JOURNAL OF FORMALIZED MATHEMATICS
Volume 11, Released 1999, Published 2003
Inst. of Computer Science, Univ. of Białystok

The Construction and Computation of Conditional Statements for SCMPDS[1]

Jing-Chao Chen
Shanghai Jiaotong University

Summary. We construct conditional statements like the usual high level program language by program blocks of SCMPDS. Roughly speaking, the article justifies such a fact that when the condition of a conditional statement is true (false), and the true (false) branch is shiftable, parahalting and does not contain any halting instruction, and the false branch is shiftable, then it is halting and its computation result equals that of the true (false) branch. The parahalting means some program halts for all states, this is strong condition. For this reason, we introduce the notions of "is_closed_on" and "is_halting_on". The predicate "A is_closed_on B" denotes program A is closed on state B, and "A is_halting_on B" denotes program A is halting on state B. We obtain a similar theorem to the above fact by replacing parahalting by "is_closed_on" and "is_halting_on".

MML Identifier: `SCMPDS_6`.

WWW: `http://mizar.org/JFM/Vol11/scmpds_6.html`

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

1. Preliminaries

For simplicity, we adopt the following convention: a denotes an Int position, i denotes an instruction of SCMPDS, s, s_1, s_2 denote states of SCMPDS, k_1 denotes an integer, l_1 denotes an instruction-location of SCMPDS, and I, J denote Program-blocks.

Next we state a number of propositions:

(1) For every state s of SCMPDS holds $\mathrm{dom}(s \restriction \text{the instruction locations of SCMPDS}) = $ the instruction locations of SCMPDS.

(2) For every state s of SCMPDS such that s is halting and for every natural number k such that $\mathrm{LifeSpan}(s) \leq k$ holds $\mathrm{CurInstr}((\mathrm{Computation}(s))(k)) = \mathbf{halt}_{\mathrm{SCMPDS}}$.

(3) For every state s of SCMPDS such that s is halting and for every natural number k such that $\mathrm{LifeSpan}(s) \leq k$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{IC}_{(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s))}$.

(4) Let s_1, s_2 be states of SCMPDS. Then s_1 and s_2 are equal outside the instruction locations of SCMPDS if and only if $\mathbf{IC}_{(s_1)} = \mathbf{IC}_{(s_2)}$ and $s_1 \restriction \text{Data-Loc}_{\mathrm{SCM}} = s_2 \restriction \text{Data-Loc}_{\mathrm{SCM}}$.

(5) For every state s of SCMPDS and for every Program-block I holds $\mathrm{Initialized}(s) +\cdot \mathrm{Initialized}(I) = s +\cdot \mathrm{Initialized}(I)$.

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

(6) For every Program-block I and for every instruction-location l of SCMPDS holds $I \subseteq I +\cdot \text{Start-At}(l)$.

(7) For every state s of SCMPDS and for every instruction-location l of SCMPDS holds $s \restriction \text{Data-Loc}_{\text{SCM}} = (s +\cdot \text{Start-At}(l)) \restriction \text{Data-Loc}_{\text{SCM}}$.

(8) For every state s of SCMPDS and for every Program-block I and for every instruction-location l of SCMPDS holds $s \restriction \text{Data-Loc}_{\text{SCM}} = (s +\cdot (I +\cdot \text{Start-At}(l))) \restriction \text{Data-Loc}_{\text{SCM}}$.

(9) For every state s of SCMPDS and for every Program-block I holds $s \restriction \text{Data-Loc}_{\text{SCM}} = (s +\cdot \text{Initialized}(I)) \restriction \text{Data-Loc}_{\text{SCM}}$.

(10) Let s be a state of SCMPDS and l be an instruction-location of SCMPDS. Then $\text{dom}(s \restriction$ the instruction locations of SCMPDS) misses $\text{dom}\, \text{Start-At}(l)$.

(11) Let s be a state of SCMPDS, I, J be Program-blocks, and l be an instruction-location of SCMPDS. Then $s +\cdot (I +\cdot \text{Start-At}(l))$ and $s +\cdot (J +\cdot \text{Start-At}(l))$ are equal outside the instruction locations of SCMPDS.

(12) Let s_1, s_2 be states of SCMPDS and I, J be Program-blocks. Suppose $s_1 \restriction \text{Data-Loc}_{\text{SCM}} = s_2 \restriction \text{Data-Loc}_{\text{SCM}}$. Then $s_1 +\cdot \text{Initialized}(I)$ and $s_2 +\cdot \text{Initialized}(J)$ are equal outside the instruction locations of SCMPDS.

(13) Let I be a programmed finite partial state of SCMPDS and x be a set. If $x \in \text{dom}\, I$, then $I(x)$ is an instruction of SCMPDS.

(14) For every state s of SCMPDS and for all instruction-locations l_2, l_3 of SCMPDS holds $s +\cdot \text{Start-At}(l_2) +\cdot \text{Start-At}(l_3) = s +\cdot \text{Start-At}(l_3)$.

(15) $\text{card}(i; I) = \text{card}\, I + 1$.

(16) $(i; I)(\text{inspos}\, 0) = i$.

(17) $I \subseteq \text{Initialized}(\text{stop}\, I)$.

(18) If $l_1 \in \text{dom}\, I$, then $l_1 \in \text{dom}\, \text{stop}\, I$.

(19) If $l_1 \in \text{dom}\, I$, then $(\text{stop}\, I)(l_1) = I(l_1)$.

(20) If $l_1 \in \text{dom}\, I$, then $(\text{Initialized}(\text{stop}\, I))(l_1) = I(l_1)$.

