Let X be a finite non empty set, f be a function from X^2 into X, and s be a state of 1GateCircuit($\langle x,y\rangle, f$). Then Following(s) is stable.

(50) For every state s of 1GateCircuit(x,y,f) holds (Following(s))($\langle\langle x,y\rangle, f\rangle$) $= f(\langle s(x),s(y)\rangle)$ and (Following(s))(x) $= s(x)$ and (Following(s))(y) $= s(y)$.

(51) For every state s of 1GateCircuit(x,y,f) holds Following(s) is stable.

Let x, y, z be sets and let f be a function from $Boolean^3$ into *Boolean*. The functor 1GateCircuit(x,y,z,f) yields a Boolean strict circuit of 1GateCircStr($\langle x,y,z\rangle, f$) with denotation held in gates and is defined as follows:

(Def. 11) 1GateCircuit(x,y,z,f) = 1GateCircuit($\langle x,y,z\rangle, f$).

Next we state four propositions:

(52) Let X be a finite non empty set, f be a function from X^3 into X, and s be a state of 1GateCircuit($\langle x,y,z\rangle, f$). Then (Following(s))($\langle\langle x,y,z\rangle, f\rangle$) $= f(\langle s(x),s(y),s(z)\rangle)$ and (Following(s))(x) $= s(x)$ and (Following(s))(y) $= s(y)$ and (Following(s))(z) $= s(z)$.

(53) Let X be a finite non empty set, f be a function from X^3 into X, and s be a state of 1GateCircuit($\langle x,y,z\rangle, f$). Then Following(s) is stable.

(54) Let f be a function from $Boolean^3$ into *Boolean* and s be a state of 1GateCircuit(x,y,z,f). Then (Following(s))($\langle\langle x,y,z\rangle, f\rangle$) $= f(\langle s(x),s(y),s(z)\rangle)$ and (Following(s))(x) $= s(x)$ and (Following(s))(y) $= s(y)$ and (Following(s))(z) $= s(z)$.

(55) For every function f from $Boolean^3$ into *Boolean* and for every state s of 1GateCircuit(x,y,z,f) holds Following(s) is stable.

6. Boolean Circuits

Let x, y, c be sets and let f be a function from $Boolean^2$ into *Boolean*. The functor 2GatesCircStr(x,y,c,f) 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. 12) 2GatesCircStr(x,y,c,f) = 1GateCircStr($\langle x,y\rangle, f$)+· 1GateCircStr($\langle\langle\langle x,y\rangle, f\rangle, c\rangle, f$).

Let x, y, c be sets and let f be a function from $Boolean^2$ into *Boolean*. The functor 2GatesCircOutput(x,y,c,f) yields an element of InnerVertices(2GatesCircStr(x,y,c,f)) and is defined by:

(Def. 13) 2GatesCircOutput(x,y,c,f) = $\langle\langle\langle\langle x,y\rangle, f\rangle, c\rangle, f\rangle$.

Let x, y, c be sets and let f be a function from $Boolean^2$ into *Boolean*. Note that 2GatesCircOutput(x,y,c,f) is pair.

Next we state two propositions:

(56) InnerVertices(2GatesCircStr(x,y,c,f)) = $\{\langle\langle x,y\rangle, f\rangle$, 2GatesCircOutput($x,y,c,f$)$\}$.

(57) If $c \neq \langle\langle x,y\rangle, f\rangle$, then InputVertices(2GatesCircStr(x,y,c,f)) = $\{x,y,c\}$.

Let x, y, c be sets and let f be a function from *Boolean*^{2} into *Boolean*. The functor 2GatesCircuit(x,y,c,f) yielding a strict Boolean circuit of 2GatesCircStr(x,y,c,f) with denotation held in gates is defined as follows:

(Def. 14) 2GatesCircuit$(x,y,c,f) =$ 1GateCircuit$(x,y,f)+\cdot$ 1GateCircuit$(\langle\langle x,y\rangle, f\rangle, c, f)$.

One can prove the following four propositions:

(58) InnerVertices(2GatesCircStr(x,y,c,f)) is a binary relation.

(59) For all non pair sets x, y, c holds InputVertices(2GatesCircStr(x,y,c,f)) has no pairs.

