FORMALIZED MATHEMATICS
Volume 4, Number 1, 1993
Université Catholique de Louvain

Euclid's Algorithm

Andrzej Trybulec
Warsaw University
Białystok

Yatsuka Nakamura
Shinshu University
Nagano

Summary. The main goal of the paper is to prove the correctness of the Euclid's algorithm for **SCM**. We define the Euclid's algorithm and describe the natural semantics of it. Eventually we prove that the Euclid's algorithm computes the Euclid's function. Let us observe that the Euclid's function is defined as a function mapping finite partial states to finite partial states of **SCM** rather than pairs of integers to integers.

MML Identifier: AMI_4.

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

1. Preliminaries

One can prove the following propositions:

(1) For all integers i, j such that $i \geq 0$ and $j > 0$ holds $i \div j \geq 0$.

(2) For all integers i, j such that $i \geq 0$ and $j > 0$ holds $|i| \bmod |j| = i \bmod j$ and $|i| \div |j| = i \div j$.

In the sequel i, j, k denote natural numbers.

Next we state the proposition

(3) For all i, j such that $i > 0$ and $j > 0$ holds $\gcd(i, j) > 0$.

The scheme *Euklides'* concerns a unary functor $\mathcal{F}$ yielding a natural number, a unary functor $\mathcal{G}$ yielding a natural number, a natural number $\mathcal{A}$, and a natural number $\mathcal{B}$, and states that:

There exists k such that $\mathcal{F}(k) = \gcd(\mathcal{A}, \mathcal{B})$ and $\mathcal{G}(k) = 0$

provided the following requirements are met:

- $0 < \mathcal{B}$,

ISSN 0777-4028

- $\mathcal{B} < \mathcal{A}$,
- $\mathcal{F}(0) = \mathcal{A}$,
- $\mathcal{G}(0) = \mathcal{B}$,
- For every k such that $\mathcal{G}(k) > 0$ holds $\mathcal{F}(k+1) = \mathcal{G}(k)$ and $\mathcal{G}(k+1) = \mathcal{F}(k) \bmod \mathcal{G}(k)$.

2. Euclid's Algorithm

The Euclid's algorithm is a programmed finite partial state of **SCM** and is defined by:

(Def.1) The Euclid's algorithm $= (\mathbf{i}_0 \mapsto (\mathbf{d}_2 := \mathbf{d}_1)) +\cdot ((\mathbf{i}_1 \mapsto \mathrm{Divide}(\mathbf{d}_0, \mathbf{d}_1)) +\cdot ((\mathbf{i}_2 \mapsto (\mathbf{d}_0 := \mathbf{d}_2)) +\cdot ((\mathbf{i}_3 \mapsto (\textbf{if } \mathbf{d}_1 > 0 \textbf{ goto } \mathbf{i}_0)) +\cdot (\mathbf{i}_4 \mapsto \mathbf{halt}_{\mathbf{SCM}}))))$.

Next we state the proposition

(4) $\mathrm{dom}\,(\text{the Euclid's algorithm}) = \{\mathbf{i}_0, \mathbf{i}_1, \mathbf{i}_2, \mathbf{i}_3, \mathbf{i}_4\}$.

3. The Natural Semantics of the Euclid's Algorithm

We now state several propositions:

(5) Let s be a state of **SCM**. Suppose the Euclid's algorithm $\subseteq s$. Given k. Suppose $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{i}_0$. Then $\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)} = \mathbf{i}_1$ and $(\mathrm{Computation}(s))(k+1)(\mathbf{d}_0) = (\mathrm{Computation}(s))(k)(\mathbf{d}_0)$ and $(\mathrm{Computation}(s))(k+1)(\mathbf{d}_1) = (\mathrm{Computation}(s))(k)(\mathbf{d}_1)$ and $(\mathrm{Computation}(s))(k+1)(\mathbf{d}_2) = (\mathrm{Computation}(s))(k)(\mathbf{d}_1)$.

(6) Let s be a state of **SCM**. Suppose the Euclid's algorithm $\subseteq s$. Given k. Suppose $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{i}_1$. Then $\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)} = \mathbf{i}_2$ and $(\mathrm{Computation}(s))(k+1)(\mathbf{d}_0) = (\mathrm{Computation}(s))(k)(\mathbf{d}_0) \div (\mathrm{Computation}(s))(k)(\mathbf{d}_1)$ and $(\mathrm{Computation}(s))(k+1)(\mathbf{d}_1) = (\mathrm{Computation}(s))(k)(\mathbf{d}_0) \bmod (\mathrm{Computation}(s))(k)(\mathbf{d}_1)$ and $(\mathrm{Computation}(s))(k+1)(\mathbf{d}_2) = (\mathrm{Computation}(s))(k)(\mathbf{d}_2)$.