(21) $\mathbf{IC}_{s +\cdot \text{Initialized}(I)} = \text{inspos}\, 0$.

(22) $\text{CurInstr}(s +\cdot \text{Initialized}(\text{stop}\, i; I)) = i$.

(23) For every state s of SCMPDS and for all natural numbers m_1, m_2 such that $\mathbf{IC}_s = \text{inspos}\, m_1$ holds $\text{ICplusConst}(s, m_2) = \text{inspos}\, m_1 + m_2$.

(24) For all Program-blocks I, J holds $\text{Shift}(\text{stop}\, J, \text{card}\, I) \subseteq \text{stop}\, I; J$.

(25) $\text{inspos}\, \text{card}\, I \in \text{dom}\, \text{stop}\, I$ and $(\text{stop}\, I)(\text{inspos}\, \text{card}\, I) = \mathbf{halt}_{\text{SCMPDS}}$.

(26) For all instruction-locations x, l of SCMPDS holds $(\text{IExec}(J, s))(x) = (\text{IExec}(I, s) +\cdot \text{Start-At}(l))(x)$.

(27) For all instruction-locations x, l of SCMPDS holds $(\text{IExec}(I, s))(x) = (s +\cdot \text{Start-At}(l))(x)$.

(28) Let s be a state of SCMPDS, i be a No-StopCode parahalting instruction of SCMPDS, J be a parahalting shiftable Program-block, and a be an Int position. Then $(\text{IExec}(i; J, s))(a) = (\text{IExec}(J, \text{Exec}(i, \text{Initialized}(s))))(a)$.

(29) For every Int position a and for all integers k_1, k_2 holds $(a, k_1) <> 0\_\text{goto}\, k_2 \neq \mathbf{halt}_{\text{SCMPDS}}$.

(30) For every Int position a and for all integers k_1, k_2 holds $(a, k_1) <= 0\_\text{goto}\, k_2 \neq \mathbf{halt}_{\text{SCMPDS}}$.

(31) For every Int position a and for all integers k_1, k_2 holds $(a, k_1) >= 0\_\text{goto}\, k_2 \neq \mathbf{halt}_{\text{SCMPDS}}$.

Let us consider k_1. The functor $\mathrm{Goto}(k_1)$ yielding a Program-block is defined by:

(Def. 1) $\mathrm{Goto}(k_1) = \mathrm{Load}(\mathrm{goto}\ k_1)$.

Let n be a natural number. Observe that goto $(n+1)$ is No-StopCode and goto $(-(n+1))$ is No-StopCode.

Let n be a natural number. Note that $\mathrm{Goto}(n+1)$ is No-StopCode and $\mathrm{Goto}(-(n+1))$ is No-StopCode.

One can prove the following propositions:

(32) $\mathrm{card}\,\mathrm{Goto}(k_1) = 1$.

(33) $\mathrm{inspos}\,0 \in \mathrm{dom}\,\mathrm{Goto}(k_1)$ and $(\mathrm{Goto}(k_1))(\mathrm{inspos}\,0) = \mathrm{goto}\ k_1$.

2. The Predicates of is_closed_on and is_halting_on

Let I be a Program-block and let s be a state of SCMPDS. We say that I is closed on s if and only if:

(Def. 2) For every natural number k holds $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I)))(k)} \in \mathrm{dom}\,\mathrm{stop}\,I$.

We say that I is halting on s if and only if:

(Def. 3) $s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I)$ is halting.

Next we state a number of propositions:

(34) For every Program-block I holds I is paraclosed iff for every state s of SCMPDS holds I is closed on s.

(35) For every Program-block I holds I is parahalting iff for every state s of SCMPDS holds I is halting on s.

(36) Let s_1, s_2 be states of SCMPDS and I be a Program-block. If $s_1 \restriction \mathrm{Data\text{-}Loc}_{\mathrm{SCM}} = s_2 \restriction \mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$, then if I is closed on s_1, then I is closed on s_2.

(37) Let s_1, s_2 be states of SCMPDS and I be a Program-block. Suppose $s_1 \restriction \mathrm{Data\text{-}Loc}_{\mathrm{SCM}} = s_2 \restriction \mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$. Suppose I is closed on s_1 and halting on s_1. Then I is closed on s_2 and halting on s_2.

(38) For every state s of SCMPDS and for all Program-blocks I, J holds I is closed on s iff I is closed on $s+\cdot\,\mathrm{Initialized}(J)$.

(39) Let I, J be Program-blocks and s be a state of SCMPDS. Suppose I is closed on s and halting on s. Then

(i) for every natural number k such that $k \leq \mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I))$ holds $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I)))(k)} = \mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I;\,J)))(k)}$, and

(ii) $(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I)))(\mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I))) \restriction \mathrm{Data\text{-}Loc}_{\mathrm{SCM}} = (\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I;\,J)))(\mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I))) \restriction \mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$.

(40) Let I be a Program-block and k be a natural number. If I is closed on s and halting on s and $k < \mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I))$, then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I)))(k)} \in \mathrm{dom}\,I$.

(41) Let I, J be Program-blocks, s be a state of SCMPDS, and k be a natural number. Suppose I is closed on s and halting on s and $k < \mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I))$. Then $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I)))(k)) = \mathrm{CurInstr}((\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I;\,J)))(k))$.

(42) Let I be a No-StopCode Program-block, s be a state of SCMPDS, and k be a natural number. If I is closed on s and halting on s and $k < \mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I))$, then $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I)))(k)) \neq \mathbf{halt}_{\mathrm{SCMPDS}}$.