(60) $x \in$ the carrier of 2GatesCircStr(x,y,c,f) and $y \in$ the carrier of 2GatesCircStr(x,y,c,f) and $c \in$ the carrier of 2GatesCircStr(x,y,c,f).

(61) $\langle\langle x,y\rangle, f\rangle \in$ the carrier of 2GatesCircStr(x,y,c,f) and $\langle\langle\langle\langle x,y\rangle, f\rangle,c\rangle, f\rangle \in$ the carrier of 2GatesCircStr(x,y,c,f).

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*.

In the sequel s is a state of 2GatesCircuit(x,y,c,f).

We now state several propositions:

(62) Suppose $c \neq \langle\langle x,y\rangle, f\rangle$. Then (Following$(s,2)$)(2GatesCircOutput$(x,y,c,f)$) $= f(\langle f(\langle s(x), s(y)\rangle), s(c)\rangle)$ and (Following$(s,2)$)$(\langle\langle x,y\rangle, f\rangle) = f(\langle s(x), s(y)\rangle)$ and (Following$(s,2)$)$(x) = s(x)$ and (Following$(s,2)$)$(y) = s(y)$ and (Following$(s,2)$)$(c) = s(c)$.

(63) If $c \neq \langle\langle x,y\rangle, f\rangle$, then Following$(s,2)$ is stable.

(64) Suppose $c \neq \langle\langle x,y\rangle, \mathrm{xor}\rangle$. Let s be a state of 2GatesCircuit(x,y,c,xor) and a_1, a_2, a_3 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(y)$ and $a_3 = s(c)$, then (Following$(s,2)$)(2GatesCircOutput(x,y,c,xor)) $= a_1 \oplus a_2 \oplus a_3$.

(65) Suppose $c \neq \langle\langle x,y\rangle, \mathrm{or}\rangle$. Let s be a state of 2GatesCircuit(x,y,c,or) and a_1, a_2, a_3 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(y)$ and $a_3 = s(c)$, then (Following$(s,2)$)(2GatesCircOutput(x,y,c,or)) $= a_1 \vee a_2 \vee a_3$.

(66) Suppose $c \neq \langle\langle x,y\rangle, \&\rangle$. Let s be a state of 2GatesCircuit$(x,y,c,\&)$ and a_1, a_2, a_3 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(y)$ and $a_3 = s(c)$, then (Following$(s,2)$)(2GatesCircOutput$(x,y,c,\&)$) $= a_1 \wedge a_2 \wedge a_3$.

7. One Bit Adder

Let x, y, c be sets. The functor BitAdderOutput(x,y,c) yielding an element of InnerVertices(2GatesCircStr(x,y,c,xor)) is defined by:

(Def. 15) BitAdderOutput$(x,y,c) =$ 2GatesCircOutput(x,y,c,xor).

Let x, y, c be sets. The functor BitAdderCirc(x,y,c) yielding a strict Boolean circuit of 2GatesCircStr(x,y,c,xor) with denotation held in gates is defined by:

(Def. 16) BitAdderCirc$(x,y,c) =$ 2GatesCircuit(x,y,c,xor).

Let x, y, c be sets. The functor MajorityIStr(x,y,c) 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 as follows:

(Def. 17) MajorityIStr$(x,y,c) =$ 1GateCircStr$(\langle x,y\rangle, \&)+\cdot$ 1GateCircStr$(\langle y,c\rangle, \&)+\cdot$ 1GateCircStr$(\langle c, x\rangle, \&)$.

Let x, y, c be sets. The functor MajorityStr(x,y,c) 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 as follows:

(Def. 18) $\text{MajorityStr}(x,y,c) = \text{MajorityIStr}(x,y,c) +\cdot \text{1GateCircStr}(\langle\langle\langle x,y\rangle, \&\rangle, \langle\langle y,c\rangle, \&\rangle, \langle\langle c,x\rangle, \&\rangle\rangle, \text{or}_3)$.

