FORMALIZED MATHEMATICS
Vol. 16, No. 4, Pages 339–353, 2008

Model Checking. Part III

Kazuhisa Ishida
Shinshu University
Nagano, Japan

Yasunari Shidama
Shinshu University
Nagano, Japan

Summary. This text includes verification of the basic algorithm in Simple On-the-fly Automatic Verification of Linear Temporal Logic (LTL). LTL formula can be transformed to Buchi automaton, and this transforming algorithm is mainly used at Simple On-the-fly Automatic Verification. In this article, we verified the transforming algorithm itself. At first, we prepared some definitions and operations for transforming. And then, we defined the Buchi automaton and verified the transforming algorithm.

MML identifier: MODELC_3, version: 7.9.03 4.108.1028

The notation and terminology used in this paper are introduced in the following articles: [5], [14], [6], [7], [1], [15], [3], [16], [2], [13], [4], [12], [10], [11], [8], and [9].

1. Definition of Basic Operations to Build an Automaton for LTL and Properties

For simplicity, we adopt the following rules: k, n, m, i, j are elements of $\mathbb{N}$, x, y, X are sets, L, L_1, L_2 are finite sequences, F, H are LTL-formulae, W, W_1, W_2 are subsets of Subformulae H, and v is an LTL-formula.

Let us consider F. Then Subformulae F is a subset of $\mathrm{WFF}_{\mathrm{LTL}}$.

Let us consider H. The functor $\mathrm{LTLNew}_1\, H$ yields a subset of Subformulae H and is defined as follows:

(Def. 1) $$\mathrm{LTLNew}_1\, H = \begin{cases} \{\mathrm{LeftArg}(H), \mathrm{RightArg}(H)\}, & \text{if } H \text{ is conjunctive,} \\ \{\mathrm{LeftArg}(H)\}, & \text{if } H \text{ is disjunctive,} \\ \emptyset, & \text{if } H \text{ has } \textit{next} \text{ operator,} \\ \{\mathrm{LeftArg}(H)\}, & \text{if } H \text{ has } \textit{until} \text{ operator,} \\ \{\mathrm{RightArg}(H)\}, & \text{if } H \text{ has } \textit{release} \text{ operator,} \\ \emptyset, & \text{otherwise.} \end{cases}$$

ISSN 1426–2630(p), 1898-9934(e)

The functor $\mathrm{LTLNew}_2\, H$ yields a subset of $\mathrm{Subformulae}\, H$ and is defined as follows:

(Def. 2) $\mathrm{LTLNew}_2\, H = \begin{cases} \emptyset, \text{ if } H \text{ is conjunctive}, \\ \{\mathrm{RightArg}(H)\}, \text{ if } H \text{ is disjunctive}, \\ \emptyset, \text{ if } H \text{ has } \textit{next} \text{ operator}, \\ \{\mathrm{RightArg}(H)\}, \text{ if } H \text{ has } \textit{until} \text{ operator}, \\ \{\mathrm{LeftArg}(H), \mathrm{RightArg}(H)\}, \text{ if } H \text{ has } \textit{release} \text{ operator}, \\ \emptyset, \text{ otherwise}. \end{cases}$

The functor $\mathrm{LTLNext}\, H$ yielding a subset of $\mathrm{Subformulae}\, H$ is defined as follows:

(Def. 3) $\mathrm{LTLNext}\, H = \begin{cases} \emptyset, \text{ if } H \text{ is conjunctive}, \\ \emptyset, \text{ if } H \text{ is disjunctive}, \\ \{\mathrm{Arg}(H)\}, \text{ if } H \text{ has } \textit{next} \text{ operator}, \\ \{H\}, \text{ if } H \text{ has } \textit{until} \text{ operator}, \\ \{H\}, \text{ if } H \text{ has } \textit{release} \text{ operator}, \\ \emptyset, \text{ otherwise}. \end{cases}$

Let us consider v. We consider LTL-nodes over v as systems

$\langle$ an old-component, a new-component, a next-component $\rangle$,

where the old-component, the new-component, and the next-component are subsets of $\mathrm{Subformulae}\, v$.