(43) Let I be a No-StopCode Program-block and s be a state of SCMPDS. If I is closed on s and halting on s, then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I)))(\mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I)))} = \mathrm{inspos}\,\mathrm{card}\,I$.

(44) Let I, J be Program-blocks and s be a state of SCMPDS. Suppose I is closed on s and halting on s. Then I; Goto$(\mathrm{card}\,J+1)$; J is halting on s and I; Goto$(\mathrm{card}\,J+1)$; J is closed on s.

(45) Let I be a shiftable Program-block. Suppose Initialized$(\mathrm{stop}\,I) \subseteq s_1$ and I is closed on s_1. Let n be a natural number. Suppose Shift$(\mathrm{stop}\,I, n) \subseteq s_2$ and $\mathbf{IC}_{(s_2)} = \mathrm{inspos}\,n$ and $s_1 \restriction \mathrm{Data\text{-}Loc_{SCM}} = s_2 \restriction \mathrm{Data\text{-}Loc_{SCM}}$. Let i be a natural number. Then $\mathbf{IC}_{(\mathrm{Computation}(s_1))(i)} + n = \mathbf{IC}_{(\mathrm{Computation}(s_2))(i)}$ and CurInstr$((\mathrm{Computation}(s_1))(i))$ = CurInstr$((\mathrm{Computation}(s_2))(i))$ and $(\mathrm{Computation}(s_1))(i) \restriction \mathrm{Data\text{-}Loc_{SCM}} = (\mathrm{Computation}(s_2))(i) \restriction \mathrm{Data\text{-}Loc_{SCM}}$.

(46) Let s be a state of SCMPDS, I be a No-StopCode Program-block, and J be a Program-block. If I is closed on s and halting on s, then $\mathbf{IC}_{\mathrm{IExec}(I;\ \mathrm{Goto}(\mathrm{card}\,J+1);\ J, s)} = \mathrm{inspos}\,\mathrm{card}\,I + \mathrm{card}\,J + 1$.

(47) Let s be a state of SCMPDS, I be a No-StopCode Program-block, and J be a Program-block. If I is closed on s and halting on s, then IExec$(I;\ \mathrm{Goto}(\mathrm{card}\,J+1);\ J, s)$ = IExec$(I, s) +\cdot$ Start-At$(\mathrm{inspos}\,\mathrm{card}\,I + \mathrm{card}\,J + 1)$.

(48) Let s be a state of SCMPDS and I be a No-StopCode Program-block. If I is closed on s and halting on s, then $\mathbf{IC}_{\mathrm{IExec}(I,s)} = \mathrm{inspos}\,\mathrm{card}\,I$.

3. The Construction of Conditional Statements

Let a be an Int position, let k be an integer, and let I, J be Program-blocks. The functor **if** $a = k$ **then** I **else** J yielding a Program-block is defined as follows:

(Def. 4) **if** $a = k$ **then** I **else** $J = ((a,k) <> 0\_\mathrm{goto}\ \mathrm{card}\,I + 2)$; I; Goto$(\mathrm{card}\,J+1)$; J.

The functor **if** $a > k$ **then** I **else** J yields a Program-block and is defined as follows:

(Def. 5) **if** $a > k$ **then** I **else** $J = ((a,k) <= 0\_\mathrm{goto}\ \mathrm{card}\,I + 2)$; I; Goto$(\mathrm{card}\,J+1)$; J.

The functor **if** $a < k$ **then** I **else** J yields a Program-block and is defined by:

(Def. 6) **if** $a < k$ **then** I **else** $J = ((a,k) >= 0\_\mathrm{goto}\ \mathrm{card}\,I + 2)$; I; Goto$(\mathrm{card}\,J+1)$; J.

Let a be an Int position, let k be an integer, and let I be a Program-block. The functor **if** $a = 0$ **then** k **else** I yields a Program-block and is defined as follows:

(Def. 7) **if** $a = 0$ **then** k **else** $I = ((a,k) <> 0\_\mathrm{goto}\ \mathrm{card}\,I + 1)$; I.

The functor **if** $a \neq 0$ **then** k **else** I yielding a Program-block is defined by:

(Def. 8) **if** $a \neq 0$ **then** k **else** $I = ((a,k) <> 0\_\mathrm{goto}\,2)$; goto $(\mathrm{card}\,I + 1)$; I.

The functor **if** $a > 0$ **then** k **else** I yielding a Program-block is defined as follows:

(Def. 9) **if** $a > 0$ **then** k **else** $I = ((a,k) <= 0\_\mathrm{goto}\ \mathrm{card}\,I + 1)$; I.

The functor **if** $a \leq 0$ **then** k **else** I yields a Program-block and is defined as follows:

(Def. 10) **if** $a \leq 0$ **then** k **else** $I = ((a,k) <= 0\_\mathrm{goto}\,2)$; goto $(\mathrm{card}\,I + 1)$; I.

The functor **if** $a < 0$ **then** k **else** I yields a Program-block and is defined by:

(Def. 11) **if** $a < 0$ **then** k **else** $I = ((a,k) >= 0\_\mathrm{goto}\ \mathrm{card}\,I + 1)$; I.

The functor **if** $a \geq 0$ **then** k **else** I yielding a Program-block is defined as follows:

(Def. 12) **if** $a \geq 0$ **then** k **else** $I = ((a,k) >= 0\_\mathrm{goto}\,2)$; goto $(\mathrm{card}\,I + 1)$; I.

