FORMALIZED MATHEMATICS
Volume 9, Number 2, 2001
University of Białystok

The Construction and Computation of While-Loop Programs for SCMPDS[1]

Jing-Chao Chen
Shanghai Jiaotong University

Summary. This article defines two while-loop statements on SCMPDS, i.e. "while<0" and "while>0", which resemble the while-statements of the common high language such as C. We previously presented a number of tricks for computing while-loop statements on SCMFSA, e.g. step-while. However, after inspecting a few realistic examples, we found that they are neither very useful nor of generalization. To cover much more computation cases of while-loop statements, we generalize the computation model of while-loop statements, based on the principle of Hoare's axioms on the verification of programs.

MML Identifier: `SCMPDS_8`.

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

1. Preliminaries

In this paper x, a denote Int positions and s denotes a state of SCMPDS.

We now state the proposition

(1) For every Int position a there exists a natural number i such that $a = \mathrm{intpos}\, i$.

Let t be a state of SCMPDS. The functor Dstate t yielding a state of SCMPDS is defined by the condition (Def. 1).

[1]This research is partially supported by the National Natural Science Foundation of China Grant No. 69873033.

ISSN 1426–2630

(Def. 1) Let x be a set. Then

(i) if $x \in \text{Data-Loc}_{\text{SCM}}$, then $(\text{Dstate}\,t)(x) = t(x)$,

(ii) if $x \in$ the instruction locations of SCMPDS, then $(\text{Dstate}\,t)(x) =$ goto 0, and

(iii) if $x = \mathbf{IC}_{\text{SCMPDS}}$, then $(\text{Dstate}\,t)(x) = \text{inspos}\,0$.

One can prove the following four propositions:

(2) For all states t_1, t_2 of SCMPDS such that $t_1 \restriction \text{Data-Loc}_{\text{SCM}} = t_2 \restriction \text{Data-Loc}_{\text{SCM}}$ holds $\text{Dstate}\,t_1 = \text{Dstate}\,t_2$.

(3) For every state t of SCMPDS and for every instruction i of SCMPDS such that $\text{InsCode}(i) \in \{0, 4, 5, 6\}$ holds $\text{Dstate}\,t = \text{Dstate}\,\text{Exec}(i, t)$.

(4) $(\text{Dstate}\,s)(a) = s(a)$.

(5) Let a be an Int position. Then there exists a function f from $\prod$(the object kind of SCMPDS) into $\mathbb{N}$ such that for every state s of SCMPDS holds

(i) if $s(a) \leqslant 0$, then $f(s) = 0$, and

(ii) if $s(a) > 0$, then $f(s) = s(a)$.

2. The Construction and Several Basic Properties of "while<0" Program

Let a be an Int position, let i be an integer, and let I be a Program-block. The functor while $< 0(a, i, I)$ yielding a Program-block is defined by:

(Def. 2) while $< 0(a, i, I) = ((a, i) >= 0\_\text{goto}\,\text{card}\,I + 2)$; I; goto $(-(\text{card}\,I + 1))$.

Let I be a shiftable Program-block, let a be an Int position, and let i be an integer. Observe that while $< 0(a, i, I)$ is shiftable.

Let I be a No-StopCode Program-block, let a be an Int position, and let i be an integer. Note that while $< 0(a, i, I)$ is No-StopCode.

Next we state several propositions:

(6) For every Int position a and for every integer i and for every Program-block I holds card while $< 0(a, i, I) = \text{card}\,I + 2$.

(7) Let a be an Int position, i be an integer, m be a natural number, and I be a Program-block. Then $m < \text{card}\,I + 2$ if and only if $\text{inspos}\,m \in$ dom while $< 0(a, i, I)$.

(8) Let a be an Int position, i be an integer, and I be a Program-block. Then (while $< 0(a, i, I))(\text{inspos}\,0) = (a, i) >= 0\_\text{goto}\,\text{card}\,I + 2$ and (while $< 0(a, i, I))(\text{inspos}\,\text{card}\,I + 1) = \text{goto}\ (-(\text{card}\,I + 1))$.

(9) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and i be an integer. If $s(\text{DataLoc}(s(a), i)) \geqslant 0$, then while $< 0(a, i, I)$ is closed on s and while $< 0(a, i, I)$ is halting on s.

