JOURNAL OF FORMALIZED MATHEMATICS
Volume 12, Released 2000, Published 2003
Inst. of Computer Science, Univ. 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`.

WWW: `http://mizar.org/JFM/Vol12/scmpds_8.html`

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

1. PRELIMINARIES

In this paper x, a are Int positions and s is a state of SCMPDS.

Next we 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 $\mathrm{Dstate}\, t$ yielding a state of SCMPDS is defined by the condition (Def. 1).

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

- (i) if $x \in \mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$, then $(\mathrm{Dstate}\, t)(x) = t(x)$,
- (ii) if $x \in$ the instruction locations of SCMPDS, then $(\mathrm{Dstate}\, t)(x) = \mathrm{goto}\ 0$, and
- (iii) if $x = \mathbf{IC}_{\mathrm{SCMPDS}}$, then $(\mathrm{Dstate}\, t)(x) = \mathrm{inspos}\, 0$.

The following four propositions are true:

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

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

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

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

(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 $s(a) \leq 0$, 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 $<$ $0(a,i,I)$ yields a Program-block and is defined by:

(Def. 2) while $< 0(a,i,I) = ((a,i) >= 0$_goto card$I+2); I;$ 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 check 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. Observe that while $< 0(a,i,I)$ is No-StopCode.

We now state several propositions:

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

(7) Let a be an Int position, i be an integer, m be a natural number, and I be a Program-block. Then $m < \mathrm{card}\,I+2$ if and only if 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 $<$ $0(a,i,I))(\mathrm{inspos}\,0) = (a,i) >= 0$_goto card$I+2$ and (while $< 0(a,i,I))(\mathrm{inspos\,card}\,I+1) =$ goto $(-(\mathrm{card}\,I+1))$.

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

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

- 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 $t(\mathrm{DataLoc}(\mathcal{A}(\mathcal{C}),\mathcal{D})) \geq 0$,
- $\mathcal{P}[\mathrm{Dstate}\,\mathcal{A}]$, 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 $\mathcal{F}(\mathrm{Dstate\,IExec}(\mathcal{B},t)) < \mathcal{F}(\mathrm{Dstate}\,t)$ and $\mathcal{P}[\mathrm{Dstate\,IExec}(\mathcal{B},t)]$.

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 IExec(while $< 0(\mathcal{C},\mathcal{D},\mathcal{B}),\mathcal{A}) =$ IExec(while $< 0(\mathcal{C},\mathcal{D},\mathcal{B}),\mathrm{IExec}(\mathcal{B},\mathcal{A}))$

provided the parameters meet the following conditions:

- 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 $t(\text{DataLoc}(\mathcal{A}(\mathcal{C}),\mathcal{D})) \geq 0$,
- $\mathcal{P}[\text{Dstate}\, \mathcal{A}]$, 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 $\mathcal{P}[\text{Dstate}\,\text{IExec}(\mathcal{B},t)]$.

Next we state four 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) card $I > 0$,

(ii) for every state t of SCMPDS such that $f(\text{Dstate}\, t) = 0$ holds $t(\text{DataLoc}(s(a),i)) \geq 0$, 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) 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 $t(\text{DataLoc}(s(a),i)) \geq 0$, 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) 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) 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 $t(\text{DataLoc}(s(a),i)) < 0$ 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 $\text{IExec}(\text{while} < 0(a,i,I),s) = \text{IExec}(\text{while} < 0(a,i,I),\text{IExec}(I,s))$.

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

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. One can verify that while $> 0(a,i,I)$ is No-StopCode.

One can prove the following propositions:

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

(18) 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 inspos $m \in$ dom while $> 0(a,i,I)$.

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

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

(23) Let s be a state of SCMPDS, I be a Program-block, a, b be Int positions, and i be an integer. If $s(\text{DataLoc}(s(a),i)) \leq 0$, then (IExec(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 following requirements are met:

- 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 $t(\text{DataLoc}(\mathcal{A}(\mathcal{C}),\mathcal{D})) \leq 0$,
- $\mathcal{P}[\text{Dstate}\,\mathcal{A}]$, 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 $\mathcal{F}(\text{Dstate IExec}(\mathcal{B},t)) < \mathcal{F}(\text{Dstate}\,t)$ and $\mathcal{P}[\text{Dstate IExec}(\mathcal{B},t)]$.

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 IExec(while $> 0(\mathcal{C},\mathcal{D},\mathcal{B}),\mathcal{A}) =$ IExec(while $> 0(\mathcal{C},\mathcal{D},\mathcal{B}),\text{IExec}(\mathcal{B},\mathcal{A}))$

provided the parameters have the following properties:

- 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 $t(\text{DataLoc}(\mathcal{A}(\mathcal{C}),\mathcal{D})) \leq 0$,
- $\mathcal{P}[\text{Dstate}\,\mathcal{A}]$, and

- Let t be a state of SCMPDS. Suppose $\mathcal{P}$[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 $\mathcal{F}(\mathrm{Dstate\,IExec}(\mathcal{B},t)) < \mathcal{F}(\mathrm{Dstate}\,t)$ and $\mathcal{P}$[Dstate IExec$(\mathcal{B},t)$].