4. The Computation of "if var=0 then block1 else block2"

One can prove the following propositions:

(49) $\mathrm{card}(\textbf{if } a = k_1 \textbf{ then } I \textbf{ else } J) = \mathrm{card}\, I + \mathrm{card}\, J + 2$.

(50) $\mathrm{inspos}\, 0 \in \mathrm{dom}(\textbf{if } a = k_1 \textbf{ then } I \textbf{ else } J)$ and $\mathrm{inspos}\, 1 \in \mathrm{dom}(\textbf{if } a = k_1 \textbf{ then } I \textbf{ else } J)$.

(51) $(\textbf{if } a = k_1 \textbf{ then } I \textbf{ else } J)(\mathrm{inspos}\, 0) = (a, k_1) <> 0\_\mathrm{goto}\ \mathrm{card}\, I + 2$.

(52) Let s be a state of SCMPDS, I, J be shiftable Program-blocks, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) = 0$ and I is closed on s and halting on s. Then $\textbf{if } a = k_1 \textbf{ then } I \textbf{ else } J$ is closed on s and $\textbf{if } a = k_1 \textbf{ then } I \textbf{ else } J$ is halting on s.

(53) Let s be a state of SCMPDS, I be a Program-block, J be a shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) \neq 0$ and J is closed on s and halting on s. Then $\textbf{if } a = k_1 \textbf{ then } I \textbf{ else } J$ is closed on s and $\textbf{if } a = k_1 \textbf{ then } I \textbf{ else } J$ is halting on s.

(54) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, J be a shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) = 0$ and I is closed on s and halting on s. Then $\mathrm{IExec}(\textbf{if } a = k_1 \textbf{ then } I \textbf{ else } J, s) = \mathrm{IExec}(I, s) +\cdot \text{Start-At}(\mathrm{inspos}\, \mathrm{card}\, I + \mathrm{card}\, J + 2)$.

(55) Let s be a state of SCMPDS, I be a Program-block, J be a No-StopCode shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) \neq 0$ and J is closed on s and halting on s. Then $\mathrm{IExec}(\textbf{if } a = k_1 \textbf{ then } I \textbf{ else } J, s) = \mathrm{IExec}(J, s) +\cdot \text{Start-At}(\mathrm{inspos}\, \mathrm{card}\, I + \mathrm{card}\, J + 2)$.

Let I, J be shiftable parahalting Program-blocks, let a be an Int position, and let k_1 be an integer. Note that $\textbf{if } a = k_1 \textbf{ then } I \textbf{ else } J$ is shiftable and parahalting.

Let I, J be No-StopCode Program-blocks, let a be an Int position, and let k_1 be an integer. Note that $\textbf{if } a = k_1 \textbf{ then } I \textbf{ else } J$ is No-StopCode.

We now state three propositions:

(56) Let s be a state of SCMPDS, I, J be No-StopCode shiftable parahalting Program-blocks, a be an Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\textbf{if } a = k_1 \textbf{ then } I \textbf{ else } J, s)} = \mathrm{inspos}\, \mathrm{card}\, I + \mathrm{card}\, J + 2$.

(57) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, J be a shiftable Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) = 0$, then $(\mathrm{IExec}(\textbf{if } a = k_1 \textbf{ then } I \textbf{ else } J, s))(b) = (\mathrm{IExec}(I, s))(b)$.

(58) Let s be a state of SCMPDS, I be a Program-block, J be a No-StopCode parahalting shiftable Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \neq 0$, then $(\mathrm{IExec}(\textbf{if } a = k_1 \textbf{ then } I \textbf{ else } J, s))(b) = (\mathrm{IExec}(J, s))(b)$.

5. The Computation of "if var=0 then block"

We now state several propositions:

(59) $\mathrm{card}(\textbf{if } a = 0 \textbf{ then } k_1 \textbf{ else } I) = \mathrm{card}\, I + 1$.

(60) $\mathrm{inspos}\, 0 \in \mathrm{dom}(\textbf{if } a = 0 \textbf{ then } k_1 \textbf{ else } I)$.

(61) $(\textbf{if } a = 0 \textbf{ then } k_1 \textbf{ else } I)(\mathrm{inspos}\, 0) = (a, k_1) <> 0\_\mathrm{goto}\ \mathrm{card}\, I + 1$.

(62) Let s be a state of SCMPDS, I be a shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) = 0$ and I is closed on s and halting on s. Then $\textbf{if } a = 0 \textbf{ then } k_1 \textbf{ else } I$ is closed on s and $\textbf{if } a = 0 \textbf{ then } k_1 \textbf{ else } I$ is halting on s.

(63) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) \neq 0$, then **if** $a = 0$ **then** k_1 **else** I is closed on s and **if** $a = 0$ **then** k_1 **else** I is halting on s.

(64) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a),k_1)) = 0$ and I is closed on s and halting on s. Then IExec(**if** $a = 0$ **then** k_1 **else** $I, s) = \mathrm{IExec}(I,s) +\cdot \mathrm{Start\text{-}At}(\mathrm{inspos}\,\mathrm{card}\,I + 1)$.

(65) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) \neq 0$, then IExec(**if** $a = 0$ **then** k_1 **else** $I, s) = s +\cdot \mathrm{Start\text{-}At}(\mathrm{inspos}\,\mathrm{card}\,I + 1)$.

Let I be a shiftable parahalting Program-block, let a be an Int position, and let k_1 be an integer. One can verify that **if** $a = 0$ **then** k_1 **else** I is shiftable and parahalting.

Let I be a No-StopCode Program-block, let a be an Int position, and let k_1 be an integer. Observe that **if** $a = 0$ **then** k_1 **else** I is No-StopCode.

