FORMALIZED MATHEMATICS
Volume 5, Number 2, 1996
Warsaw University - Białystok

Introduction to Circuits, II [1]

Yatsuka Nakamura
Shinshu University, Nagano

Piotr Rudnicki
University of Alberta, Edmonton

Andrzej Trybulec
Warsaw University, Białystok

Pauline N. Kawamoto
Shinshu University, Nagano

Summary. This article is the last in a series of four articles (preceded by [23,22,21]) about modelling circuits by many sorted algebras.

The notion of a circuit computation is defined as a sequence of circuit states. For a state of a circuit the next state is given by executing operations at circuit vertices in the current state, according to denotations of the operations. The values at input vertices at each state of a computation are provided by an external sequence of input values. The process of how input values propagate through a circuit is described in terms of a homomorphism of the free envelope algebra of the circuit into itself. We prove that every computation of a circuit over a finite monotonic signature and with constant input values stabilizes after executing the number of steps equal to the depth of the circuit.

MML Identifier: `CIRCUIT2`.

The articles [27], [30], [31], [12], [13], [18], [14], [3], [9], [16], [5], [7], [4], [28], [1], [6], [29], [2], [15], [10], [26], [19], [25], [11], [20], [17], [24], [23], [22], [21], and [8] provide the terminology and notation for this paper.

1. Circuit Inputs

In this paper I_1 will be a monotonic circuit-like non void non empty many sorted signature.

The following proposition is true

[1]Partial funding for this work has been provided by: Shinshu Endowment Fund for Information Science, NSERC Grant OGP9207, JSTF award 651-93-S009.

ISSN 1426–2630

(1) Let X be a non-empty many sorted set indexed by the carrier of I_1, and let H be a many sorted function from Free(X) into Free(X), and let H_1 be a function yielding function, and let v be a sort symbol of I_1, and let p be a decorated tree yielding finite sequence, and let t be an element of (the sorts of Free(X))(v). Suppose that
 (i) $v \in$ InnerVertices(I_1),
 (ii) $t = \langle$the action at v, the carrier of $I_1\rangle$-tree(p),
 (iii) H is a homomorphism of Free(X) into Free(X), and
 (iv) $H_1 = H \cdot$ Arity(the action at v).

Then there exists a decorated tree yielding finite sequence H_2 such that $H_2 = H_1 \looparrowleft p$ and $H(v)(t) = \langle$the action at v, the carrier of $I_1\rangle$-tree(H_2).

Let us consider I_1, let S_1 be a non-empty circuit of I_1, let s be a state of S_1, and let i_1 be an input assignment of S_1. Then $s +\cdot i_1$ is a state of S_1.

Let us consider I_1, let A be a non-empty circuit of I_1, and let i_1 be an input assignment of A. The functor FixInput(i_1) yields a many sorted function from FreeGenerator(the sorts of A) into the sorts of FreeEnvelope(A) and is defined by the condition (Def.1).

(Def.1) Let v be a vertex of I_1. Then
 (i) if $v \in$ InputVertices(I_1), then (FixInput(i_1))(v) = FreeGenerator(v, the sorts of A) $\longmapsto$ the root tree of $\langle i_1(v), v\rangle$,
 (ii) if $v \in$ SortsWithConstants(I_1), then (FixInput(i_1))(v) = FreeGenerator(v, the sorts of A) $\longmapsto$ the root tree of $\langle$the action at v, the carrier of $I_1\rangle$, and
 (iii) if $v \in$ InnerVertices(I_1) $\setminus$ SortsWithConstants(I_1), then (FixInput(i_1))(v) = $\mathrm{id}_{\mathrm{FreeGenerator}(v, \text{the sorts of } A)}$.

Let us consider I_1, let A be a non-empty circuit of I_1, and let i_1 be an input assignment of A. The functor FixInputExt(i_1) yields a many sorted function from FreeEnvelope(A) into FreeEnvelope(A) and is defined by:

(Def.2) FixInputExt(i_1) is a homomorphism of FreeEnvelope(A) into FreeEnvelope(A) and FixInput(i_1) $\subseteq$ FixInputExt(i_1).

The following propositions are true:

(2) Let A be a non-empty circuit of I_1, and let i_1 be an input assignment of A, and let v be a vertex of I_1, and let e be an element of (the sorts of FreeEnvelope(A))(v), and let x be arbitrary. If $v \in$ InnerVertices(I_1) $\setminus$ SortsWithConstants(I_1) and e = the root tree of $\langle x, v\rangle$, then (FixInputExt(i_1))(v)(e) = e.