(10) Let s be a state of SCMPDS, I be a Program-block, a, c be Int positions, and i be an integer. If $s(\mathrm{DataLoc}(s(a), i)) \geqslant 0$, then $\mathrm{IExec}(\mathrm{while} < 0(a, i, I), s) = s +\cdot \mathrm{Start\text{-}At}(\mathrm{inspos}\,\mathrm{card}\,I + 2)$.

(11) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and i be an integer. If $s(\mathrm{DataLoc}(s(a), i)) \geqslant 0$, then $\mathbf{IC}_{\mathrm{IExec}(\mathrm{while}<0(a,i,I),s)} = \mathrm{inspos}\,\mathrm{card}\,I + 2$.

(12) Let s be a state of SCMPDS, I be a Program-block, a, b be Int positions, and i be an integer. If $s(\mathrm{DataLoc}(s(a), i)) \geqslant 0$, then $(\mathrm{IExec}(\mathrm{while} < 0(a, i, I), s))(b) = s(b)$.

In this article we present several logical schemes. The scheme *WhileLHalt* deals with a unary functor $\mathcal{F}$ yielding a natural number, a state $\mathcal{A}$ of SCMPDS, a No-StopCode shiftable Program-block $\mathcal{B}$, an Int position $\mathcal{C}$, an integer $\mathcal{D}$, and a unary predicate $\mathcal{P}$, and states that:

$\mathcal{F}(\mathcal{A}) = \mathcal{F}(\mathcal{A})$ or $\mathcal{P}[\mathcal{A}]$ but while $< 0(\mathcal{C}, \mathcal{D}, \mathcal{B})$ is closed on $\mathcal{A}$ but while $< 0(\mathcal{C}, \mathcal{D}, \mathcal{B})$ is halting on $\mathcal{A}$

provided the following conditions are met:

- $\mathrm{card}\,\mathcal{B} > 0$,
- For every state t of SCMPDS such that $\mathcal{P}[\mathrm{Dstate}\,t]$ and $\mathcal{F}(\mathrm{Dstate}\,t) = 0$ holds $t(\mathrm{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) \geqslant 0$,
- $\mathcal{P}[\mathrm{Dstate}\,\mathcal{A}]$, and
- Let t be a state of SCMPDS. Suppose $\mathcal{P}[\mathrm{Dstate}\,t]$ and $t(\mathcal{C}) = \mathcal{A}(\mathcal{C})$ and $t(\mathrm{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) < 0$. Then $(\mathrm{IExec}(\mathcal{B}, t))(\mathcal{C}) = t(\mathcal{C})$ and $\mathcal{B}$ is closed on t and $\mathcal{B}$ is halting on t and $\mathcal{F}(\mathrm{Dstate}\,\mathrm{IExec}(\mathcal{B}, t)) < \mathcal{F}(\mathrm{Dstate}\,t)$ and $\mathcal{P}[\mathrm{Dstate}\,\mathrm{IExec}(\mathcal{B}, t)]$.

The scheme *WhileLExec* deals with a unary functor $\mathcal{F}$ yielding a natural number, a state $\mathcal{A}$ of SCMPDS, a No-StopCode shiftable Program-block $\mathcal{B}$, an Int position $\mathcal{C}$, an integer $\mathcal{D}$, and a unary predicate $\mathcal{P}$, and states that:

$\mathcal{F}(\mathcal{A}) = \mathcal{F}(\mathcal{A})$ or $\mathcal{P}[\mathcal{A}]$ but $\mathrm{IExec}(\mathrm{while} < 0(\mathcal{C}, \mathcal{D}, \mathcal{B}), \mathcal{A}) = \mathrm{IExec}(\mathrm{while} < 0(\mathcal{C}, \mathcal{D}, \mathcal{B}), \mathrm{IExec}(\mathcal{B}, \mathcal{A}))$

provided the parameters meet the following conditions:

- $\mathrm{card}\,\mathcal{B} > 0$,
- $\mathcal{A}(\mathrm{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) < 0$,
- For every state t of SCMPDS such that $\mathcal{P}[\mathrm{Dstate}\,t]$ and $\mathcal{F}(\mathrm{Dstate}\,t) = 0$ holds $t(\mathrm{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) \geqslant 0$,
- $\mathcal{P}[\mathrm{Dstate}\,\mathcal{A}]$, and
- Let t be a state of SCMPDS. Suppose $\mathcal{P}[\mathrm{Dstate}\,t]$ and $t(\mathcal{C}) = \mathcal{A}(\mathcal{C})$ and $t(\mathrm{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) < 0$. Then $(\mathrm{IExec}(\mathcal{B}, t))(\mathcal{C}) = t(\mathcal{C})$ and $\mathcal{B}$ is closed on t and $\mathcal{B}$ is halting on t and $\mathcal{F}(\mathrm{Dstate}\,\mathrm{IExec}(\mathcal{B}, t)) < \mathcal{F}(\mathrm{Dstate}\,t)$ and $\mathcal{P}[\mathrm{Dstate}\,\mathrm{IExec}(\mathcal{B}, t)]$.

One can prove the following propositions:

(13) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, i be an integer, X be a set, and f be a function from $\prod$(the object kind of SCMPDS) into $\mathbb{N}$. Suppose that

(i) $\operatorname{card} I > 0$,

(ii) for every state t of SCMPDS such that $f(\text{Dstate}\,t) = 0$ holds $t(\text{DataLoc}(s(a), i)) \geqslant 0$, and

(iii) for every state t of SCMPDS such that for every Int position x such that $x \in X$ holds $t(x) = s(x)$ and $t(a) = s(a)$ and $t(\text{DataLoc}(s(a), i)) < 0$ holds $(\text{IExec}(I, t))(a) = t(a)$ and $f(\text{Dstate}\,\text{IExec}(I, t)) < f(\text{Dstate}\,t)$ and I is closed on t and halting on t and for every Int position x such that $x \in X$ holds $(\text{IExec}(I, t))(x) = t(x)$.

Then while $< 0(a, i, I)$ is closed on s and while $< 0(a, i, I)$ is halting on s.

(14) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, i be an integer, X be a set, and f be a function from $\prod$(the object kind of SCMPDS) into $\mathbb{N}$. Suppose that

(i) $\operatorname{card} I > 0$,

(ii) $s(\text{DataLoc}(s(a), i)) < 0$,

(iii) for every state t of SCMPDS such that $f(\text{Dstate}\,t) = 0$ holds $t(\text{DataLoc}(s(a), i)) \geqslant 0$, and

(iv) for every state t of SCMPDS such that for every Int position x such that $x \in X$ holds $t(x) = s(x)$ and $t(a) = s(a)$ and $t(\text{DataLoc}(s(a), i)) < 0$ holds $(\text{IExec}(I, t))(a) = t(a)$ and I is closed on t and halting on t and $f(\text{Dstate}\,\text{IExec}(I, t)) < f(\text{Dstate}\,t)$ and for every Int position x such that $x \in X$ holds $(\text{IExec}(I, t))(x) = t(x)$.

Then $\text{IExec}(\text{while} < 0(a, i, I), s) = \text{IExec}(\text{while} < 0(a, i, I), \text{IExec}(I, s))$.

(15) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, i be an integer, and X be a set. Suppose that

(i) $\operatorname{card} I > 0$, and

(ii) for every state t of SCMPDS such that for every Int position x such that $x \in X$ holds $t(x) = s(x)$ and $t(a) = s(a)$ and $t(\text{DataLoc}(s(a), i)) < 0$ holds $(\text{IExec}(I, t))(a) = t(a)$ and $(\text{IExec}(I, t))(\text{DataLoc}(s(a), i)) > t(\text{DataLoc}(s(a), i))$ and I is closed on t and halting on t and for every Int position x such that $x \in X$ holds $(\text{IExec}(I, t))(x) = t(x)$.

Then while $< 0(a, i, I)$ is closed on s and while $< 0(a, i, I)$ is halting on s.

