JOURNAL OF FORMALIZED MATHEMATICS
Volume 15, Released 2003, Published 2003
Inst. of Computer Science, Univ. of Białystok

A Tree of Execution of a Macroinstruction[1]

Artur Korniłowicz
University of Bialystok, Poland

Summary. A tree of execution of a macroinstruction has been defined. It is a tree decorated by the instruction locations of a computer. Successors of each vertex are determined by the set of all possible values of the instruction counter after execution of the instruction placed in the location indicated by given vertex.

MML Identifier: AMISTD_3.

WWW: http://mizar.org/JFM/Vol15/amistd_3.html

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

For simplicity, we adopt the following convention: x, y, X denote sets, m, n denote natural numbers, O denotes an ordinal number, and R, S denote binary relations.

Let D be a set, let f be a partial function from D to $\mathbb{N}$, and let n be a set. Observe that $f(n)$ is natural.

Let R be an empty binary relation and let X be a set. One can check that $R{\restriction}X$ is empty.

We now state two propositions:

(1) If $\mathrm{dom}\,R = \{x\}$ and $\mathrm{rng}\,R = \{y\}$, then $R = x \dot{\longmapsto} y$.

(2) $\mathrm{field}\{\langle x, x\rangle\} = \{x\}$.

Let X be an infinite set and let a be a set. Note that $X \longmapsto a$ is infinite.

One can verify that there exists a function which is infinite.

Let R be a finite binary relation. One can check that $\mathrm{field}\,R$ is finite.

The following proposition is true

(3) If $\mathrm{field}\,R$ is finite, then R is finite.

Let R be an infinite binary relation. One can verify that $\mathrm{field}\,R$ is infinite.

We now state the proposition

(4) If $\mathrm{dom}\,R$ is finite and $\mathrm{rng}\,R$ is finite, then R is finite.

Let us observe that $\subseteq_{\emptyset}$ is empty.

Let X be a non empty set. Note that $\subseteq_X$ is non empty.

We now state two propositions:

(5) $\subseteq_{\{x\}} = \{\langle x, x\rangle\}$.

(6) $\subseteq_X \subseteq [: X, X :]$.

[1]The paper was written during author's post-doctoral fellowship granted by Shinshu University, Japan.

Let X be a finite set. One can check that ${}^{\subseteq}{}_{X}$ is finite.
Next we state the proposition

(7) If ${}^{\subseteq}{}_{X}$ is finite, then X is finite.

Let X be an infinite set. Observe that ${}^{\subseteq}{}_{X}$ is infinite.
We now state four propositions:

(8) If R and S are isomorphic and R is well-ordering, then S is well-ordering.

(9) If R and S are isomorphic and R is finite, then S is finite.

(10) $x \overset{\cdot}{\longmapsto} y$ is an isomorphism between $\{\langle x, x\rangle\}$ and $\{\langle y, y\rangle\}$.

(11) $\{\langle x, x\rangle\}$ and $\{\langle y, y\rangle\}$ are isomorphic.

Let us note that $\overline{\emptyset}$ is empty.
We now state four propositions:

(12) $\overline{{}^{\subseteq}{}_{O}} = O$.

(13) For every finite set X such that $X \subseteq O$ holds $\overline{{}^{\subseteq}{}_{X}} = \mathrm{card}\, X$.

(14) If $\{x\} \subseteq O$, then $\overline{{}^{\subseteq}{}_{\{x\}}} = 1$.

(15) If $\{x\} \subseteq O$, then the canonical isomorphism between ${}^{\subseteq}{}_{\overline{{}^{\subseteq}{}_{\{x\}}}}$ and ${}^{\subseteq}{}_{\{x\}} = 0 \overset{\cdot}{\longmapsto} x$.

Let O be an ordinal number, let X be a subset of O, and let n be a set. Note that (the canonical isomorphism between ${}^{\subseteq}{}_{\overline{{}^{\subseteq}{}_{X}}}$ and ${}^{\subseteq}{}_{X})(n)$ is ordinal.

Let X be a natural-membered set and let n be a set. Note that (the canonical isomorphism between ${}^{\subseteq}{}_{\overline{{}^{\subseteq}{}_{X}}}$ and ${}^{\subseteq}{}_{X})(n)$ is natural.

We now state three propositions:

(16) If $n \mapsto x = m \mapsto x$, then $n = m$.

(17) For every tree T and for every element t of T holds $t \restriction \mathrm{Seg}\, n \in T$.