Let us consider v, let N be an LTL-node over v, and let us consider H. Let us assume that $H \in$ the new-component of N. The functor $\mathrm{SuccNode}_1(H, N)$ yielding a strict LTL-node over v is defined by the conditions (Def. 4).

(Def. 4)(i) The old-component of $\mathrm{SuccNode}_1(H, N)$ = (the old-component of N) $\cup \{H\}$,

(ii) the new-component of $\mathrm{SuccNode}_1(H, N)$ = ((the new-component of N) $\setminus \{H\}$) $\cup$ ($\mathrm{LTLNew}_1\, H \setminus$ the old-component of N), and

(iii) the next-component of $\mathrm{SuccNode}_1(H, N)$ = (the next-component of N) $\cup\, \mathrm{LTLNext}\, H$.

Let us consider v, let N be an LTL-node over v, and let us consider H. Let us assume that $H \in$ the new-component of N and H is either disjunctive or has *until* operator or *release* operator. The functor $\mathrm{SuccNode}_2(H, N)$ yields a strict LTL-node over v and is defined by the conditions (Def. 5).

(Def. 5)(i) The old-component of $\mathrm{SuccNode}_2(H, N)$ = (the old-component of N) $\cup \{H\}$,

(ii) the new-component of $\mathrm{SuccNode}_2(H, N)$ = ((the new-component of N) $\setminus \{H\}$) $\cup$ ($\mathrm{LTLNew}_2\, H \setminus$ the old-component of N), and

(iii) the next-component of $\mathrm{SuccNode}_2(H, N)$ = the next-component of N.

Let us consider v, let N_1, N_2 be LTL-nodes over v, and let us consider H. We say that N_2 is a successor of N_1 and H if and only if the conditions (Def. 6) are satisfied.

(Def. 6)(i) $H \in$ the new-component of N_1, and

(ii) $N_2 = \mathrm{SuccNode}_1(H, N_1)$ or H is either disjunctive or has *until* operator or *release* operator and $N_2 = \mathrm{SuccNode}_2(H, N_1)$.

Let us consider v and let N_1, N_2 be LTL-nodes over v. We say that N_2 is a 1st successor of N_1 if and only if:

(Def. 7) There exists H such that $H \in$ the new-component of N_1 and $N_2 = \mathrm{SuccNode}_1(H, N_1)$.

We say that N_2 is a 2nd successor of N_1 if and only if the condition (Def. 8) is satisfied.

(Def. 8) There exists H such that

(i) $H \in$ the new-component of N_1,

(ii) H is either disjunctive or has *until* operator or *release* operator, and

(iii) $N_2 = \mathrm{SuccNode}_2(H, N_1)$.

Let us consider v and let N_1, N_2 be LTL-nodes over v. We say that N_2 is a successor of N_1 if and only if:

(Def. 9) N_2 is a 1st successor of N_1 or a 2nd successor of N_1.

Let us consider v and let N be an LTL-node over v. We say that N is failure if and only if:

(Def. 10) There exist H, F such that H is atomic and $F = \neg H$ and $H \in$ the old-component of N and $F \in$ the old-component of N.

Let us consider v and let N be an LTL-node over v. We say that N is elementary if and only if:

(Def. 11) The new-component of $N = \emptyset$.

Let us consider v and let N be an LTL-node over v. We say that N is final if and only if:

(Def. 12) N is elementary and the next-component of $N = \emptyset$.

Let us consider v. The functor $\emptyset_v$ yielding a subset of Subformulae v is defined as follows:

(Def. 13) $\emptyset_v = \emptyset$.

Let us consider v. The functor Seed v yielding a subset of Subformulae v is defined by:

(Def. 14) $\mathrm{Seed}\, v = \{v\}$.

Let us consider v. Note that there exists an LTL-node over v which is elementary and strict.

Let us consider v. The functor FinalNode v yields an elementary strict LTL-node over v and is defined by:

(Def. 15) $\mathrm{FinalNode}\, v = \langle \emptyset_v, \emptyset_v, \emptyset_v \rangle$.