Next we state three propositions:

(66) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a be an Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\mathbf{if}\ a=0\ \mathbf{then}\ k_1\ \mathbf{else}\ I,s)} = \mathrm{inspos}\,\mathrm{card}\,I + 1$.

(67) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) = 0$, then (IExec(**if** $a = 0$ **then** k_1 **else** $I, s))(b) = (\mathrm{IExec}(I,s))(b)$.

(68) Let s be a state of SCMPDS, I be a Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) \neq 0$, then (IExec(**if** $a = 0$ **then** k_1 **else** $I, s))(b) = s(b)$.

6. THE COMPUTATION OF "IF VAR¡¿0 THEN BLOCK"

Next we state several propositions:

(69) card(**if** $a \neq 0$ **then** k_1 **else** $I) = \mathrm{card}\,I + 2$.

(70) $\mathrm{inspos}\,0 \in$ dom(**if** $a \neq 0$ **then** k_1 **else** I) and $\mathrm{inspos}\,1 \in$ dom(**if** $a \neq 0$ **then** k_1 **else** I).

(71) (**if** $a \neq 0$ **then** k_1 **else** $I)(\mathrm{inspos}\,0) = (a,k_1) <> 0\_\mathrm{goto}\,2$ and (**if** $a \neq 0$ **then** k_1 **else** $I)(\mathrm{inspos}\,1) =$ goto $(\mathrm{card}\,I + 1)$.

(72) Let s be a state of SCMPDS, I be a shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a),k_1)) \neq 0$ and I is closed on s and halting on s. Then **if** $a \neq 0$ **then** k_1 **else** I is closed on s and **if** $a \neq 0$ **then** k_1 **else** I is halting on s.

(73) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) = 0$, then **if** $a \neq 0$ **then** k_1 **else** I is closed on s and **if** $a \neq 0$ **then** k_1 **else** I is halting on s.

(74) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a),k_1)) \neq 0$ and I is closed on s and halting on s. Then IExec(**if** $a \neq 0$ **then** k_1 **else** $I, s) = \mathrm{IExec}(I,s) +\cdot \mathrm{Start\text{-}At}(\mathrm{inspos}\,\mathrm{card}\,I + 2)$.

(75) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) = 0$, then IExec(**if** $a \neq 0$ **then** k_1 **else** $I, s) = s +\cdot \mathrm{Start\text{-}At}(\mathrm{inspos}\,\mathrm{card}\,I + 2)$.

Let I be a shiftable parahalting Program-block, let a be an Int position, and let k_1 be an integer. Observe that **if** $a \neq 0$ **then** k_1 **else** I is shiftable and parahalting.

Let I be a No-StopCode Program-block, let a be an Int position, and let k_1 be an integer. One can check that **if** $a \neq 0$ **then** k_1 **else** I is No-StopCode.

One can prove the following three propositions:

(76) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a be an Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\mathbf{if}\ a \neq 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I,s)} = \mathrm{inspos}\,\mathrm{card}\,I+2$.

(77) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) \neq 0$, then $(\mathrm{IExec}(\mathbf{if}\ a \neq 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I,s))(b) = (\mathrm{IExec}(I,s))(b)$.

(78) Let s be a state of SCMPDS, I be a Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) = 0$, then $(\mathrm{IExec}(\mathbf{if}\ a \neq 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I,s))(b) = s(b)$.

7. THE COMPUTATION OF "IF VAR¿0 THEN BLOCK1 ELSE BLOCK2"

We now state several propositions:

(79) $\mathrm{card}(\mathbf{if}\ a > k_1\ \mathbf{then}\ I\ \mathbf{else}\ J) = \mathrm{card}\,I + \mathrm{card}\,J + 2$.

(80) $\mathrm{inspos}\,0 \in \mathrm{dom}(\mathbf{if}\ a > k_1\ \mathbf{then}\ I\ \mathbf{else}\ J)$ and $\mathrm{inspos}\,1 \in \mathrm{dom}(\mathbf{if}\ a > k_1\ \mathbf{then}\ I\ \mathbf{else}\ J)$.

(81) $(\mathbf{if}\ a > k_1\ \mathbf{then}\ I\ \mathbf{else}\ J)(\mathrm{inspos}\,0) = (a,k_1) <= 0\_\mathrm{goto}\ \mathrm{card}\,I+2$.

(82) Let s be a state of SCMPDS, I, J be shiftable Program-blocks, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a),k_1)) > 0$ and I is closed on s and halting on s. Then $\mathbf{if}\ a > k_1\ \mathbf{then}\ I\ \mathbf{else}\ J$ is closed on s and $\mathbf{if}\ a > k_1\ \mathbf{then}\ I\ \mathbf{else}\ J$ is halting on s.

(83) Let s be a state of SCMPDS, I be a Program-block, J be a shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a),k_1)) \leq 0$ and J is closed on s and halting on s. Then $\mathbf{if}\ a > k_1\ \mathbf{then}\ I\ \mathbf{else}\ J$ is closed on s and $\mathbf{if}\ a > k_1\ \mathbf{then}\ I\ \mathbf{else}\ J$ is halting on s.

(84) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, J be a shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a),k_1)) > 0$ and I is closed on s and halting on s. Then $\mathrm{IExec}(\mathbf{if}\ a > k_1\ \mathbf{then}\ I\ \mathbf{else}\ J,s) = \mathrm{IExec}(I,s) +\cdot\ \text{Start-At}(\mathrm{inspos}\,\mathrm{card}\,I + \mathrm{card}\,J + 2)$.