(16) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, i be an integer, and X be a set. Suppose that

(i) $s(\text{DataLoc}(s(a), i)) < 0$,

(ii) $\operatorname{card} I > 0$, and

(iii) for every state t of SCMPDS such that for every Int position x such

that $x \in X$ holds $t(x) = s(x)$ and $t(a) = s(a)$ and $t(\mathrm{DataLoc}(s(a), i)) < 0$ holds $(\mathrm{IExec}(I, t))(a) = t(a)$ and $(\mathrm{IExec}(I, t))(\mathrm{DataLoc}(s(a), i)) > t(\mathrm{DataLoc}(s(a), i))$ and I is closed on t and halting on t and for every Int position x such that $x \in X$ holds $(\mathrm{IExec}(I, t))(x) = t(x)$.
Then $\mathrm{IExec}(\mathrm{while} < 0(a, i, I), s) = \mathrm{IExec}(\mathrm{while} < 0(a, i, I), \mathrm{IExec}(I, s))$.

3. The Construction and Several Basic Properties of "while>0" Program

Let a be an Int position, let i be an integer, and let I be a Program-block. The functor while $> 0(a, i, I)$ yields a Program-block and is defined by:

(Def. 3) $\mathrm{while} > 0(a, i, I) = ((a, i) <= 0\_\mathrm{goto}\,\mathrm{card}\,I + 2); I; \mathrm{goto}\ (-(\mathrm{card}\,I + 1))$.

Let I be a shiftable Program-block, let a be an Int position, and let i be an integer. One can verify that while $> 0(a, i, I)$ is shiftable.

Let I be a No-StopCode Program-block, let a be an Int position, and let i be an integer. Note that while $> 0(a, i, I)$ is No-StopCode.

Next we state several propositions:

(17) For every Int position a and for every integer i and for every Program-block I holds $\mathrm{card}\,\mathrm{while} > 0(a, i, I) = \mathrm{card}\,I + 2$.

(18) Let a be an Int position, i be an integer, m be a natural number, and I be a Program-block. Then $m < \mathrm{card}\,I + 2$ if and only if $\mathrm{inspos}\,m \in \mathrm{dom}\,\mathrm{while} > 0(a, i, I)$.

(19) Let a be an Int position, i be an integer, and I be a Program-block. Then $(\mathrm{while} > 0(a, i, I))(\mathrm{inspos}\,0) = (a, i) <= 0\_\mathrm{goto}\,\mathrm{card}\,I + 2$ and $(\mathrm{while} > 0(a, i, I))(\mathrm{inspos}\,\mathrm{card}\,I + 1) = \mathrm{goto}\ (-(\mathrm{card}\,I + 1))$.

(20) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and i be an integer. If $s(\mathrm{DataLoc}(s(a), i)) \leqslant 0$, then while $> 0(a, i, I)$ is closed on s and while $> 0(a, i, I)$ is halting on s.

(21) Let s be a state of SCMPDS, I be a Program-block, a, c be Int positions, and i be an integer. If $s(\mathrm{DataLoc}(s(a), i)) \leqslant 0$, then $\mathrm{IExec}(\mathrm{while} > 0(a, i, I), s) = s {+\cdot}\, \mathrm{Start\text{-}At}(\mathrm{inspos}\,\mathrm{card}\,I + 2)$.

(22) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and i be an integer. If $s(\mathrm{DataLoc}(s(a), i)) \leqslant 0$, then $\mathbf{IC}_{\mathrm{IExec}(\mathrm{while}>0(a,i,I),s)} = \mathrm{inspos}\,\mathrm{card}\,I + 2$.

(23) Let s be a state of SCMPDS, I be a Program-block, a, b be Int positions, and i be an integer. If $s(\mathrm{DataLoc}(s(a), i)) \leqslant 0$, then $(\mathrm{IExec}(\mathrm{while} > 0(a, i, I), s))(b) = s(b)$.