Let us consider x, v. The functor CastNode(x, v) yields a strict LTL-node over v and is defined by:

(Def. 16) $\mathrm{CastNode}(x, v) = \begin{cases} x, \text{ if } x \text{ is a strict LTL-node over } v, \\ \langle \emptyset_v, \emptyset_v, \emptyset_v \rangle, \text{ otherwise.} \end{cases}$

Let us consider v. The functor $\mathrm{init}\, v$ yields an elementary strict LTL-node over v and is defined by:

(Def. 17) $\mathrm{init}\, v = \langle \emptyset_v, \emptyset_v, \mathrm{Seed}\, v \rangle$.

Let us consider v and let N be an LTL-node over v. The functor $\mathcal{X}\, N$ yields a strict LTL-node over v and is defined as follows:

(Def. 18) $\mathcal{X}\, N = \langle \emptyset_v, \text{the next-component of } N, \emptyset_v \rangle$.

We follow the rules: N, N_1, N_2, M are strict LTL-nodes over v and w is an element of the infinite sequences of AtomicFamily.

Let us consider v, L. We say that L is a successor sequence for v if and only if:

(Def. 19) For every k such that $1 \leq k < \mathrm{len}\, L$ there exist N, M such that $N = L(k)$ and $M = L(k+1)$ and M is a successor of N.

Let us consider v, N_1, N_2. We say that N_2 is next to N_1 if and only if the conditions (Def. 20) are satisfied.

(Def. 20)(i) N_1 is elementary,

(ii) N_2 is elementary, and

(iii) there exists L such that $1 \leq \mathrm{len}\, L$ and L is a successor sequence for v and $L(1) = \mathcal{X}\, N_1$ and $L(\mathrm{len}\, L) = N_2$.

Let us consider v and let W be a subset of $\mathrm{Subformulae}\, v$. The functor $\mathrm{Cast}_{\mathrm{LTL}}\, W$ yielding a subset of $\mathrm{WFF}_{\mathrm{LTL}}$ is defined by:

(Def. 21) $\mathrm{Cast}_{\mathrm{LTL}}\, W = W$.

Let us consider v, N. The functor $\cdot N$ yields a subset of $\mathrm{WFF}_{\mathrm{LTL}}$ and is defined by:

(Def. 22) $\cdot N = (\text{the old-component of } N) \cup (\text{the new-component of } N) \cup \mathcal{X}\, \mathrm{Cast}_{\mathrm{LTL}}\, (\text{the next-component of } N)$.

We now state three propositions:

(1) Suppose $H \in$ the new-component of N and H is either atomic, or negative, or conjunctive, or has *next* operator. Then $w \models \cdot N$ if and only if $w \models \cdot \mathrm{SuccNode}_1(H, N)$.

(2) Suppose $H \in$ the new-component of N and H is either disjunctive or has *until* operator or *release* operator. Then $w \models \cdot N$ if and only if one of the following conditions is satisfied:

(i) $w \models \cdot \mathrm{SuccNode}_1(H, N)$, or

(ii) $w \models \cdot \mathrm{SuccNode}_2(H, N)$.

(3) There exists L such that $\mathrm{Subformulae}\, H = \mathrm{rng}\, L$.

Let us consider H. Observe that $\mathrm{Subformulae}\, H$ is finite.

Let us consider H, W, L, x. The length of L wrt W and x yields a natural number and is defined as follows:

(Def. 23) The length of L wrt W and $x = \begin{cases} \operatorname{len} \operatorname{Cast}_{\mathrm{LTL}} L(x), & \text{if } L(x) \in W, \\ 0, & \text{otherwise.} \end{cases}$

Let us consider H, W, L. The partial sequence of L wrt W yields a sequence of real numbers and is defined by the condition (Def. 24).

(Def. 24) Let given k. Then

(i) if $L(k) \in W$, then (the partial sequence of L wrt W)$(k) = \operatorname{len} \operatorname{Cast}_{\mathrm{LTL}} L(k)$, and

(ii) if $L(k) \notin W$, then (the partial sequence of L wrt W)$(k) = 0$.