(3) Let A be a non-empty circuit of I_1, and let i_1 be an input assignment of A, and let v be a vertex of I_1, and let x be an element of (the sorts of A)(v). If $v \in$ InputVertices(I_1), then (FixInputExt(i_1))(v)(the root tree of $\langle x, v\rangle$) = the root tree of $\langle i_1(v), v\rangle$.

(4) Let A be a non-empty circuit of I_1, and let i_1 be an input assignment of A, and let v be a vertex of I_1, and let e be an element of (the sorts

of FreeEnvelope(A))(v), and let p, q be decorated tree yielding finite sequences. Suppose that

(i) $v \in \mathrm{InnerVertices}(I_1)$,

(ii) $e = \langle$the action at v, the carrier of $I_1\rangle$-tree(p),

(iii) $\mathrm{dom}\, p = \mathrm{dom}\, q$, and

(iv) for every natural number k such that $k \in \mathrm{dom}\, p$ holds $q(k) = (\mathrm{FixInputExt}(i_1))(\pi_k \,\mathrm{Arity}(\text{the action at } v))(p(k))$.

Then $(\mathrm{FixInputExt}(i_1))(v)(e) = \langle$the action at v, the carrier of $I_1\rangle$-tree(q).

(5) Let A be a non-empty circuit of I_1, and let i_1 be an input assignment of A, and let v be a vertex of I_1, and let e be an element of (the sorts of FreeEnvelope(A))(v). Suppose $v \in \mathrm{SortsWithConstants}(I_1)$. Then $(\mathrm{FixInputExt}(i_1))(v)(e) =$ the root tree of $\langle$the action at v, the carrier of $I_1\rangle$.

(6) Let A be a non-empty circuit of I_1, and let i_1 be an input assignment of A, and let v be a vertex of I_1, and let e, e_1 be elements of (the sorts of FreeEnvelope(A))(v), and let t, t_1 be decorated trees. If $t = e$ and $t_1 = e_1$ and $e_1 = (\mathrm{FixInputExt}(i_1))(v)(e)$, then $\mathrm{dom}\, t = \mathrm{dom}\, t_1$.

(7) Let A be a non-empty circuit of I_1, and let i_1 be an input assignment of A, and let v be a vertex of I_1, and let e, e_1 be elements of (the sorts of FreeEnvelope(A))(v). If $e_1 = (\mathrm{FixInputExt}(i_1))(v)(e)$, then $\mathrm{card}\, e = \mathrm{card}\, e_1$.

Let us consider I_1, let S_1 be a non-empty circuit of I_1, let v be a vertex of I_1, and let i_1 be an input assignment of S_1. The functor $\mathrm{InputGenTree}(v, i_1)$ yields an element of (the sorts of FreeEnvelope(S_1))(v) and is defined by:

(Def.3) There exists an element e of (the sorts of FreeEnvelope(S_1))(v) such that $\mathrm{card}\, e = \mathrm{size}(v, S_1)$ and $\mathrm{InputGenTree}(v, i_1) = (\mathrm{FixInputExt}(i_1))(v)(e)$.

We now state two propositions:

(8) Let S_1 be a non-empty circuit of I_1, and let v be a vertex of I_1, and let i_1 be an input assignment of S_1. Then $\mathrm{InputGenTree}(v, i_1) = (\mathrm{FixInputExt}(i_1))(v)(\mathrm{InputGenTree}(v, i_1))$.

(9) Let S_1 be a non-empty circuit of I_1, and let v be a vertex of I_1, and let i_1 be an input assignment of S_1, and let p be a decorated tree yielding finite sequence. Suppose that

(i) $v \in \mathrm{InnerVertices}(I_1)$,

(ii) $\mathrm{dom}\, p = \mathrm{dom}\, \mathrm{Arity}(\text{the action at } v)$, and

(iii) for every natural number k such that $k \in \mathrm{dom}\, p$ holds $p(k) = \mathrm{InputGenTree}(\pi_k \,\mathrm{Arity}(\text{the action at } v), i_1)$.

Then $\mathrm{InputGenTree}(v, i_1) = \langle$the action at v, the carrier of $I_1\rangle$-tree(p).

Let us consider I_1, let S_1 be a non-empty circuit of I_1, let v be a vertex of I_1, and let i_1 be an input assignment of S_1. The functor $\mathrm{InputGenValue}(v, i_1)$ yields an element of (the sorts of S_1)(v) and is defined by:

(Def.4) $\mathrm{InputGenValue}(v, i_1) = (\mathrm{Eval}(S_1))(v)(\mathrm{InputGenTree}(v, i_1))$.