The following propositions are true:

(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) card $I > 0$,

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

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

(iv) for every state t of SCMPDS such that for every x such that $x \in X$ holds $t(x) \geq c + t(\mathrm{DataLoc}(s(a),i))$ 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 $f(\mathrm{Dstate\,IExec}(I,t)) < f(\mathrm{Dstate}\,t)$ and for every x such that $x \in X$ holds $(\mathrm{IExec}(I,t))(x) \geq c + (\mathrm{IExec}(I,t))(\mathrm{DataLoc}(s(a),i))$ 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) card $I > 0$,

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

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

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

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

(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) card $I > 0$,

(ii) for every state t of SCMPDS such that $f(\mathrm{Dstate}\,t) = 0$ holds $t(\mathrm{DataLoc}(s(a),i)) \leq 0$, 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 $f(\mathrm{Dstate\,IExec}(I,t)) < f(\mathrm{Dstate}\,t)$ 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 $\mathrm{IExec}(\mathrm{while} > 0(a,i,I),s) = \mathrm{IExec}(\mathrm{while} > 0(a,i,I), \mathrm{IExec}(I,s))$.

(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) card $I > 0$,

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

(iii) for every state t of SCMPDS such that for every x such that $x \in X$ holds $t(x) \geq c + t(\mathrm{DataLoc}(s(a),i))$ 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 $(\mathrm{IExec}(I,t))(\mathrm{DataLoc}(s(a),i)) < t(\mathrm{DataLoc}(s(a),i))$ and for every x such that $x \in X$ holds $(\mathrm{IExec}(I,t))(x) \geq c + (\mathrm{IExec}(I,t))(\mathrm{DataLoc}(s(a),i))$ 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 and if $s(\mathrm{DataLoc}(s(a),i)) > 0$, then $\mathrm{IExec}(\text{while} > 0(a,i,I),s) = \mathrm{IExec}(\text{while} > 0(a,i,I),\mathrm{IExec}(I,s))$.

(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) $\mathrm{card}\, I > 0$, 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 $t(\mathrm{DataLoc}(s(a),i)) > 0$ holds $(\mathrm{IExec}(I,t))(a) = t(a)$ and I is closed on t and halting on t and $(\mathrm{IExec}(I,t))(\mathrm{DataLoc}(s(a),i)) < t(\mathrm{DataLoc}(s(a),i))$ 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 $\mathrm{IExec}(\text{while} > 0(a,i,I),s) = \mathrm{IExec}(\text{while} > 0(a,i,I),\mathrm{IExec}(I,s))$.

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

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

(iii) for every state t of SCMPDS such that for every x such that $x \in X$ holds $t(x) \geq c + t(\mathrm{DataLoc}(s(a),i))$ 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 $(\mathrm{IExec}(I,t))(\mathrm{DataLoc}(s(a),i)) < t(\mathrm{DataLoc}(s(a),i))$ and for every x such that $x \in X$ holds $(\mathrm{IExec}(I,t))(x) \geq c + (\mathrm{IExec}(I,t))(\mathrm{DataLoc}(s(a),i))$.

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

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 and Andrzej Trybulec. Miscellaneous facts about functions. *Journal of Formalized Mathematics*, 8, 1996. http://mizar.org/JFM/Vol8/funct_7.html.

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

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

[7] 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.

[8] Jing-Chao Chen. Computation and program shift in the SCMPDS computer. *Journal of Formalized Mathematics*, 11, 1999. http://mizar.org/JFM/Vol11/scmpds_3.html.

[9] Jing-Chao Chen. Computation of two consecutive program blocks for SCMPDS. *Journal of Formalized Mathematics*, 11, 1999. http://mizar.org/JFM/Vol11/scmpds_5.html.

[10] Jing-Chao Chen. The construction and computation of conditional statements for SCMPDS. *Journal of Formalized Mathematics*, 11, 1999. http://mizar.org/JFM/Vol11/scmpds_6.html.

[11] Jing-Chao Chen. The construction and shiftability of program blocks for SCMPDS. *Journal of Formalized Mathematics*, 11, 1999. http://mizar.org/JFM/Vol11/scmpds_4.html.

[12] Jing-Chao Chen. Recursive Euclide algorithm. *Journal of Formalized Mathematics*, 11, 1999. http://mizar.org/JFM/Vol11/scmp_gcd.html.

[13] Jing-Chao Chen. The SCMPDS computer and the basic semantics of its instructions. *Journal of Formalized Mathematics*, 11, 1999. http://mizar.org/JFM/Vol11/scmpds_2.html.

[14] Krzysztof Hryniewiecki. Recursive definitions. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/recdef_1.html.

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

[16] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Journal of Formalized Mathematics*, 4, 1992. http://mizar.org/JFM/Vol4/ami_2.html.

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

[18] Andrzej Trybulec. Domains and their Cartesian products. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/domain_1.html.

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

[20] 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.

[21] Michał J. Trybulec. Integers. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/int_1.html.

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

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

Received June 14, 2000

Published January 2, 2004