Let us consider H, W, L. The functor $\operatorname{len}(L, W)$ yields a real number and is defined as follows:

(Def. 25) $\operatorname{len}(L, W) = \sum_{\kappa=0}^{\operatorname{len} L}$ (the partial sequence of L wrt W)(κ).

We now state several propositions:

(4) $\operatorname{len}(L, \emptyset_H) = 0$.

(5) If $F \notin W$, then $\operatorname{len}(L, W \setminus \{F\}) = \operatorname{len}(L, W)$.

(6) If $\operatorname{rng} L = \operatorname{Subformulae} H$ and L is one-to-one and $F \in W$, then $\operatorname{len}(L, W \setminus \{F\}) = \operatorname{len}(L, W) - \operatorname{len} F$.

(7) If $\operatorname{rng} L = \operatorname{Subformulae} H$ and L is one-to-one and $F \notin W$ and $W_1 = W \cup \{F\}$, then $\operatorname{len}(L, W_1) = \operatorname{len}(L, W) + \operatorname{len} F$.

(8) If $\operatorname{rng} L_1 = \operatorname{Subformulae} H$ and L_1 is one-to-one and $\operatorname{rng} L_2 = \operatorname{Subformulae} H$ and L_2 is one-to-one, then $\operatorname{len}(L_1, W) = \operatorname{len}(L_2, W)$.

Let us consider H, W. The functor $\operatorname{len} W$ yields a real number and is defined by:

(Def. 26) There exists L such that $\operatorname{rng} L = \operatorname{Subformulae} H$ and L is one-to-one and $\operatorname{len} W = \operatorname{len}(L, W)$.

The following propositions are true:

(9) If $F \notin W$, then $\operatorname{len}(W \setminus \{F\}) = \operatorname{len} W$.

(10) If $F \in W$, then $\operatorname{len}(W \setminus \{F\}) = \operatorname{len} W - \operatorname{len} F$.

(11) If $F \notin W$ and $W_1 = W \cup \{F\}$, then $\operatorname{len} W_1 = \operatorname{len} W + \operatorname{len} F$.

(12) $\operatorname{len}(W \cup \{F\}) \leq \operatorname{len} W + \operatorname{len} F$.

(13) $\operatorname{len}(\emptyset_H) = 0$.

(14) $\operatorname{len}(\{F\}) = \operatorname{len} F$.

(15) If $W \subseteq W_1$, then $\operatorname{len} W \leq \operatorname{len} W_1$.

(16) If $\operatorname{len} W < 1$, then $W = \emptyset_H$.

(17) $\operatorname{len} W \geq 0$.

(18) $\operatorname{len}(W_1 \cup W_2) \leq \operatorname{len} W_1 + \operatorname{len} W_2$.

Let us consider v, H. Let us assume that $H \in$ Subformulae v. The functor LTLNew$_1(H, v)$ yielding a subset of Subformulae v is defined by:

(Def. 27) LTLNew$_1(H, v) =$ LTLNew$_1$ H.

The functor LTLNew$_2(H, v)$ yields a subset of Subformulae v and is defined by:

(Def. 28) LTLNew$_2(H, v) =$ LTLNew$_2$ H.

The following propositions are true:

(19) If N_2 is a 1st successor of N_1, then len (the new-component of N_2) $\leq$ len (the new-component of N_1) $- 1$.

(20) If N_2 is a 2nd successor of N_1, then len (the new-component of N_2) $\leq$ len (the new-component of N_1) $- 1$.

Let us consider v, N. The functor len N yields a natural number and is defined by:

(Def. 29) len $N = \lfloor$len (the new-component of N)$\rfloor$.

The following propositions are true:

(21) If N_2 is a successor of N_1, then len $N_2 \leq$ len $N_1 - 1$.

(22) If len $N \leq 0$, then the new-component of $N = \emptyset_v$.

(23) If len $N > 0$, then the new-component of $N \neq \emptyset_v$.

(24) There exist n, L, M such that $1 \leq n$ and len $L = n$ and $L(1) = N$ and $L(n) = M$ and the new-component of $M = \emptyset_v$ and L is a successor sequence for v.