(18) For all trees T_1, T_2 such that for every natural number n holds $T_1\text{-level}(n) = T_2\text{-level}(n)$ holds $T_1 = T_2$.

The functor TrivialInfiniteTree is defined as follows:

(Def. 1) $\mathrm{TrivialInfiniteTree} = \{k \mapsto 0 : k$ ranges over natural numbers$\}$.

One can check that TrivialInfiniteTree is non empty and tree-like.
Next we state the proposition

(19) $\mathbb{N} \approx \mathrm{TrivialInfiniteTree}$.

One can verify that TrivialInfiniteTree is infinite.
Next we state the proposition

(20) For every natural number n holds $\mathrm{TrivialInfiniteTree}\text{-level}(n) = \{n \mapsto 0\}$.

For simplicity, we adopt the following convention: N is a set with non empty elements, S is a standard IC-Ins-separated definite non empty non void AMI over N, L, l_1 are instruction-locations of S, J is an instruction of S, and F is a subset of the instruction locations of S.

Let N be a set with non empty elements, let S be a standard IC-Ins-separated definite non empty non void AMI over N, and let F be a finite partial state of S. Let us assume that F is non empty and F is programmed. The functor $\mathrm{FirstLoc}(F)$ yielding an instruction-location of S is defined by the condition (Def. 2).

(Def. 2) There exists a non empty subset M of $\mathbb{N}$ such that $M = \{\mathrm{locnum}(l); l$ ranges over elements of the instruction locations of S: $l \in \mathrm{dom}\,F\}$ and $\mathrm{FirstLoc}(F) = \mathrm{il}_S(\min M)$.

One can prove the following propositions:

(21) For every non empty programmed finite partial state F of S holds $\mathrm{FirstLoc}(F) \in \mathrm{dom}\,F$.

(22) For all non empty programmed finite partial states F, G of S such that $F \subseteq G$ holds $\mathrm{FirstLoc}(G) \leq \mathrm{FirstLoc}(F)$.

(23) For every non empty programmed finite partial state F of S such that $l_1 \in \mathrm{dom}\,F$ holds $\mathrm{FirstLoc}(F) \leq l_1$.

(24) For every lower non empty programmed finite partial state F of S holds $\mathrm{FirstLoc}(F) = \mathrm{il}_S(0)$.

Let N be a set with non empty elements, let S be a standard IC-Ins-separated definite non empty non void AMI over N, and let F be a subset of the instruction locations of S. The functor $\mathrm{LocNums}(F)$ yields a subset of $\mathbb{N}$ and is defined as follows:

(Def. 3) $\mathrm{LocNums}(F) = \{\mathrm{locnum}(l); l$ ranges over instruction-locations of S: $l \in F\}$.

The following proposition is true

(25) $\mathrm{locnum}(l_1) \in \mathrm{LocNums}(F)$ iff $l_1 \in F$.

Let N be a set with non empty elements, let S be a standard IC-Ins-separated definite non empty non void AMI over N, and let F be an empty subset of the instruction locations of S. One can check that $\mathrm{LocNums}(F)$ is empty.

Let N be a set with non empty elements, let S be a standard IC-Ins-separated definite non empty non void AMI over N, and let F be a non empty subset of the instruction locations of S. One can verify that $\mathrm{LocNums}(F)$ is non empty.

We now state several propositions:

(26) If $F = \{\mathrm{il}_S(n)\}$, then $\mathrm{LocNums}(F) = \{n\}$.

(27) $F \approx \mathrm{LocNums}(F)$.

(28) $\overline{\overline{F}} \subseteq \overline{\subseteq_{\mathrm{LocNums}(F)}}$.

(29) If S is realistic and J is halting, then $\mathrm{LocNums}(\mathrm{NIC}(J, L)) = \{\mathrm{locnum}(L)\}$.

(30) If S is realistic and J is sequential, then $\mathrm{LocNums}(\mathrm{NIC}(J, L)) = \{\mathrm{locnum}(\mathrm{NextLoc}\,L)\}$.

Let N be a set with non empty elements, let S be a standard IC-Ins-separated definite non empty non void AMI over N, and let M be a subset of the instruction locations of S. The functor $\mathrm{LocSeq}(M)$ yields a transfinite sequence of elements of the instruction locations of S and is defined by:

(Def. 4) $\mathrm{dom}\,\mathrm{LocSeq}(M) = \overline{\overline{M}}$ and for every set m such that $m \in \overline{\overline{M}}$ holds $(\mathrm{LocSeq}(M))(m) = \mathrm{il}_S(($the canonical isomorphism between $\subseteq_{\overline{\subseteq_{\mathrm{LocNums}(M)}}}$ and $\subseteq_{\mathrm{LocNums}(M)})(m))$.