(7) Let s be a state of **SCM**. Suppose the Euclid's algorithm $\subseteq s$. Given k. Suppose $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{i}_2$. Then $\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)} = \mathbf{i}_3$ and $(\mathrm{Computation}(s))(k+1)(\mathbf{d}_0) = (\mathrm{Computation}(s))(k)(\mathbf{d}_2)$ and $(\mathrm{Computation}(s))(k+1)(\mathbf{d}_1) = (\mathrm{Computation}(s))(k)(\mathbf{d}_1)$ and $(\mathrm{Computation}(s))(k+1)(\mathbf{d}_2) = (\mathrm{Computation}(s))(k)(\mathbf{d}_2)$.

(8) Let s be a state of **SCM**. Suppose the Euclid's algorithm $\subseteq s$. Given k. Suppose $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{i}_3$. Then

(i) if $(\mathrm{Computation}(s))(k)(\mathbf{d}_1) > 0$, then $\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)} = \mathbf{i}_0$,

(ii) if $(\mathrm{Computation}(s))(k)(\mathbf{d}_1) \leq 0$, then $\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)} = \mathbf{i}_4$,

(iii) $(\mathrm{Computation}(s))(k+1)(\mathbf{d}_0) = (\mathrm{Computation}(s))(k)(\mathbf{d}_0)$, and

(iv) $(\mathrm{Computation}(s))(k+1)(\mathbf{d}_1) = (\mathrm{Computation}(s))(k)(\mathbf{d}_1)$.

(9) For every state s of **SCM** such that the Euclid's algorithm $\subseteq s$ and for all k, i such that $\mathbf{IC}_{(\text{Computation}(s))(k)} = \mathbf{i}_4$ holds $(\text{Computation}(s))(k+i) = (\text{Computation}(s))(k)$.

(10) Let s be a state of **SCM**. Suppose s starts at $\mathbf{i}_0$ and the Euclid's algorithm $\subseteq s$. Let x, y be integers. If $s(\mathbf{d}_0) = x$ and $s(\mathbf{d}_1) = y$ and $x > 0$ and $y > 0$, then $(\text{Result}(s))(\mathbf{d}_0) = \gcd(x, y)$.

The Euclid's function is a partial function from FinPartSt(**SCM**) to FinPartSt(**SCM**) and is defined by the condition (Def.2).

(Def.2) Let p, q be finite partial states of **SCM**. Then $\langle p, q \rangle \in$ the Euclid's function if and only if there exist integers x, y such that $x > 0$ and $y > 0$ and $p = [\mathbf{d}_0 \longmapsto x, \mathbf{d}_1 \longmapsto y]$ and $q = \mathbf{d}_0 \dot{\longmapsto} \gcd(x, y)$.

The following three propositions are true:

(11) Let p be arbitrary. Then $p \in \text{dom}(\text{the Euclid's function})$ if and only if there exist integers x, y such that $x > 0$ and $y > 0$ and $p = [\mathbf{d}_0 \longmapsto x, \mathbf{d}_1 \longmapsto y]$.

(12) For all integers i, j such that $i > 0$ and $j > 0$ holds $(\text{the Euclid's function})([\mathbf{d}_0 \longmapsto i, \mathbf{d}_1 \longmapsto j]) = \mathbf{d}_0 \dot{\longmapsto} \gcd(i, j)$.

(13) Start-At$(\mathbf{i}_0)$+·(the Euclid's algorithm) computes the Euclid's function.

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 and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.

[4] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.

[5] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.

[6] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.

[7] 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.

[8] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.

[9] Czesław Byliński. Products and coproducts in categories. *Formalized Mathematics*, 2(**5**):701–709, 1991.

[10] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.

[11] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.

[12] Rafał Kwiatek and Grzegorz Zwara. The divisibility of integers and integer relative primes. *Formalized Mathematics*, 1(**5**):829–832, 1990.

[13] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.

[14] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Formalized Mathematics*, 3(**2**):241–250, 1992.

[15] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.

[16] Dariusz Surowik. Cyclic groups and some of their properties - part I. *Formalized Mathematics*, 2(**5**):623–627, 1991.

[17] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[18] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[19] Andrzej Trybulec. Function domains and Frænkel operator. *Formalized Mathematics*, 1(**3**):495–500, 1990.
[20] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[21] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[22] Michal J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[23] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received October 8, 1993