(25) Suppose N_2 is a successor of N_1. Then
  (i) the old-component of $N_1 \subseteq$ the old-component of N_2, and
  (ii) the next-component of $N_1 \subseteq$ the next-component of N_2.

(26) If L is a successor sequence for v and $m \leq$ len L and $L_1 = L \restriction$ Seg m, then L_1 is a successor sequence for v.

(27) Suppose that
  (i) L is a successor sequence for v,
  (ii) $F \notin$ the old-component of CastNode$(L(1), v)$,
  (iii) $1 < n$,
  (iv) $n \leq$ len L, and
  (v) $F \in$ the old-component of CastNode$(L(n), v)$.

  Then there exists m such that $1 \leq m < n$ and $F \notin$ the old-component of CastNode$(L(m), v)$ and $F \in$ the old-component of CastNode$(L(m+1), v)$.

(28) Suppose N_2 is a successor of N_1 and $F \notin$ the old-component of N_1 and $F \in$ the old-component of N_2. Then N_2 is a successor of N_1 and F.

(29) Suppose that
  (i) L is a successor sequence for v,
  (ii) $F \in$ the new-component of CastNode$(L(1), v)$,
  (iii) $1 < n$,

(iv) $n \leq \operatorname{len} L$, and

(v) $F \notin$ the new-component of $\operatorname{CastNode}(L(n), v)$.

Then there exists m such that $1 \leq m < n$ and $F \in$ the new-component of $\operatorname{CastNode}(L(m), v)$ and $F \notin$ the new-component of $\operatorname{CastNode}(L(m+1), v)$.

(30) Suppose N_2 is a successor of N_1 and $F \in$ the new-component of N_1 and $F \notin$ the new-component of N_2. Then N_2 is a successor of N_1 and F.

(31) Suppose L is a successor sequence for v and $1 \leq m \leq n \leq \operatorname{len} L$. Then

(i) the old-component of $\operatorname{CastNode}(L(m), v) \subseteq$ the old-component of $\operatorname{CastNode}(L(n), v)$, and

(ii) the next-component of $\operatorname{CastNode}(L(m), v) \subseteq$ the next-component of $\operatorname{CastNode}(L(n), v)$.

(32) If N_2 is a successor of N_1 and F, then $F \in$ the old-component of N_2.

(33) Suppose L is a successor sequence for v and $1 \leq \operatorname{len} L$ and the new-component of $\operatorname{CastNode}(L(\operatorname{len} L), v) = \emptyset_v$. Then the new-component of $\operatorname{CastNode}(L(1), v) \subseteq$ the old-component of $\operatorname{CastNode}(L(\operatorname{len} L), v)$.

(34) Suppose L is a successor sequence for v and $1 \leq m \leq \operatorname{len} L$ and the new-component of $\operatorname{CastNode}(L(\operatorname{len} L), v) = \emptyset_v$. Then the new-component of $\operatorname{CastNode}(L(m), v) \subseteq$ the old-component of $\operatorname{CastNode}(L(\operatorname{len} L), v)$.

(35) If L is a successor sequence for v and $1 \leq k < \operatorname{len} L$, then $\operatorname{CastNode}(L(k+1), v)$ is a successor of $\operatorname{CastNode}(L(k), v)$.

(36) If L is a successor sequence for v and $1 \leq k \leq \operatorname{len} L$, then $\operatorname{len} \operatorname{CastNode}(L(k), v) \leq (\operatorname{len} \operatorname{CastNode}(L(1), v) - k) + 1$.

In the sequel s, s_0, s_1, s_2 denote elementary strict LTL-nodes over v.

The following propositions are true:

(37) If s_2 is next to s_1, then the next-component of $s_1 \subseteq$ the old-component of s_2.

(38) Suppose s_2 is next to s_1 and $F \in$ the old-component of s_2. Then there exist L, m such that

$1 \leq \operatorname{len} L$ and L is a successor sequence for v and $L(1) = \mathcal{X}\, s_1$ and $L(\operatorname{len} L) = s_2$ and $1 \leq m < \operatorname{len} L$ and $\operatorname{CastNode}(L(m+1), v)$ is a successor of $\operatorname{CastNode}(L(m), v)$ and F.