Now we present two schemes. The scheme *WhileGHalt* deals with a unary functor $\mathcal{F}$ yielding a natural number, a state $\mathcal{A}$ of SCMPDS, a No-StopCode shiftable Program-block $\mathcal{B}$, an Int position $\mathcal{C}$, an integer $\mathcal{D}$, and a unary predicate $\mathcal{P}$, and states that:

$\mathcal{F}(\mathcal{A}) = \mathcal{F}(\mathcal{A})$ or $\mathcal{P}[\mathcal{A}]$ but while $> 0(\mathcal{C}, \mathcal{D}, \mathcal{B})$ is closed on $\mathcal{A}$ but while $> 0(\mathcal{C}, \mathcal{D}, \mathcal{B})$ is halting on $\mathcal{A}$

provided the parameters meet the following conditions:

- $\operatorname{card} \mathcal{B} > 0$,
- For every state t of SCMPDS such that $\mathcal{P}[\text{Dstate}\, t]$ and $\mathcal{F}(\text{Dstate}\, t) = 0$ holds $t(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) \leqslant 0$,
- $\mathcal{P}[\text{Dstate}\, \mathcal{A}]$, and
- Let t be a state of SCMPDS. Suppose $\mathcal{P}[\text{Dstate}\, t]$ and $t(\mathcal{C}) = \mathcal{A}(\mathcal{C})$ and $t(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) > 0$. Then $(\text{IExec}(\mathcal{B}, t))(\mathcal{C}) = t(\mathcal{C})$ and $\mathcal{B}$ is closed on t and $\mathcal{B}$ is halting on t and $\mathcal{F}(\text{Dstate}\,\text{IExec}(\mathcal{B}, t)) < \mathcal{F}(\text{Dstate}\, t)$ and $\mathcal{P}[\text{Dstate}\,\text{IExec}(\mathcal{B}, t)]$.

The scheme *WhileGExec* deals with a unary functor $\mathcal{F}$ yielding a natural number, a state $\mathcal{A}$ of SCMPDS, a No-StopCode shiftable Program-block $\mathcal{B}$, an Int position $\mathcal{C}$, an integer $\mathcal{D}$, and a unary predicate $\mathcal{P}$, and states that:

$\mathcal{F}(\mathcal{A}) = \mathcal{F}(\mathcal{A})$ or $\mathcal{P}[\mathcal{A}]$ but $\text{IExec}(\text{while} > 0(\mathcal{C}, \mathcal{D}, \mathcal{B}), \mathcal{A}) = \text{IExec}(\text{while} > 0(\mathcal{C}, \mathcal{D}, \mathcal{B}), \text{IExec}(\mathcal{B}, \mathcal{A}))$

provided the following conditions are satisfied:

- $\operatorname{card} \mathcal{B} > 0$,
- $\mathcal{A}(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) > 0$,
- For every state t of SCMPDS such that $\mathcal{P}[\text{Dstate}\, t]$ and $\mathcal{F}(\text{Dstate}\, t) = 0$ holds $t(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) \leqslant 0$,
- $\mathcal{P}[\text{Dstate}\, \mathcal{A}]$, and
- Let t be a state of SCMPDS. Suppose $\mathcal{P}[\text{Dstate}\, t]$ and $t(\mathcal{C}) = \mathcal{A}(\mathcal{C})$ and $t(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) > 0$. Then $(\text{IExec}(\mathcal{B}, t))(\mathcal{C}) = t(\mathcal{C})$ and $\mathcal{B}$ is closed on t and $\mathcal{B}$ is halting on t and $\mathcal{F}(\text{Dstate}\,\text{IExec}(\mathcal{B}, t)) < \mathcal{F}(\text{Dstate}\, t)$ and $\mathcal{P}[\text{Dstate}\,\text{IExec}(\mathcal{B}, t)]$.

One can prove the following propositions:

(24) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, i, c be integers, X, Y be sets, and f be a function from $\prod$(the object kind of SCMPDS) into $\mathbb{N}$. Suppose that

(i) $\operatorname{card} I > 0$,

(ii) for every state t of SCMPDS such that $f(\text{Dstate}\, t) = 0$ holds $t(\text{DataLoc}(s(a), i)) \leqslant 0$,

(iii) for every x such that $x \in X$ holds $s(x) \geqslant c + s(\text{DataLoc}(s(a), i))$, and

(iv) for every state t of SCMPDS such that for every x such that $x \in X$ holds $t(x) \geqslant c + t(\text{DataLoc}(s(a), i))$ and for every x such that $x \in Y$ holds $t(x) = s(x)$ and $t(a) = s(a)$ and $t(\text{DataLoc}(s(a), i)) > 0$ holds $(\text{IExec}(I, t))(a) = t(a)$ and I is closed on t and halting on t and $f(\text{Dstate}\,\text{IExec}(I, t)) < f(\text{Dstate}\, t)$ and for every x such that $x \in X$ holds $(\text{IExec}(I, t))(x) \geqslant$

$c + (\mathrm{IExec}(I,t))(\mathrm{DataLoc}(s(a),i))$ and for every x such that $x \in Y$ holds $(\mathrm{IExec}(I,t))(x) = t(x)$.