(85) Let s be a state of SCMPDS, I be a Program-block, J be a No-StopCode shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a),k_1)) \leq 0$ and J is closed on s and halting on s. Then $\mathrm{IExec}(\mathbf{if}\ a > k_1\ \mathbf{then}\ I\ \mathbf{else}\ J,s) = \mathrm{IExec}(J,s) +\cdot\ \text{Start-At}(\mathrm{inspos}\,\mathrm{card}\,I + \mathrm{card}\,J + 2)$.

Let I, J be shiftable parahalting Program-blocks, let a be an Int position, and let k_1 be an integer. Observe that $\mathbf{if}\ a > k_1\ \mathbf{then}\ I\ \mathbf{else}\ J$ is shiftable and parahalting.

Let I, J be No-StopCode Program-blocks, let a be an Int position, and let k_1 be an integer. Note that $\mathbf{if}\ a > k_1\ \mathbf{then}\ I\ \mathbf{else}\ J$ is No-StopCode.

The following three propositions are true:

(86) Let s be a state of SCMPDS, I, J be No-StopCode shiftable parahalting Program-blocks, a be an Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\mathbf{if}\ a>k_1\ \mathbf{then}\ I\ \mathbf{else}\ J,s)} = \mathrm{inspos}\,\mathrm{card}\,I + \mathrm{card}\,J + 2$.

(87) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, J be a shiftable Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) > 0$, then $(\mathrm{IExec}(\mathbf{if}\ a > k_1\ \mathbf{then}\ I\ \mathbf{else}\ J,s))(b) = (\mathrm{IExec}(I,s))(b)$.

(88) Let s be a state of SCMPDS, I be a Program-block, J be a No-StopCode parahalting shiftable Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) \leq 0$, then $(\mathrm{IExec}(\mathbf{if}\ a > k_1\ \mathbf{then}\ I\ \mathbf{else}\ J,s))(b) = (\mathrm{IExec}(J,s))(b)$.

8. The Computation of "if var¿0 then block"

Next we state several propositions:

(89) card(**if** $a > 0$ **then** k_1 **else** I) $=$ card $I+1$.

(90) inspos $0 \in$ dom(**if** $a > 0$ **then** k_1 **else** I).

(91) (**if** $a > 0$ **then** k_1 **else** I)(inspos 0) $= (a, k_1) <= 0$_goto card $I+1$.

(92) Let s be a state of SCMPDS, I be a shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) > 0$ and I is closed on s and halting on s. Then **if** $a > 0$ **then** k_1 **else** I is closed on s and **if** $a > 0$ **then** k_1 **else** I is halting on s.

(93) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \leq 0$, then **if** $a > 0$ **then** k_1 **else** I is closed on s and **if** $a > 0$ **then** k_1 **else** I is halting on s.

(94) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) > 0$ and I is closed on s and halting on s. Then IExec(**if** $a > 0$ **then** k_1 **else** I, s) $=$ IExec$(I, s) +\cdot$ Start-At(inspos card $I+1$).

(95) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \leq 0$, then IExec(**if** $a > 0$ **then** k_1 **else** I, s) $=$ $s +\cdot$ Start-At(inspos card $I+1$).

Let I be a shiftable parahalting Program-block, let a be an Int position, and let k_1 be an integer. Observe that **if** $a > 0$ **then** k_1 **else** I is shiftable and parahalting.

Let I be a No-StopCode Program-block, let a be an Int position, and let k_1 be an integer. Observe that **if** $a > 0$ **then** k_1 **else** I is No-StopCode.

We now state three propositions:

(96) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a be an Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\mathbf{if}\ a>0\ \mathbf{then}\ k_1\ \mathbf{else}\ I, s)} =$ inspos card $I+1$.

(97) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) > 0$, then (IExec(**if** $a > 0$ **then** k_1 **else** I, s))$(b) =$ (IExec(I, s))(b).

(98) Let s be a state of SCMPDS, I be a Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \leq 0$, then (IExec(**if** $a > 0$ **then** k_1 **else** I, s))$(b) = s(b)$.

9. The Computation of "if var¡=0 then block"

Next we state several propositions:

(99) card(**if** $a \leq 0$ **then** k_1 **else** I) $=$ card $I+2$.

(100) inspos $0 \in$ dom(**if** $a \leq 0$ **then** k_1 **else** I) and inspos $1 \in$ dom(**if** $a \leq 0$ **then** k_1 **else** I).

(101) (**if** $a \leq 0$ **then** k_1 **else** I)(inspos 0) $= (a, k_1) <= 0$_goto 2 and (**if** $a \leq 0$ **then** k_1 **else** I)(inspos 1) $=$ goto (card $I+1$).

(102) Let s be a state of SCMPDS, I be a shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) \leq 0$ and I is closed on s and halting on s. Then **if** $a \leq 0$ **then** k_1 **else** I is closed on s and **if** $a \leq 0$ **then** k_1 **else** I is halting on s.

(103) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) > 0$, then **if** $a \leq 0$ **then** k_1 **else** I is closed on s and **if** $a \leq 0$ **then** k_1 **else** I is halting on s.

(104) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\text{DataLoc}(s(a),k_1)) \leq 0$ and I is closed on s and halting on s. Then $\text{IExec}(\textbf{if } a \leq 0 \textbf{ then } k_1 \textbf{ else } I, s) = \text{IExec}(I,s) +\cdot \text{Start-At}(\text{inspos}\,\text{card}\,I+2)$.