(39) Suppose s_2 is next to s_1 and H has *release* operator and $H \in$ the old-component of s_2 and $\operatorname{LeftArg}(H) \notin$ the old-component of s_2. Then $\operatorname{RightArg}(H) \in$ the old-component of s_2 and $H \in$ the next-component of s_2.

(40) Suppose s_2 is next to s_1 and H has *release* operator and $H \in$ the next-component of s_1. Then $\operatorname{RightArg}(H) \in$ the old-component of s_2 and $H \in$ the old-component of s_2.

(41) Suppose s_1 is next to s_0 and $H \in$ the old-component of s_1. Then

(i) if H is conjunctive, then $\mathrm{LeftArg}(H) \in$ the old-component of s_1 and $\mathrm{RightArg}(H) \in$ the old-component of s_1,

(ii) if H is either disjunctive or has *until* operator, then $\mathrm{LeftArg}(H) \in$ the old-component of s_1 or $\mathrm{RightArg}(H) \in$ the old-component of s_1,

(iii) if H has *next* operator, then $\mathrm{Arg}(H) \in$ the next-component of s_1, and

(iv) if H has *release* operator, then $\mathrm{RightArg}(H) \in$ the old-component of s_1.

(42) Suppose s_1 is next to s_0 and s_2 is next to s_1 and $H \in$ the old-component of s_1 and H has *until* operator. Then $\mathrm{RightArg}(H) \in$ the old-component of s_1 or $\mathrm{LeftArg}(H) \in$ the old-component of s_1 and $H \in$ the old-component of s_2.

Let us consider v. The functor $\mathrm{Nodes}_{\mathrm{LTL}}\, v$ yields a non empty set and is defined as follows:

(Def. 30) $x \in \mathrm{Nodes}_{\mathrm{LTL}}\, v$ iff there exists a strict LTL-node N over v such that $x = N$.

Let us consider v. Note that $\mathrm{Nodes}_{\mathrm{LTL}}\, v$ is finite.

Let us consider v. The functor $\mathrm{States}_{\mathrm{LTL}}\, v$ yields a non empty set and is defined by:

(Def. 31) $\mathrm{States}_{\mathrm{LTL}}\, v = \{x \in \mathrm{Nodes}_{\mathrm{LTL}}\, v : x \text{ is an elementary strict LTL-node over } v\}$.

Let us consider v. Observe that $\mathrm{States}_{\mathrm{LTL}}\, v$ is finite.

The following propositions are true:

(43) $\mathrm{init}\, v$ is an element of $\mathrm{States}_{\mathrm{LTL}}\, v$.

(44) s is an element of $\mathrm{States}_{\mathrm{LTL}}\, v$.

(45) x is an element of $\mathrm{States}_{\mathrm{LTL}}\, v$ iff there exists s such that $s = x$.

Let us consider v, let us consider w, and let f be a function. We say that f is a successor homomorphism from v to w if and only if:

(Def. 32) For every x such that $x \in \mathrm{Nodes}_{\mathrm{LTL}}\, v$ and $\mathrm{CastNode}(x, v)$ is non elementary and $w \models \cdot\, \mathrm{CastNode}(x, v)$ holds $\mathrm{CastNode}(f(x), v)$ is a successor of $\mathrm{CastNode}(x, v)$ and $w \models \cdot\, \mathrm{CastNode}(f(x), v)$.

We say that f is a homomorphism of v into w if and only if:

(Def. 33) For every x such that $x \in \mathrm{Nodes}_{\mathrm{LTL}}\, v$ and $\mathrm{CastNode}(x, v)$ is non elementary and $w \models \cdot\, \mathrm{CastNode}(x, v)$ holds $w \models \cdot\, \mathrm{CastNode}(f(x), v)$.

The following propositions are true:

(46) Let f be a function from $\mathrm{Nodes}_{\mathrm{LTL}}\, v$ into $\mathrm{Nodes}_{\mathrm{LTL}}\, v$. Suppose f is a successor homomorphism from v to w. Then f is a homomorphism of v into w.