Then while $> 0(a,i,I)$ is closed on s and while $> 0(a,i,I)$ is halting on s.

(25) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, i, c be integers, X, Y be sets, and f be a function from $\prod$(the object kind of SCMPDS) into $\mathbb{N}$. Suppose that

(i) $s(\mathrm{DataLoc}(s(a),i)) > 0$,

(ii) $\mathrm{card}\, I > 0$,

(iii) for every state t of SCMPDS such that $f(\mathrm{Dstate}\, t) = 0$ holds $t(\mathrm{DataLoc}(s(a),i)) \leqslant 0$,

(iv) for every x such that $x \in X$ holds $s(x) \geqslant c + s(\mathrm{DataLoc}(s(a),i))$, and

(v) for every state t of SCMPDS such that for every x such that $x \in X$ holds $t(x) \geqslant c+t(\mathrm{DataLoc}(s(a),i))$ and for every x such that $x \in Y$ holds $t(x) = s(x)$ and $t(a) = s(a)$ and $t(\mathrm{DataLoc}(s(a),i)) > 0$ holds $(\mathrm{IExec}(I,t))(a) = t(a)$ and I is closed on t and halting on t and $f(\mathrm{Dstate}\,\mathrm{IExec}(I,t)) < f(\mathrm{Dstate}\, t)$ and for every x such that $x \in X$ holds $(\mathrm{IExec}(I,t))(x) \geqslant c + (\mathrm{IExec}(I,t))(\mathrm{DataLoc}(s(a),i))$ and for every x such that $x \in Y$ holds $(\mathrm{IExec}(I,t))(x) = t(x)$.

Then $\mathrm{IExec}(\text{while} > 0(a,i,I),s) = \mathrm{IExec}(\text{while} > 0(a,i,I),\mathrm{IExec}(I,s))$.

(26) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, i be an integer, X be a set, and f be a function from $\prod$(the object kind of SCMPDS) into $\mathbb{N}$. Suppose that

(i) $\mathrm{card}\, I > 0$,

(ii) for every state t of SCMPDS such that $f(\mathrm{Dstate}\, t) = 0$ holds $t(\mathrm{DataLoc}(s(a),i)) \leqslant 0$, and

(iii) for every state t of SCMPDS such that for every x such that $x \in X$ holds $t(x) = s(x)$ and $t(a) = s(a)$ and $t(\mathrm{DataLoc}(s(a),i)) > 0$ holds $(\mathrm{IExec}(I,t))(a) = t(a)$ and I is closed on t and halting on t and $f(\mathrm{Dstate}\,\mathrm{IExec}(I,t)) < f(\mathrm{Dstate}\, t)$ and for every x such that $x \in X$ holds $(\mathrm{IExec}(I,t))(x) = t(x)$.

Then while $> 0(a,i,I)$ is closed on s and while $> 0(a,i,I)$ is halting on s and if $s(\mathrm{DataLoc}(s(a),i)) > 0$, then $\mathrm{IExec}(\text{while} > 0(a,i,I),s) = \mathrm{IExec}(\text{while} > 0(a,i,I),\mathrm{IExec}(I,s))$.

(27) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, i, c be integers, and X, Y be sets. Suppose that

(i) $\mathrm{card}\, I > 0$,

(ii) for every x such that $x \in X$ holds $s(x) \geqslant c + s(\mathrm{DataLoc}(s(a),i))$, and

(iii) for every state t of SCMPDS such that for every x such that $x \in X$ holds $t(x) \geqslant c + t(\mathrm{DataLoc}(s(a),i))$ and for every x such that $x \in Y$ holds $t(x) = s(x)$ and $t(a) = s(a)$ and $t(\mathrm{DataLoc}(s(a),i)) > 0$ holds

(IExec(I, t))(a) $=$ $t(a)$ and I is closed on t and halting on t and (IExec(I, t))(DataLoc($s(a), i$)) $<$ t(DataLoc($s(a), i$)) and for every x such that $x \in X$ holds (IExec(I, t))(x) $\geqslant c+$(IExec(I, t))(DataLoc($s(a), i$)) and for every x such that $x \in Y$ holds (IExec(I, t))(x) $= t(x)$.

