FORMALIZED MATHEMATICS
Volume 5, Number 3, 1996
Warsaw University - 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 [16,17,14,15]. 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 [13].

MML Identifier: `FACIRC_1`.

The terminology and notation used in this paper are introduced in the following papers: [23], [25], [20], [1], [24], [27], [7], [8], [5], [11], [6], [19], [9], [26], [18], [3], [2], [4], [10], [12], [22], [21], [16], [17], [14], [15], and [13].

1. Combining of Many Sorted Signatures

A set is pair if:

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

Let us mention that every set which is pair is also non empty.

Let x, y be sets. Observe that $\langle x, y \rangle$ is pair.

Let us mention 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.

A set has a pair if:

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

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

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

ISSN 1426–2630

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

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

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

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

One can verify that every set which is empty is also relation-like. Let x be a pair set. One can check that $\{x\}$ is relation-like. Let y be a pair set. Observe that $\{x, y\}$ is relation-like. Let z be a pair set. One can check that $\{x, y, z\}$ is relation-like.

Let us note that every set which is relation-like and has no pairs is also empty.

A function is nonpair yielding if:

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

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

One can prove the following 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.

One can check that there exists a finite sequence which is one-to-one and nonpair yielding.

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

The following propositions are true:

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

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

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

(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 $\mathrm{InputVertices}(S_1)$ has no pairs and $\mathrm{InnerVertices}(S_2)$ is a binary relation. Then $\mathrm{InputVertices}(S_1) \subseteq \mathrm{InputVertices}(S_1 {+\cdot} S_2)$

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. Combinig of Circuits

In this article we present several logical schemes. The scheme *2AryBooleDef* concerns 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 *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 by:

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

The function & from $Boolean^2$ into *Boolean* is defined as follows:

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

The function or_3 from $Boolean^3$ into *Boolean* is defined by:

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

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

The following proposition is true

(10) Let S be a circuit-like non void non empty many sorted signature, and let A be a non-empty circuit of S, and let s be a state of A, and let g be a gate of S. Then (Following(s))(the result sort of g) = (Den(g, A))($s \cdot$ 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 Following(s, n) yielding a state of A 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 Following(s, n) $= f(n)$ and $f(0) = s$ and for every natural number n and for every state x of A such that $x = f(n)$ holds $f(n+1) =$ Following(x).

The following propositions are true:

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

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

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

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

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

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

The following propositions are true:

(20) Let S_1, S_2 be non empty many sorted signatures and let 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 let 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, and let A_1 be a Boolean circuit of S_1 with denotation held in gates, and let 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, and let A_1 be a Boolean circuit of S_1, and let A_2 be a Boolean circuit of S_2, and let 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, and let A_1 be a Boolean non-empty circuit of S_1 with denotation held in gates, and let

A_2 be a Boolean non-empty circuit of S_2 with denotation held in gates, and let 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 InnerVertices($S_1 +\cdot S_2$) = InnerVertices(S_1) $\cup$ 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 InnerVertices(S_2) misses InputVertices(S_1). Let A_1 be a Boolean circuit of S_1 with denotation held in gates, and let A_2 be a Boolean circuit of S_2 with denotation held in gates, and let s be a state of $A_1 +\cdot A_2$, and let s_1 be a state of A_1. If $s_1 = s \restriction$ (the carrier of S_1), then Following(s) $\restriction$ (the carrier of S_1) = 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 InnerVertices(S_1) misses InputVertices(S_2). Let A_1 be a Boolean circuit of S_1 with denotation held in gates, and let A_2 be a Boolean circuit of S_2 with denotation held in gates, and let s be a state of $A_1 +\cdot A_2$, and let s_2 be a state of A_2. If $s_2 = s \restriction$ (the carrier of S_2), then Following(s) $\restriction$ (the carrier of S_2) = 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 InnerVertices(S_2) misses InputVertices(S_1). Let A_1 be a Boolean circuit of S_1 with denotation held in gates, and let A_2 be a Boolean circuit of S_2 with denotation held in gates, and let s be a state of $A_1 +\cdot A_2$, and let 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 Following(s, n) $\restriction$ (the carrier of S_1) = 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 InnerVertices(S_1) misses InputVertices(S_2). Let A_1 be a Boolean circuit of S_1 with denotation held in gates, and let A_2 be a Boolean circuit of S_2 with denotation held in gates, and let s be a state of $A_1 +\cdot A_2$, and let 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 Following(s, n) $\restriction$ (the carrier of S_2) = 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 InnerVertices(S_2) misses InputVertices(S_1). Let A_1 be a Boolean circuit of S_1 with denotation held in gates, and let A_2 be a Boolean circuit of S_2 with denotation held in gates, and let s be a state of $A_1 +\cdot A_2$, and let 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 (Following(s, n))(v) = (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