(47) Let f be a function from $\mathrm{Nodes}_{\mathrm{LTL}}\, v$ into $\mathrm{Nodes}_{\mathrm{LTL}}\, v$. Suppose f is a homomorphism of v into w. Let given x. Suppose $x \in \mathrm{Nodes}_{\mathrm{LTL}}\, v$ and

CastNode(x, v) is non elementary and $w \models \cdot\,\mathrm{CastNode}(x, v)$. Let given k. If for every i such that $i \leq k$ holds CastNode$(f^i(x), v)$ is non elementary, then $w \models \cdot\,\mathrm{CastNode}(f^k(x), v)$.

(48) Let f be a function from $\mathrm{Nodes}_{\mathrm{LTL}}\, v$ into $\mathrm{Nodes}_{\mathrm{LTL}}\, v$. Suppose f is a successor homomorphism from v to w. Let given x. Suppose $x \in \mathrm{Nodes}_{\mathrm{LTL}}\, v$ and CastNode(x, v) is non elementary and $w \models \cdot\,\mathrm{CastNode}(x, v)$. Let given k. Suppose that for every i such that $i \leq k$ holds CastNode$(f^i(x), v)$ is non elementary. Then CastNode$(f^{k+1}(x), v)$ is a successor of CastNode$(f^k(x), v)$ and $w \models \cdot\,\mathrm{CastNode}(f^k(x), v)$.

(49) Let f be a function from $\mathrm{Nodes}_{\mathrm{LTL}}\, v$ into $\mathrm{Nodes}_{\mathrm{LTL}}\, v$. Suppose f is a successor homomorphism from v to w. Let given x. Suppose $x \in \mathrm{Nodes}_{\mathrm{LTL}}\, v$ and CastNode(x, v) is non elementary and $w \models \cdot\,\mathrm{CastNode}(x, v)$. Then there exists n such that for every i such that $i < n$ holds CastNode$(f^i(x), v)$ is non elementary and CastNode$(f^n(x), v)$ is elementary.

(50) Let f be a function from $\mathrm{Nodes}_{\mathrm{LTL}}\, v$ into $\mathrm{Nodes}_{\mathrm{LTL}}\, v$. Suppose f is a homomorphism of v into w. Let given x. Suppose $x \in \mathrm{Nodes}_{\mathrm{LTL}}\, v$ and CastNode(x, v) is non elementary. Let given k. If CastNode$(f^k(x), v)$ is non elementary and $w \models \cdot\,\mathrm{CastNode}(f^k(x), v)$, then $w \models \cdot\,\mathrm{CastNode}(f^{k+1}(x), v)$.

(51) Let f be a function from $\mathrm{Nodes}_{\mathrm{LTL}}\, v$ into $\mathrm{Nodes}_{\mathrm{LTL}}\, v$. Suppose f is a successor homomorphism from v to w. Let given x. Suppose $x \in \mathrm{Nodes}_{\mathrm{LTL}}\, v$ and CastNode(x, v) is non elementary and $w \models \cdot\,\mathrm{CastNode}(x, v)$. Then there exists n such that

(i) for every i such that $i < n$ holds CastNode$(f^i(x), v)$ is non elementary and CastNode$(f^{i+1}(x), v)$ is a successor of CastNode$(f^i(x), v)$,

(ii) CastNode$(f^n(x), v)$ is elementary, and

(iii) for every i such that $i \leq n$ holds $w \models \cdot\,\mathrm{CastNode}(f^i(x), v)$.

In the sequel q denotes a sequence of $\mathrm{States}_{\mathrm{LTL}}\, v$.

One can prove the following propositions:

(52) There exists s such that $s = \mathrm{CastNode}(q(n), v)$.

(53) Suppose H has *until* operator and $H \in$ the old-component of CastNode$(q(1), v)$ and for every i holds CastNode$(q(i + 1), v)$ is next to CastNode$(q(i), v)$. Suppose that for every i such that $1 \leq i < n$ holds RightArg$(H) \notin$ the old-component of CastNode$(q(i), v)$. Let given i. Suppose $1 \leq i < n$. Then LeftArg$(H) \in$ the old-component of CastNode$(q(i), v)$ and $H \in$ the old-component of CastNode$(q(i), v)$.