(105) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and k_1 be an integer. If $s(\text{DataLoc}(s(a),k_1)) > 0$, then $\text{IExec}(\textbf{if } a \leq 0 \textbf{ then } k_1 \textbf{ else } I, s) = s +\cdot \text{Start-At}(\text{inspos}\,\text{card}\,I+2)$.

Let I be a shiftable parahalting Program-block, let a be an Int position, and let k_1 be an integer. Note that **if** $a \leq 0$ **then** k_1 **else** I is shiftable and parahalting.

Let I be a No-StopCode Program-block, let a be an Int position, and let k_1 be an integer. Note that **if** $a \leq 0$ **then** k_1 **else** I is No-StopCode.

One can prove the following three propositions:

(106) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a be an Int position, and k_1 be an integer. Then $\mathbf{IC}_{\text{IExec}(\textbf{if } a \leq 0 \textbf{ then } k_1 \textbf{ else } I, s)} = \text{inspos}\,\text{card}\,I+2$.

(107) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a, b be Int positions, and k_1 be an integer. If $s(\text{DataLoc}(s(a),k_1)) \leq 0$, then $(\text{IExec}(\textbf{if } a \leq 0 \textbf{ then } k_1 \textbf{ else } I, s))(b) = (\text{IExec}(I,s))(b)$.

(108) Let s be a state of SCMPDS, I be a Program-block, a, b be Int positions, and k_1 be an integer. If $s(\text{DataLoc}(s(a),k_1)) > 0$, then $(\text{IExec}(\textbf{if } a \leq 0 \textbf{ then } k_1 \textbf{ else } I, s))(b) = s(b)$.

10. The Computation of “if var¡0 then block1 else block2”

Next we state several propositions:

(109) $\text{card}(\textbf{if } a < k_1 \textbf{ then } I \textbf{ else } J) = \text{card}\,I + \text{card}\,J + 2$.

(110) $\text{inspos}\,0 \in \text{dom}(\textbf{if } a < k_1 \textbf{ then } I \textbf{ else } J)$ and $\text{inspos}\,1 \in \text{dom}(\textbf{if } a < k_1 \textbf{ then } I \textbf{ else } J)$.

(111) $(\textbf{if } a < k_1 \textbf{ then } I \textbf{ else } J)(\text{inspos}\,0) = (a,k_1) >= 0\_\text{goto}\ \text{card}\,I+2$.

(112) Let s be a state of SCMPDS, I, J be shiftable Program-blocks, a be an Int position, and k_1 be an integer. Suppose $s(\text{DataLoc}(s(a),k_1)) < 0$ and I is closed on s and halting on s. Then **if** $a < k_1$ **then** I **else** J is closed on s and **if** $a < k_1$ **then** I **else** J is halting on s.

(113) Let s be a state of SCMPDS, I be a Program-block, J be a shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\text{DataLoc}(s(a),k_1)) \geq 0$ and J is closed on s and halting on s. Then **if** $a < k_1$ **then** I **else** J is closed on s and **if** $a < k_1$ **then** I **else** J is halting on s.

(114) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, J be a shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\text{DataLoc}(s(a),k_1)) < 0$ and I is closed on s and halting on s. Then $\text{IExec}(\textbf{if } a < k_1 \textbf{ then } I \textbf{ else } J, s) = \text{IExec}(I,s) +\cdot \text{Start-At}(\text{inspos}\,\text{card}\,I + \text{card}\,J + 2)$.

(115) Let s be a state of SCMPDS, I be a Program-block, J be a No-StopCode shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\text{DataLoc}(s(a),k_1)) \geq 0$ and J is closed on s and halting on s. Then $\text{IExec}(\textbf{if } a < k_1 \textbf{ then } I \textbf{ else } J, s) = \text{IExec}(J,s) +\cdot \text{Start-At}(\text{inspos}\,\text{card}\,I + \text{card}\,J + 2)$.

Let I, J be shiftable parahalting Program-blocks, let a be an Int position, and let k_1 be an integer. One can check that **if** $a < k_1$ **then** I **else** J is shiftable and parahalting.

Let I, J be No-StopCode Program-blocks, let a be an Int position, and let k_1 be an integer. Note that **if** $a < k_1$ **then** I **else** J is No-StopCode.

The following propositions are true:

(116) Let s be a state of SCMPDS, I, J be No-StopCode shiftable parahalting Program-blocks, a be an Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\mathbf{if}\ a<k_1\ \mathbf{then}\ I\ \mathbf{else}\ J,s)} = \mathrm{inspos}\,\mathrm{card}\,I + \mathrm{card}\,J + 2$.

(117) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, J be a shiftable Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) < 0$, then $(\mathrm{IExec}(\mathbf{if}\ a < k_1\ \mathbf{then}\ I\ \mathbf{else}\ J,s))(b) = (\mathrm{IExec}(I,s))(b)$.

(118) Let s be a state of SCMPDS, I be a Program-block, J be a No-StopCode parahalting shiftable Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) \geq 0$, then $(\mathrm{IExec}(\mathbf{if}\ a < k_1\ \mathbf{then}\ I\ \mathbf{else}\ J,s))(b) = (\mathrm{IExec}(J,s))(b)$.

11. THE COMPUTATION OF "IF VAR¡0 THEN BLOCK"

Next we state several propositions:

(119) $\mathrm{card}(\mathbf{if}\ a < 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I) = \mathrm{card}\,I + 1$.

(120) $\mathrm{inspos}\,0 \in \mathrm{dom}(\mathbf{if}\ a < 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I)$.

(121) $(\mathbf{if}\ a < 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I)(\mathrm{inspos}\,0) = (a,k_1) >= 0\_\mathrm{goto}\,\mathrm{card}\,I + 1$.