The following propositions are true:

(10) Let S_1 be a non-empty circuit of I_1, and let v be a vertex of I_1, and let i_1 be an input assignment of S_1. If $v \in \text{InputVertices}(I_1)$, then $\text{InputGenValue}(v, i_1) = i_1(v)$.

(11) Let S_1 be a non-empty circuit of I_1, and let v be a vertex of I_1, and let i_1 be an input assignment of S_1. If $v \in \text{SortsWithConstants}(I_1)$, then $\text{InputGenValue}(v, i_1) = (\text{Set-Constants}(S_1))(v)$.

2. Circuit Computations

Let I_1 be a circuit-like non void non empty many sorted signature, let S_1 be a non-empty circuit of I_1, and let s be a state of S_1. The functor $\text{Following}(s)$ yielding a state of S_1 is defined by the condition (Def.5).

(Def.5) Let v be a vertex of I_1. Then if $v \in \text{InputVertices}(I_1)$, then $(\text{Following}(s))(v) = s(v)$ and if $v \in \text{InnerVertices}(I_1)$, then $(\text{Following}(s))(v) = (\text{Den}(\text{the action at } v, S_1))((\text{the action at } v) \text{ depends-on-in } s)$.

Next we state the proposition

(12) Let S_1 be a non-empty circuit of I_1, and let s be a state of S_1, and let i_1 be an input assignment of S_1. If $i_1 \subseteq s$, then $i_1 \subseteq \text{Following}(s)$.

Let I_1 be a circuit-like non void non empty many sorted signature and let S_1 be a non-empty circuit of I_1. A state of S_1 is stable if:

(Def.6) $\text{It} = \text{Following}(\text{it})$.

Let us consider I_1, let S_1 be a non-empty circuit of I_1, let s be a state of S_1, and let i_1 be an input assignment of S_1. The functor $\text{Following}(s, i_1)$ yielding a state of S_1 is defined by:

(Def.7) $\text{Following}(s, i_1) = \text{Following}(s +\cdot i_1)$.

Let us consider I_1, let S_1 be a non-empty circuit of I_1, let I_2 be an input function of S_1, and let s be a state of S_1. The functor $\text{InitialComp}(s, I_2)$ yielding a state of S_1 is defined as follows:

(Def.8) $\text{InitialComp}(s, I_2) = s +\cdot (0\text{-}th\text{-input}(I_2)) +\cdot \text{Set-Constants}(S_1)$.

Let us consider I_1, let S_1 be a non-empty circuit of I_1, let I_2 be an input function of S_1, and let s be a state of S_1. The functor $\text{Computation}(s, I_2)$ yielding a function from $\mathbb{N}$ into $\prod(\text{the sorts of } S_1)$ is defined by the conditions (Def.9).

(Def.9) (i) $(\text{Computation}(s, I_2))(0) = \text{InitialComp}(s, I_2)$, and

(ii) for every natural number i and for every state x of S_1 such that $x = (\text{Computation}(s, I_2))(i)$ holds $(\text{Computation}(s, I_2))(i+1) = \text{Following}(x, (i+1)\text{-}th\text{-input}(I_2))$.

In the sequel S_1 denotes a non-empty circuit of I_1, s denotes a state of S_1, and i_1 denotes an input assignment of S_1.

Next we state the proposition

(13) Let k be a natural number. Suppose that for every vertex v of I_1 such that $\mathrm{depth}(v, S_1) \leq k$ holds $s(v) = \mathrm{InputGenValue}(v, i_1)$. Let v_1 be a vertex of I_1. If $\mathrm{depth}(v_1, S_1) \leq k + 1$, then $(\mathrm{Following}(s))(v_1) = \mathrm{InputGenValue}(v_1, i_1)$.

For simplicity we adopt the following convention: I_1 is a finite monotonic circuit-like non void non empty many sorted signature, S_1 is a non-empty circuit of I_1, I_2 is an input function of S_1, s is a state of S_1, and i_1 is an input assignment of S_1.

We now state several propositions:

(14) If $\mathrm{commute}(I_2)$ is constant and $\mathrm{InputVertices}(I_1)$ is non empty, then for all s, i_1 such that $i_1 = (\mathrm{commute}(I_2))(0)$ and for every natural number k holds $i_1 \subseteq (\mathrm{Computation}(s, I_2))(k)$.