The following proposition is true

(31) If $F = \{\mathrm{il}_S(n)\}$, then $\mathrm{LocSeq}(F) = 0 \mapsto \mathrm{il}_S(n)$.

Let N be a set with non empty elements, let S be a standard IC-Ins-separated definite non empty non void AMI over N, and let M be a subset of the instruction locations of S. One can verify that $\mathrm{LocSeq}(M)$ is one-to-one.

Let N be a set with non empty elements, let S be a standard IC-Ins-separated definite non empty non void AMI over N, and let M be a finite partial state of S. The functor $\mathrm{ExecTree}(M)$ yields a tree decorated with elements of the instruction locations of S and is defined by the conditions (Def. 5).

(Def. 5)(i) $(\text{ExecTree}(M))(\emptyset) = \text{FirstLoc}(M)$, and

(ii) for every element t of $\text{dom}\,\text{ExecTree}(M)$ holds $\text{succ}\,t = \{t \frown \langle k \rangle; k$ ranges over natural numbers: $k \in \overline{\overline{\text{NIC}(\pi_{(\text{ExecTree}(M))(t)}M, (\text{ExecTree}(M))(t))}}\}$ and for every natural number m such that $m \in \overline{\overline{\text{NIC}(\pi_{(\text{ExecTree}(M))(t)}M, (\text{ExecTree}(M))(t))}}$ holds $(\text{ExecTree}(M))(t \frown \langle m \rangle) = (\text{LocSeq}(\text{NIC}(\pi_{(\text{ExecTree}(M))(t)}M, (\text{ExecTree}(M))(t))))(m)$.

Next we state the proposition

(32) For every standard halting realistic IC-Ins-separated definite non empty non void AMI S over N holds $\text{ExecTree}(\text{Stop}\,S) = \text{TrivialInfiniteTree} \longmapsto \text{il}_S(0)$.

ACKNOWLEDGMENTS

The author wishes to thank Andrzej Trybulec and Grzegorz Bancerek for their very useful comments during writing this article.

REFERENCES

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

[2] Grzegorz Bancerek. Introduction to trees. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/trees_1.html`.

[3] Grzegorz Bancerek. The ordinal numbers. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/ordinal1.html`.

[4] Grzegorz Bancerek. Sequences of ordinal numbers. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/ordinal2.html`.

[5] Grzegorz Bancerek. The well ordering relations. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/wellord1.html`.

[6] Grzegorz Bancerek. Zermelo theorem and axiom of choice. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/wellord2.html`.

[7] Grzegorz Bancerek. König's Lemma. *Journal of Formalized Mathematics*, 3, 1991. `http://mizar.org/JFM/Vol3/trees_2.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] Czesław Byliński. Functions and their basic properties. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/funct_1.html`.

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

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

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

[13] Czesław Byliński. A classical first order language. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/cqc_lang.html`.

[14] 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`.

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

[16] Agata Darmochwał and Andrzej Trybulec. Similarity of formulae. *Journal of Formalized Mathematics*, 3, 1991. `http://mizar.org/JFM/Vol3/cqc_sim1.html`.

[17] Artur Korniłowicz. On the composition of macro instructions of standard computers. *Journal of Formalized Mathematics*, 12, 2000. `http://mizar.org/JFM/Vol12/amistd_2.html`.

[18] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Journal of Formalized Mathematics*, 4, 1992. `http://mizar.org/JFM/Vol4/ami_1.html`.

[19] Yasushi Tanaka. On the decomposition of the states of SCM. *Journal of Formalized Mathematics*, 5, 1993. `http://mizar.org/JFM/Vol5/ami_5.html`.

[20] Andrzej Trybulec. Binary operations applied to functions. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/funcop_1.html`.

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

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

[23] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Journal of Formalized Mathematics*, 5, 1993. http://mizar.org/JFM/Vol5/ami_3.html.

[24] Andrzej Trybulec, Piotr Rudnicki, and Artur Korniłowicz. Standard ordering of instruction locations. *Journal of Formalized Mathematics*, 12, 2000. http://mizar.org/JFM/Vol12/amistd_1.html.

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

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

Received December 10, 2003

Published December 10, 2003