Let x, y, c be sets. The functor $\text{MajorityICirc}(x,y,c)$ yields a strict Boolean circuit of $\text{MajorityIStr}(x,y,c)$ with denotation held in gates and is defined as follows:

(Def. 19) $\text{MajorityICirc}(x,y,c) = \text{1GateCircuit}(x,y,\&) +\cdot \text{1GateCircuit}(y,c,\&) +\cdot \text{1GateCircuit}(c,x,\&)$.

One can prove the following propositions:

(67) $\text{InnerVertices}(\text{MajorityStr}(x,y,c))$ is a binary relation.

(68) For all non pair sets x, y, c holds $\text{InputVertices}(\text{MajorityStr}(x,y,c))$ has no pairs.

(69) For every state s of $\text{MajorityICirc}(x,y,c)$ and for all elements a, b of *Boolean* such that $a = s(x)$ and $b = s(y)$ holds $(\text{Following}(s))(\langle\langle x,y\rangle, \&\rangle) = a \wedge b$.

(70) For every state s of $\text{MajorityICirc}(x,y,c)$ and for all elements a, b of *Boolean* such that $a = s(y)$ and $b = s(c)$ holds $(\text{Following}(s))(\langle\langle y,c\rangle, \&\rangle) = a \wedge b$.

(71) For every state s of $\text{MajorityICirc}(x,y,c)$ and for all elements a, b of *Boolean* such that $a = s(c)$ and $b = s(x)$ holds $(\text{Following}(s))(\langle\langle c,x\rangle, \&\rangle) = a \wedge b$.

Let x, y, c be sets. The functor $\text{MajorityOutput}(x,y,c)$ yields an element of $\text{InnerVertices}(\text{MajorityStr}(x,y,c))$ and is defined by:

(Def. 20) $\text{MajorityOutput}(x,y,c) = \langle\langle\langle\langle x,y\rangle, \&\rangle, \langle\langle y,c\rangle, \&\rangle, \langle\langle c,x\rangle, \&\rangle\rangle, \text{or}_3\rangle$.

Let x, y, c be sets. The functor $\text{MajorityCirc}(x,y,c)$ yields a strict Boolean circuit of $\text{MajorityStr}(x,y,c)$ with denotation held in gates and is defined as follows:

(Def. 21) $\text{MajorityCirc}(x,y,c) = \text{MajorityICirc}(x,y,c) +\cdot \text{1GateCircuit}(\langle\langle x,y\rangle, \&\rangle, \langle\langle y,c\rangle, \&\rangle, \langle\langle c,x\rangle, \&\rangle, \text{or}_3)$.

We now state a number of propositions:

(72) $x \in$ the carrier of $\text{MajorityStr}(x,y,c)$ and $y \in$ the carrier of $\text{MajorityStr}(x,y,c)$ and $c \in$ the carrier of $\text{MajorityStr}(x,y,c)$.

(73) $\langle\langle x,y\rangle, \&\rangle \in \text{InnerVertices}(\text{MajorityStr}(x,y,c))$ and $\langle\langle y,c\rangle, \&\rangle \in \text{InnerVertices}(\text{MajorityStr}(x,y,c))$ and $\langle\langle c,x\rangle, \&\rangle \in \text{InnerVertices}(\text{MajorityStr}(x,y,c))$.

(74) For all non pair sets x, y, c holds $x \in \text{InputVertices}(\text{MajorityStr}(x,y,c))$ and $y \in \text{InputVertices}(\text{MajorityStr}(x,y,c))$ and $c \in \text{InputVertices}(\text{MajorityStr}(x,y,c))$.

(75) For all non pair sets x, y, c holds $\text{InputVertices}(\text{MajorityStr}(x,y,c)) = \{x,y,c\}$ and $\text{InnerVertices}(\text{MajorityStr}(x,y,c)) = \{\langle\langle x,y\rangle, \&\rangle, \langle\langle y,c\rangle, \&\rangle, \langle\langle c,x\rangle, \&\rangle\} \cup \{\text{MajorityOutput}(x,y,c)\}$.

(76) Let x, y, c be non pair sets, s be a state of $\text{MajorityCirc}(x,y,c)$, and a_1, a_2 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(y)$, then $(\text{Following}(s))(\langle\langle x,y\rangle, \&\rangle) = a_1 \wedge a_2$.

