JOURNAL OF FORMALIZED MATHEMATICS
Volume 7, Released 1995, Published 2003
Inst. of Computer Science, Univ. of 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 [22], [23], [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`.

WWW: `http://mizar.org/JFM/Vol7/circuit2.html`

The articles [25], [13], [32], [29], [33], [28], [11], [12], [18], [14], [2], [8], [16], [5], [6], [7], [30], [1], [3], [31], [4], [15], [9], [26], [19], [27], [10], [20], [17], [24], [22], [23], and [21] provide the notation and terminology for this paper.

1. Circuit Inputs

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

The following proposition is true

(1) Let X be a non-empty many sorted set indexed by the carrier of I_1, H be a many sorted function from $\mathrm{Free}(X)$ into $\mathrm{Free}(X)$, H_1 be a function yielding function, v be a sort symbol of I_1, p be a decorated tree yielding finite sequence, and t be an element of (the sorts of $\mathrm{Free}(X))(v)$. Suppose that

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

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

(iii) H is a homomorphism of $\mathrm{Free}(X)$ into $\mathrm{Free}(X)$, and

(iv) $H_1 = H \cdot \mathrm{Arity}(\text{the action at } v)$.

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

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

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 $\mathrm{FixInput}(i_1)$ yielding a many sorted function from FreeGenerator(the sorts of A) into the sorts of $\mathrm{FreeEnvelope}(A)$ is defined by the condition (Def. 1).

(Def. 1) Let v be a vertex of I_1. Then

(i) if $v \in \mathrm{InputVertices}(I_1)$, then $(\mathrm{FixInput}(i_1))(v) = \mathrm{FreeGenerator}(v, \text{the sorts of } A) \longmapsto$ the root tree of $\langle i_1(v), v\rangle$,

(ii) if $v \in \mathrm{SortsWithConstants}(I_1)$, then $(\mathrm{FixInput}(i_1))(v) = \mathrm{FreeGenerator}(v, \text{the sorts of } A) \longmapsto$ the root tree of $\langle$the action at v, the carrier of $I_1\rangle$, and

(iii) if $v \in \mathrm{InnerVertices}(I_1) \setminus \mathrm{SortsWithConstants}(I_1)$, then $(\mathrm{FixInput}(i_1))(v) = \mathrm{id}_{\mathrm{FreeGenerator}(v, \text{the sorts of } A)}$.

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 $\mathrm{FixInputExt}(i_1)$ yielding a many sorted function from $\mathrm{FreeEnvelope}(A)$ into $\mathrm{FreeEnvelope}(A)$ is defined as follows:

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

We now state several propositions:

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

(3) Let A be a non-empty circuit of I_1, i_1 be an input assignment of A, v be a vertex of I_1, and x be an element of (the sorts of $A)(v)$. If $v \in \mathrm{InputVertices}(I_1)$, then $(\mathrm{FixInputExt}(i_1))(v)$(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, i_1 be an input assignment of A, v be a vertex of I_1, e be an element of (the sorts of $\mathrm{FreeEnvelope}(A))(v)$, and 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 $q(k) = (\mathrm{FixInputExt}(i_1))(\mathrm{Arity}(\text{the action at } v)_k)(p(k))$.

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, i_1 be an input assignment of A, v be a vertex of I_1, and e be an element of (the sorts of $\mathrm{FreeEnvelope}(A))(v)$. 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, i_1 be an input assignment of A, v be a vertex of I_1, e, e_1 be elements of (the sorts of $\mathrm{FreeEnvelope}(A))(v)$, and 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, i_1 be an input assignment of A, v be a vertex of I_1, and e, e_1 be elements of (the sorts of $\mathrm{FreeEnvelope}(A))(v)$. 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 $\mathrm{InputGenTree}(v, i_1)$ yielding an element of (the sorts of $\mathrm{FreeEnvelope}(S_1))(v)$ is defined as follows:

(Def. 3) There exists an element e of (the sorts of $\mathrm{FreeEnvelope}(S_1))(v)$ 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, v be a vertex of I_1, and 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, v be a vertex of I_1, i_1 be an input assignment of S_1, and 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 $p(k) = \mathrm{InputGenTree}(\mathrm{Arity}(\text{the action at } v)_k, i_1)$.

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

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 $\mathrm{InputGenValue}(v,i_1)$ yielding an element of (the sorts of $S_1)(v)$ is defined as follows:

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

One can prove the following propositions:

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

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

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 $\mathrm{Following}(s)$ yields a state of S_1 and is defined by the condition (Def. 5).

(Def. 5) Let v be a vertex of I_1. Then

(i) if $v \in \mathrm{InputVertices}(I_1)$, then $(\mathrm{Following}(s))(v) = s(v)$, and

(ii) if $v \in \mathrm{InnerVertices}(I_1)$, then $(\mathrm{Following}(s))(v) = (\mathrm{Den}(\text{the action at } v, S_1))((\text{the action at } v) \text{ depends-on-in } s)$.

The following proposition is true

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

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 I_2 be a state of S_1. We say that I_2 is stable if and only if:

(Def. 6) $I_2 = \mathrm{Following}(I_2)$.

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. The functor $\mathrm{Following}(s,i_1)$ yields a state of S_1 and is defined by:

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

Let us consider I_1, let S_1 be a non-empty circuit of I_1, let I_3 be an input function of S_1, and let s be a state of S_1. The functor $\mathrm{InitialComp}(s,I_3)$ yielding a state of S_1 is defined as follows:

(Def. 8) $\mathrm{InitialComp}(s,I_3) = s {+\cdot} (0\text{-}th\text{-input}(I_3)) {+\cdot} \mathrm{Set\text{-}Constants}(S_1)$.