InnerVertices(S_1) misses InputVertices(S_2). Let A_1 be a Boolean circuit of S_1 with denotation held in gates, and let A_2 be a Boolean circuit of S_2 with denotation held in gates, and let s be a state of $A_1{+}{\cdot}A_2$, and let 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 (Following(s, n))(v) = (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.

Next we state four propositions:

(34) Let S be a circuit-like non void non empty many sorted signature with denotation held in gates and let A be a non-empty circuit of S. Suppose A has denotation held in gates. Let s be a state of A and let g be a gate of S. Then (Following(s))(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, and let A be a Boolean non-empty circuit of S with denotation held in gates, and let s be a state of A, and let p be a finite sequence, and let f be a function. If $\langle p, f\rangle \in$ the operation symbols of S, then (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, and let A be a Boolean non-empty circuit of S with denotation held in gates, and let s be a state of A, and let p be a finite sequence, and let 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 Following(s) is stable at $\langle p, f\rangle$.

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

5. One Gate Circuits

We now state a number of propositions:

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

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

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

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

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

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

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

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

(46) For all sets f, x and for every finite sequence p holds 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 $\langle p, f\rangle \in$ InnerVertices(1GateCircStr(p, f)).

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

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

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

We now state four propositions:

(48) Let X be a finite non empty set, and let f be a function from X^2 into X, and let 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, and let f be a function from X^2 into X, and let 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 by:

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

We now state four propositions:

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

(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 as follows:

(Def.12) 2GatesCircStr$(x, y, c, f) =$ 1GateCircStr$(\langle x, y\rangle, f) +\cdot$ 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 as follows:

(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*. One can verify that 2GatesCircOutput(x, y, c, f) is pair.

One can prove the following 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) yields a strict Boolean circuit of 2GatesCircStr(x, y, c, f) with denotation held in gates and is defined by:

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

We now state 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 will be a state of 2GatesCircuit(x, y, c, f).

One can prove the following propositions:

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

(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 let 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 $(\mathrm{Following}(s, 2))(\mathrm{2GatesCircOutput}(x, y, c, \mathrm{xor})) = a_1 \oplus a_2 \oplus a_3$.

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

(66) Suppose $c \neq \langle\langle x, y\rangle, \&\rangle$. Let s be a state of 2GatesCircuit$(x, y, c, \&)$ and let 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 $(\mathrm{Following}(s, 2))(\mathrm{2GatesCircOutput}(x, y, c, \&)) = a_1 \wedge a_2 \wedge a_3$.

7. One Bit Adder

Let x, y, c be sets. The functor BitAdderOutput(x, y, c) yields an element of InnerVertices(2GatesCircStr(x, y, c, xor)) and is defined as follows:

(Def.15) $\mathrm{BitAdderOutput}(x, y, c) = \mathrm{2GatesCircOutput}(x, y, c, \mathrm{xor})$.

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

(Def.16) $\mathrm{BitAdderCirc}(x, y, c) = \mathrm{2GatesCircuit}(x, y, c, \mathrm{xor})$.