(77) Let x, y, c be non pair sets, s be a state of $\text{MajorityCirc}(x,y,c)$, and a_2, a_3 be elements of *Boolean*. If $a_2 = s(y)$ and $a_3 = s(c)$, then $(\text{Following}(s))(\langle\langle y,c\rangle, \&\rangle) = a_2 \wedge a_3$.

(78) Let x, y, c be non pair sets, s be a state of $\text{MajorityCirc}(x,y,c)$, and a_1, a_3 be elements of *Boolean*. If $a_1 = s(x)$ and $a_3 = s(c)$, then $(\text{Following}(s))(\langle\langle c,x\rangle, \&\rangle) = a_3 \wedge a_1$.

(79) Let x, y, c be non pair sets, s be a state of $\text{MajorityCirc}(x,y,c)$, and a_1, a_2, a_3 be elements of *Boolean*. If $a_1 = s(\langle\langle x,y\rangle, \&\rangle)$ and $a_2 = s(\langle\langle y,c\rangle, \&\rangle)$ and $a_3 = s(\langle\langle c,x\rangle, \&\rangle)$, then $(\text{Following}(s))(\text{MajorityOutput}(x,y,c)) = a_1 \vee a_2 \vee a_3$.

(80) Let x, y, c be non pair sets, s be a state of $\text{MajorityCirc}(x,y,c)$, and a_1, a_2 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(y)$, then $(\text{Following}(s,2))(\langle\langle x,y\rangle, \&\rangle) = a_1 \wedge a_2$.

(81) Let x, y, c be non pair sets, s be a state of MajorityCirc(x,y,c), and a_2, a_3 be elements of *Boolean*. If $a_2 = s(y)$ and $a_3 = s(c)$, then $(\mathrm{Following}(s,2))(\langle\langle y,c\rangle, \&\rangle) = a_2 \wedge a_3$.

(82) Let x, y, c be non pair sets, s be a state of MajorityCirc(x,y,c), and a_1, a_3 be elements of *Boolean*. If $a_1 = s(x)$ and $a_3 = s(c)$, then $(\mathrm{Following}(s,2))(\langle\langle c,x\rangle, \&\rangle) = a_3 \wedge a_1$.

(83) Let x, y, c be non pair sets, s be a state of MajorityCirc(x,y,c), and a_1, a_2, a_3 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(y)$ and $a_3 = s(c)$, then $(\mathrm{Following}(s,2))(\mathrm{MajorityOutput}(x,y,c)) = a_1 \wedge a_2 \vee a_2 \wedge a_3 \vee a_3 \wedge a_1$.

(84) For all non pair sets x, y, c and for every state s of MajorityCirc(x,y,c) holds Following$(s,2)$ is stable.

Let x, y, c be sets. The functor BitAdderWithOverflowStr(x,y,c) 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 as follows:

(Def. 22) $\mathrm{BitAdderWithOverflowStr}(x,y,c) = \mathrm{2GatesCircStr}(x,y,c,\mathrm{xor}) +\cdot \mathrm{MajorityStr}(x,y,c)$.

One can prove the following three propositions:

(85) For all non pair sets x, y, c holds InputVertices(BitAdderWithOverflowStr$(x,y,c)) = \{x,y,c\}$.

(86) For all non pair sets x, y, c holds InnerVertices(BitAdderWithOverflowStr$(x,y,c)) = \{\langle\langle x, y\rangle, \mathrm{xor}\rangle, \mathrm{2GatesCircOutput}(x,y,c,\mathrm{xor})\} \cup \{\langle\langle x,y\rangle, \&\rangle, \langle\langle y,c\rangle, \&\rangle, \langle\langle c,x\rangle, \&\rangle\} \cup \{\mathrm{MajorityOutput}(x,y,c)\}$.

(87) Let S be a non empty many sorted signature. Suppose $S = \mathrm{BitAdderWithOverflowStr}(x,y,c)$. Then $x \in$ the carrier of S and $y \in$ the carrier of S and $c \in$ the carrier of S.