Let us consider I_1, let S_1 be a non-empty circuit of I_1, let I_3 be an input function of S_1, and let s be a state of S_1. The functor Computation(s, I_3) yields a function from $\mathbb{N}$ into $\prod$(the sorts of S_1) and is defined by:

(Def. 9) $(\text{Computation}(s,I_3))(0) = \text{InitialComp}(s,I_3)$ and for every natural number i holds $(\text{Computation}(s,I_3))(i+1) = \text{Following}((\text{Computation}(s,I_3))(i), (i+1)\text{-}th\text{-input}(I_3))$.

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

We now state the proposition

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

For simplicity, we adopt the following rules: 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_3 is an input function of S_1, s is a state of S_1, and i_1 is an input assignment of S_1.

The following propositions are true:

(14) If commute(I_3) is constant and InputVertices(I_1) is non empty, then for all s, i_1 such that $i_1 = (\text{commute}(I_3))(0)$ and for every natural number k holds $i_1 \subseteq (\text{Computation}(s,I_3))(k)$.

(15) Let n be a natural number. Suppose commute(I_3) is constant and InputVertices(I_1) is non empty and $(\text{Computation}(s,I_3))(n)$ is stable. Let m be a natural number. If $n \leq m$, then $(\text{Computation}(s,I_3))(n) = (\text{Computation}(s,I_3))(m)$.

(16) Suppose commute(I_3) is constant and InputVertices(I_1) is non empty. Let given s, i_1. Suppose $i_1 = (\text{commute}(I_3))(0)$. Let k be a natural number and v be a vertex of I_1. If $\text{depth}(v, S_1) \leq k$, then $((\text{Computation}(s,I_3))(k)$ **qua** element of $\prod(\text{the sorts of } S_1))(v) = \text{InputGenValue}(v, i_1)$.

(17) Suppose commute(I_3) is constant and InputVertices(I_1) is non empty and $i_1 = (\text{commute}(I_3))(0)$. Let s be a state of S_1, v be a vertex of I_1, and n be an element of $\mathbb{N}$. If $n = \text{depth}(S_1)$, then $((\text{Computation}(s,I_3))(n)$ **qua** state of $S_1)(v) = \text{InputGenValue}(v, i_1)$.

(18) Suppose commute(I_3) is constant and InputVertices(I_1) is non empty. Let s be a state of S_1 and n be an element of $\mathbb{N}$. If $n = \text{depth}(S_1)$, then $(\text{Computation}(s,I_3))(n)$ is stable.

(19) If commute(I_3) is constant and InputVertices(I_1) is non empty, then for all states s_1, s_2 of S_1 holds $(\text{Computation}(s_1,I_3))(\text{depth}(S_1)) = (\text{Computation}(s_2,I_3))(\text{depth}(S_1))$.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/card_1.html.

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

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

[4] Grzegorz Bancerek. Cartesian product of functions. *Journal of Formalized Mathematics*, 3, 1991. http://mizar.org/JFM/Vol3/funct_6.html.

[5] Grzegorz Bancerek. König's Lemma. *Journal of Formalized Mathematics*, 3, 1991. http://mizar.org/JFM/Vol3/trees_2.html.

[6] Grzegorz Bancerek. Sets and functions of trees and joining operations of trees. *Journal of Formalized Mathematics*, 4, 1992. http://mizar.org/JFM/Vol4/trees_3.html.

[7] Grzegorz Bancerek. Joining of decorated trees. *Journal of Formalized Mathematics*, 5, 1993. http://mizar.org/JFM/Vol5/trees_4.html.

[8] 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.

[9] Grzegorz Bancerek and Piotr Rudnicki. On defining functions on trees. *Journal of Formalized Mathematics*, 5, 1993. http://mizar.org/JFM/Vol5/dtconstr.html.

[10] Ewa Burakowska. Subalgebras of many sorted algebra. Lattice of subalgebras. *Journal of Formalized Mathematics*, 6, 1994. http://mizar.org/JFM/Vol6/msualg_2.html.

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

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

[13] Czesław Byliński. Some basic properties of sets. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/zfmisc_1.html.

[14] 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.

[15] Patricia L. Carlson and Grzegorz Bancerek. Context-free grammar — part I. *Journal of Formalized Mathematics*, 4, 1992. http://mizar.org/JFM/Vol4/lang1.html.

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

[17] Małgorzata Korolkiewicz. Homomorphisms of many sorted algebras. *Journal of Formalized Mathematics*, 6, 1994. http://mizar.org/JFM/Vol6/msualg_3.html.

[18] Jarosław Kotowicz. Monotone real sequences. Subsequences. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/seqm_3.html.

[19] Beata Madras. Product of family of universal algebras. *Journal of Formalized Mathematics*, 5, 1993. http://mizar.org/JFM/Vol5/pralg_1.html.

[20] Beata Madras. Products of many sorted algebras. *Journal of Formalized Mathematics*, 6, 1994. http://mizar.org/JFM/Vol6/pralg_2.html.

[21] 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.

[22] 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.

[23] 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.

[24] Beata Perkowska. Free many sorted universal algebra. *Journal of Formalized Mathematics*, 6, 1994. http://mizar.org/JFM/Vol6/msafree.html.

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

[26] Andrzej Trybulec. Many-sorted sets. *Journal of Formalized Mathematics*, 5, 1993. http://mizar.org/JFM/Vol5/pboole.html.

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

[28] Andrzej Trybulec. On the sets inhabited by numbers. *Journal of Formalized Mathematics*, 15, 2003. http://mizar.org/JFM/Vol16/membered.html.

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

[30] Wojciech A. Trybulec. Groups. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/group_1.html.

[31] Wojciech A. Trybulec. Pigeon hole principle. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/finseq_4.html.

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

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

Received April 10, 1995

Published January 2, 2004