(122) Let s be a state of SCMPDS, I be a shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a),k_1)) < 0$ and I is closed on s and halting on s. Then $\mathbf{if}\ a < 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I$ is closed on s and $\mathbf{if}\ a < 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I$ is halting on s.

(123) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) \geq 0$, then $\mathbf{if}\ a < 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I$ is closed on s and $\mathbf{if}\ a < 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I$ is halting on s.

(124) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a),k_1)) < 0$ and I is closed on s and halting on s. Then $\mathrm{IExec}(\mathbf{if}\ a < 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I,s) = \mathrm{IExec}(I,s) +\cdot\ \mathrm{Start\text{-}At}(\mathrm{inspos}\,\mathrm{card}\,I + 1)$.

(125) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) \geq 0$, then $\mathrm{IExec}(\mathbf{if}\ a < 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I,s) = s +\cdot\ \mathrm{Start\text{-}At}(\mathrm{inspos}\,\mathrm{card}\,I + 1)$.

Let I be a shiftable parahalting Program-block, let a be an Int position, and let k_1 be an integer. Observe that $\mathbf{if}\ a < 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I$ is shiftable and parahalting.

Let I be a No-StopCode Program-block, let a be an Int position, and let k_1 be an integer. Note that $\mathbf{if}\ a < 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I$ is No-StopCode.

One can prove the following three propositions:

(126) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a be an Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\mathbf{if}\ a<0\ \mathbf{then}\ k_1\ \mathbf{else}\ I,s)} = \mathrm{inspos}\,\mathrm{card}\,I + 1$.

(127) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) < 0$, then $(\mathrm{IExec}(\mathbf{if}\ a < 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I,s))(b) = (\mathrm{IExec}(I,s))(b)$.

(128) Let s be a state of SCMPDS, I be a Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a),k_1)) \geq 0$, then $(\mathrm{IExec}(\mathbf{if}\ a < 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I,s))(b) = s(b)$.

12. THE COMPUTATION OF "IF VAR¿=0 THEN BLOCK"

Next we state several propositions:

(129) $\mathrm{card}(\textbf{if } a \geq 0 \textbf{ then } k_1 \textbf{ else } I) = \mathrm{card}\, I + 2$.

(130) $\mathrm{inspos}\, 0 \in \mathrm{dom}(\textbf{if } a \geq 0 \textbf{ then } k_1 \textbf{ else } I)$ and $\mathrm{inspos}\, 1 \in \mathrm{dom}(\textbf{if } a \geq 0 \textbf{ then } k_1 \textbf{ else } I)$.

(131) $(\textbf{if } a \geq 0 \textbf{ then } k_1 \textbf{ else } I)(\mathrm{inspos}\, 0) = (a, k_1) >= 0\_\mathrm{goto}\, 2$ and $(\textbf{if } a \geq 0 \textbf{ then } k_1 \textbf{ else } I)(\mathrm{inspos}\, 1) = \mathrm{goto}\, (\mathrm{card}\, I + 1)$.

(132) Let s be a state of SCMPDS, I be a shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) \geq 0$ and I is closed on s and halting on s. Then **if** $a \geq 0$ **then** k_1 **else** I is closed on s and **if** $a \geq 0$ **then** k_1 **else** I is halting on s.

(133) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) < 0$, then **if** $a \geq 0$ **then** k_1 **else** I is closed on s and **if** $a \geq 0$ **then** k_1 **else** I is halting on s.

(134) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) \geq 0$ and I is closed on s and halting on s. Then $\mathrm{IExec}(\textbf{if } a \geq 0 \textbf{ then } k_1 \textbf{ else } I, s) = \mathrm{IExec}(I, s) +\cdot \mathrm{Start\text{-}At}(\mathrm{inspos}\, \mathrm{card}\, I + 2)$.

(135) Let s be a state of SCMPDS, I be a Program-block, a be an Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) < 0$, then $\mathrm{IExec}(\textbf{if } a \geq 0 \textbf{ then } k_1 \textbf{ else } I, s) = s +\cdot \mathrm{Start\text{-}At}(\mathrm{inspos}\, \mathrm{card}\, I + 2)$.

Let I be a shiftable parahalting Program-block, let a be an Int position, and let k_1 be an integer. One can verify that **if** $a \geq 0$ **then** k_1 **else** I is shiftable and parahalting.

Let I be a No-StopCode Program-block, let a be an Int position, and let k_1 be an integer. One can check that **if** $a \geq 0$ **then** k_1 **else** I is No-StopCode.

We now state three propositions:

(136) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a be an Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\textbf{if } a \geq 0 \textbf{ then } k_1 \textbf{ else } I, s)} = \mathrm{inspos}\, \mathrm{card}\, I + 2$.

(137) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \geq 0$, then $(\mathrm{IExec}(\textbf{if } a \geq 0 \textbf{ then } k_1 \textbf{ else } I, s))(b) = (\mathrm{IExec}(I, s))(b)$.

(138) Let s be a state of SCMPDS, I be a Program-block, a, b be Int positions, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) < 0$, then $(\mathrm{IExec}(\textbf{if } a \geq 0 \textbf{ then } k_1 \textbf{ else } I, s))(b) = s(b)$.

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 and Piotr Rudnicki. Development of terminology for **scm**. *Journal of Formalized Mathematics*, 5, 1993. http://mizar.org/JFM/Vol5/scm_1.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. 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

[13] Jan Popiołek. Some properties of functions modul and signum. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/absvalue.html.

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

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

[16] 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.

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

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

Received June 15, 1999

Published January 2, 2004