Let x, y, c be sets. The functor BitAdderWithOverflowCirc(x,y,c) yielding a strict Boolean circuit of BitAdderWithOverflowStr(x,y,c) with denotation held in gates is defined by:

(Def. 23) $\mathrm{BitAdderWithOverflowCirc}(x,y,c) = \mathrm{BitAdderCirc}(x,y,c) +\cdot \mathrm{MajorityCirc}(x,y,c)$.

One can prove the following propositions:

(88) InnerVertices(BitAdderWithOverflowStr(x,y,c)) is a binary relation.

(89) For all non pair sets x, y, c holds InputVertices(BitAdderWithOverflowStr(x,y,c)) has no pairs.

(90) $\mathrm{BitAdderOutput}(x,y,c) \in$ InnerVertices(BitAdderWithOverflowStr(x,y,c)) and $\mathrm{MajorityOutput}(x,y,c) \in$ InnerVertices(BitAdderWithOverflowStr(x,y,c)).

(91) Let x, y, c be non pair sets, s be a state of BitAdderWithOverflowCirc(x,y,c), and a_1, a_2, a_3 be elements of *Boolean*. Suppose $a_1 = s(x)$ and $a_2 = s(y)$ and $a_3 = s(c)$. Then $(\mathrm{Following}(s,2))(\mathrm{BitAdderOutput}(x,y,c)) = a_1 \oplus a_2 \oplus a_3$ and $(\mathrm{Following}(s,2))(\mathrm{MajorityOutput}(x,y,c)) = a_1 \wedge a_2 \vee a_2 \wedge a_3 \vee a_3 \wedge a_1$.

(92) For all non pair sets x, y, c and for every state s of BitAdderWithOverflowCirc(x,y,c) holds Following$(s,2)$ is stable.

REFERENCES

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/nat_1.html.

[2] Grzegorz Bancerek. König's theorem. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/card_3.html.

[3] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/finseq_1.html.

[4] Czesław Byliński. Functions and their basic properties. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/funct_1.html.

[5] Czesław Byliński. Functions from a set to a set. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/funct_2.html.

[6] Czesław Byliński. Partial functions. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/partfun1.html.

[7] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/finseq_2.html.

[8] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/funct_4.html.

[9] Agata Darmochwał. Finite sets. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/finset_1.html.

[10] Yatsuka Nakamura and Grzegorz Bancerek. Combining of circuits. *Journal of Formalized Mathematics*, 7, 1995. http://mizar.org/JFM/Vol7/circcomb.html.

[11] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Introduction to circuits, I. *Journal of Formalized Mathematics*, 6, 1994. http://mizar.org/JFM/Vol6/circuit1.html.

[12] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, I. *Journal of Formalized Mathematics*, 6, 1994. http://mizar.org/JFM/Vol6/pre_circ.html.

[13] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, II. *Journal of Formalized Mathematics*, 6, 1994. http://mizar.org/JFM/Vol6/msafree2.html.

[14] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Introduction to circuits, II. *Journal of Formalized Mathematics*, 7, 1995. http://mizar.org/JFM/Vol7/circuit2.html.

[15] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Journal of Formalized Mathematics*, 5, 1993. http://mizar.org/JFM/Vol5/binarith.html.

[16] Andrzej Trybulec. Enumerated sets. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/enumset1.html.

[17] Andrzej Trybulec. Tarski Grothendieck set theory. *Journal of Formalized Mathematics*, Axiomatics, 1989. http://mizar.org/JFM/Axiomatics/tarski.html.

[18] Andrzej Trybulec. Tuples, projections and Cartesian products. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/mcart_1.html.

[19] Andrzej Trybulec. Many sorted algebras. *Journal of Formalized Mathematics*, 6, 1994. http://mizar.org/JFM/Vol6/msualg_1.html.

[20] Andrzej Trybulec. Subsets of real numbers. *Journal of Formalized Mathematics*, Addenda, 2003. http://mizar.org/JFM/Addenda/numbers.html.

[21] Zinaida Trybulec. Properties of subsets. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/subset_1.html.

[22] Edmund Woronowicz. Relations and their basic properties. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/relat_1.html.

[23] Edmund Woronowicz. Many-argument relations. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/margrel1.html.

Received August 10, 1995

Published January 2, 2004