Then while $> 0(a, i, I)$ is closed on s and while $> 0(a, i, I)$ is halting on s and if s(DataLoc($s(a), i$)) > 0, then IExec(while $> 0(a, i, I), s$) $=$ IExec(while $> 0(a, i, I)$, IExec(I, s)).

(28) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, i be an integer, and X be a set. Suppose that

(i) card $I > 0$, and

(ii) for every state t of SCMPDS such that for every x such that $x \in X$ holds $t(x) = s(x)$ and $t(a) = s(a)$ and t(DataLoc($s(a), i$)) > 0 holds (IExec(I, t))(a) $=$ $t(a)$ and I is closed on t and halting on t and (IExec(I, t))(DataLoc($s(a), i$)) $<$ t(DataLoc($s(a), i$)) and for every x such that $x \in X$ holds (IExec(I, t))(x) $= t(x)$.

Then while $> 0(a, i, I)$ is closed on s and while $> 0(a, i, I)$ is halting on s and if s(DataLoc($s(a), i$)) > 0, then IExec(while $> 0(a, i, I), s$) $=$ IExec(while $> 0(a, i, I)$, IExec(I, s)).

(29) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, i, c be integers, and X be a set. Suppose that

(i) card $I > 0$,

(ii) for every x such that $x \in X$ holds $s(x) \geqslant c + s$(DataLoc($s(a), i$)), and

(iii) for every state t of SCMPDS such that for every x such that $x \in X$ holds $t(x) \geqslant c + t$(DataLoc($s(a), i$)) and $t(a) = s(a)$ and t(DataLoc($s(a), i$)) > 0 holds (IExec(I, t))(a) $= t(a)$ and I is closed on t and halting on t and (IExec(I, t))(DataLoc($s(a), i$)) $<$ t(DataLoc($s(a), i$)) and for every x such that $x \in X$ holds (IExec(I, t))(x) $\geqslant c +$ (IExec(I, t))(DataLoc($s(a), i$)).

Then while $> 0(a, i, I)$ is closed on s and while $> 0(a, i, I)$ is halting on s and if s(DataLoc($s(a), i$)) > 0, then IExec(while $> 0(a, i, I), s$) $=$ IExec(while $> 0(a, i, I)$, IExec(I, s)).

References

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

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

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

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

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

[6] Jing-Chao Chen. Computation and program shift in the SCMPDS computer. *Formalized Mathematics*, 8(**1**):193–199, 1999.

[7] Jing-Chao Chen. Computation of two consecutive program blocks for SCMPDS. *Formalized Mathematics*, 8(**1**):211–217, 1999.
[8] Jing-Chao Chen. The construction and computation of conditional statements for SCMPDS. *Formalized Mathematics*, 8(**1**):219–234, 1999.
[9] Jing-Chao Chen. The construction and shiftability of program blocks for SCMPDS. *Formalized Mathematics*, 8(**1**):201–210, 1999.
[10] Jing-Chao Chen. The SCMPDS computer and the basic semantics of its instructions. *Formalized Mathematics*, 8(**1**):183–191, 1999.
[11] Jing-Chao Chen. Recursive Euclide algorithm. *Formalized Mathematics*, 9(**1**):1–4, 2001.
[12] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[13] Krzysztof Hryniewiecki. Recursive definitions. *Formalized Mathematics*, 1(**2**):321–328, 1990.
[14] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[15] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Formalized Mathematics*, 3(**2**):241–250, 1992.
[16] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.
[17] Andrzej Trybulec. Domains and their Cartesian products. *Formalized Mathematics*, 1(**1**):115–122, 1990.
[18] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[19] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[20] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[21] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[22] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[23] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

Received June 14, 2000