(15) Let n be a natural number. Suppose $\mathrm{commute}(I_2)$ is constant and $\mathrm{InputVertices}(I_1)$ is non empty and $(\mathrm{Computation}(s, I_2))(n)$ is stable. Let m be a natural number. If $n \leq m$, then $(\mathrm{Computation}(s, I_2))(n) = (\mathrm{Computation}(s, I_2))(m)$.

(16) Suppose $\mathrm{commute}(I_2)$ is constant and $\mathrm{InputVertices}(I_1)$ is non empty. Given s, i_1. Suppose $i_1 = (\mathrm{commute}(I_2))(0)$. Let k be a natural number and let v be a vertex of I_1. If $\mathrm{depth}(v, S_1) \leq k$, then $((\mathrm{Computation}(s, I_2))(k)$ **qua** element of $\prod(\text{the sorts of } S_1))(v) = \mathrm{InputGenValue}(v, i_1)$.

(17) Suppose $\mathrm{commute}(I_2)$ is constant and $\mathrm{InputVertices}(I_1)$ is non empty and $i_1 = (\mathrm{commute}(I_2))(0)$. Let s be a state of S_1 and let v be a vertex of I_1. Then $((\mathrm{Computation}(s, I_2))(\mathrm{depth}(S_1))$ **qua** state of $S_1)(v) = \mathrm{InputGenValue}(v, i_1)$.

(18) If $\mathrm{commute}(I_2)$ is constant and $\mathrm{InputVertices}(I_1)$ is non empty, then for every state s of S_1 holds $(\mathrm{Computation}(s, I_2))(\mathrm{depth}(S_1))$ is stable.

(19) If $\mathrm{commute}(I_2)$ is constant and $\mathrm{InputVertices}(I_1)$ is non empty, then for all states s_1, s_2 of S_1 holds $(\mathrm{Computation}(s_1, I_2))(\mathrm{depth}(S_1)) = (\mathrm{Computation}(s_2, I_2))(\mathrm{depth}(S_1))$.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.

[2] Grzegorz Bancerek. Cartesian product of functions. *Formalized Mathematics*, 2(**4**):547–552, 1991.

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

[4] Grzegorz Bancerek. Joining of decorated trees. *Formalized Mathematics*, 4(**1**):77–82, 1993.

[5] Grzegorz Bancerek. König's lemma. *Formalized Mathematics*, 2(**3**):397–402, 1991.

[6] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.

[7] Grzegorz Bancerek. Sets and functions of trees and joining operations of trees. *Formalized Mathematics*, 3(**2**):195–204, 1992.

[8] Grzegorz Bancerek. Terms over many sorted universal algebra. *Formalized Mathematics*, 5(**2**):191–198, 1996.

[9] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[10] Grzegorz Bancerek and Piotr Rudnicki. On defining functions on trees. *Formalized Mathematics*, 4(**1**):91–101, 1993.
[11] Ewa Burakowska. Subalgebras of many sorted algebra. Lattice of subalgebras. *Formalized Mathematics*, 5(**1**):47–54, 1996.
[12] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[13] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[14] 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.
[15] Patricia L. Carlson and Grzegorz Bancerek. Context-free grammar - part 1. *Formalized Mathematics*, 2(**5**):683–687, 1991.
[16] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[17] Małgorzata Korolkiewicz. Homomorphisms of many sorted algebras. *Formalized Mathematics*, 5(**1**):61–65, 1996.
[18] Jarosław Kotowicz. Monotone real sequences. Subsequences. *Formalized Mathematics*, 1(**3**):471–475, 1990.
[19] Beata Madras. Product of family of universal algebras. *Formalized Mathematics*, 4(**1**):103–108, 1993.
[20] Beata Madras. Products of many sorted algebras. *Formalized Mathematics*, 5(**1**):55–60, 1996.
[21] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Introduction to circuits, I. *Formalized Mathematics*, 5(**2**):227–232, 1996.
[22] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, II. *Formalized Mathematics*, 5(**2**):215–220, 1996.
[23] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, I. *Formalized Mathematics*, 5(**2**):167–172, 1996.
[24] Beata Perkowska. Free many sorted universal algebra. *Formalized Mathematics*, 5(**1**):67–74, 1996.
[25] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[26] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[27] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[28] Wojciech A. Trybulec. Groups. *Formalized Mathematics*, 1(**5**):821–827, 1990.
[29] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[30] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[31] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received April 10, 1995