Let x, y, c be sets. The functor MajorityIStr(x, y, c) 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.17) $\mathrm{MajorityIStr}(x, y, c) = \mathrm{1GateCircStr}(\langle x, y\rangle, \&) +\cdot \mathrm{1GateCircStr}(\langle y, c\rangle, \&) +\cdot \mathrm{1GateCircStr}(\langle c, x\rangle, \&)$.

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

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

(Def.19) MajorityICirc(x, y, c) = 1GateCircuit$(x, y, \&)$+· 1GateCircuit$(y, c, \&)$ +· 1GateCircuit$(c, x, \&)$.

Next we state several propositions:

(67) InnerVertices(MajorityStr(x, y, c)) is a binary relation.

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

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

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

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

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

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

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

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

Next we state a number of propositions:

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

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

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

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

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

(77) Let x, y, c be non pair sets, and let s be a state of MajorityCirc(x, y, c), and let a_2, a_3 be elements of *Boolean*. If $a_2 = s(y)$ and $a_3 = s(c)$, then

(Following(s))($\langle\!\langle\langle y, c\rangle, \&\rangle\!\rangle$) $= a_2 \wedge a_3$.

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

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

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

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

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

(83) Let x, y, c be non pair sets, and let s be a state of MajorityCirc(x, y, c), and let 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($s, 2$))(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) BitAdderWithOverflowStr(x, y, c) = 2GatesCircStr(x, y, c, xor) $+\cdot$ MajorityStr(x, y, c).

The following three propositions are true:

(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)) $= \{\langle\!\langle\langle x, y\rangle, \text{xor}\rangle\!\rangle, \text{2GatesCircOutput}(x, y, c, \text{xor})\} \cup \{\langle\!\langle\langle x, y\rangle, \&\rangle\!\rangle, \langle\!\langle\langle y, c\rangle, \&\rangle\!\rangle, \langle\!\langle\langle c, x\rangle, \&\rangle\!\rangle\} \cup \{\text{MajorityOutput}(x, y, c)\}$.

(87) Let S be a non empty many sorted signature. Suppose S = 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 as follows:

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

We now state several 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) BitAdderOutput(x, y, c) $\in$ InnerVertices(BitAdderWithOverflowStr(x, y, c)) and MajorityOutput(x, y, c) $\in$ InnerVertices(BitAdderWithOverflow Str(x, y, c)).

(91) Let x, y, c be non pair sets, and let s be a state of BitAdderWithOverflowCirc(x, y, c), and let 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 (Following($s, 2$))(BitAdderOutput(x, y, c)) $= a_1 \oplus a_2 \oplus a_3$ and (Following($s, 2$))(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. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[2] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.
[3] Grzegorz Bancerek. The ordinal numbers. *Formalized Mathematics*, 1(**1**):91–96, 1990.
[4] Grzegorz Bancerek. Tarski's classes and ranks. *Formalized Mathematics*, 1(**3**):563–567, 1990.
[5] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[6] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(**3**):529–536, 1990.
[7] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[8] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[9] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.
[10] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.
[11] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[12] Anna Lango and Grzegorz Bancerek. Product of families of groups and vector spaces. *Formalized Mathematics*, 3(**2**):235–240, 1992.
[13] Yatsuka Nakamura and Grzegorz Bancerek. Combining of circuits. *Formalized Mathematics*, 5(**2**):283–295, 1996.
[14] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Introduction to circuits, I. *Formalized Mathematics*, 5(**2**):227–232, 1996.
[15] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Introduction to circuits, II. *Formalized Mathematics*, 5(**2**):273–278, 1996.
[16] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, I. *Formalized Mathematics*, 5(**2**):167–172, 1996.
[17] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, II. *Formalized Mathematics*, 5(**2**):215–220, 1996.
[18] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[19] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[20] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.

[21] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[22] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[23] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[24] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[25] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[26] Edmund Woronowicz. Many–argument relations. *Formalized Mathematics*, 1(**4**):733–737, 1990.
[27] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received August 10, 1995