(54) Suppose H has *until* operator and $H \in$ the old-component of CastNode$(q(1), v)$ and for every i holds CastNode$(q(i + 1), v)$ is next to CastNode$(q(i), v)$. Then

(i) for every i such that $i \geq 1$ holds $H \in$ the old-component of CastNode$(q(i), v)$ and LeftArg$(H) \in$ the old-component of CastNode$(q(i), v)$ and RightArg$(H) \notin$ the old-component of CastNode$(q(i), v)$, or

(ii) there exists j such that $j \geq 1$ and RightArg$(H) \in$ the old-component of CastNode$(q(j), v)$ and for every i such that $1 \leq i < j$ holds $H \in$ the old-component of CastNode$(q(i), v)$ and LeftArg$(H) \in$ the old-component of CastNode$(q(i), v)$.

(55) $\bigcup(2_+^X) = X$.

(56) If N is non elementary, then the new-component of $N \neq \emptyset$ and the new-component of $N \in 2_+^{\text{Subformulae } v}$.

Let us consider v. One can verify that $\bigcup(2_+^{\text{Subformulae } v})$ is non empty and $2_+^{\text{Subformulae } v}$ is non empty.

We now state the proposition

(57) There exists a choice function of $2_+^{\text{Subformulae } v}$ which is a function from $2_+^{\text{Subformulae } v}$ into Subformulae v.

In the sequel U denotes a choice function of $2_+^{\text{Subformulae } v}$.

Let us consider v, let us consider U, and let us consider N. Let us assume that N is non elementary. The U-chosen formula of N yielding an LTL-formula is defined as follows:

(Def. 34) The U-chosen formula of $N = U$(the new-component of N).

The following proposition is true

(58) If N is non elementary, then the U-chosen formula of $N \in$ the new-component of N.

Let us consider w, let us consider v, let us consider U, and let us consider N. The U-chosen successor of N w.r.t. w, v yields a strict LTL-node over v and is defined by:

(Def. 35) The U-chosen successor of N w.r.t. w, v

$$= \begin{cases} \text{SuccNode}_1(\text{the } U\text{-chosen formula of } N, N), \\ \quad \text{if the } U\text{-chosen formula of } N \text{does not have } \textit{until} \text{ operator and} \\ \quad w \models \cdot \text{SuccNode}_1(\text{the } U\text{-chosen formula of } N, N) \text{ or} \\ \quad \text{the } U\text{-chosen formula of } N \text{ has } \textit{until} \text{ operator and} \\ \quad w \not\models \text{RightArg}(\text{the } U\text{-chosen formula of } N), \\ \text{SuccNode}_2(\text{the } U\text{-chosen formula of } N,\ N), \text{otherwise.} \end{cases}$$

One can prove the following propositions:

(59) Suppose $w \models \cdot N$ and N is non elementary. Then

(i) $w \models \cdot$(the U-chosen successor of N w.r.t. w, v), and

(ii) the U-chosen successor of N w.r.t. w, v is a successor of N.

(60) Suppose $w \models \cdot N$ and N is non elementary. Suppose the U-chosen formula of N has *until* operator and $w \models$ RightArg(the U-chosen formula of N).

Then

(i) RightArg(the U-chosen formula of N) $\in$ the new-component of the U-chosen successor of N w.r.t. w, v or RightArg(the U-chosen formula of N) $\in$ the old-component of N, and

(ii) the U-chosen formula of $N \in$ the old-component of the U-chosen successor of N w.r.t. w, v.

(61) Suppose $w \models \cdot N$ and N is non elementary. Then

(i) the old-component of $N \subseteq$ the old-component of the U-chosen successor of N w.r.t. w, v, and

(ii) the next-component of $N \subseteq$ the next-component of the U-chosen successor of N w.r.t. w, v.

Let us consider w, let us consider v, and let us consider U. The U-choice successor function w.r.t. w, v yielding a function from $\mathrm{Nodes}_{\mathrm{LTL}}\, v$ into $\mathrm{Nodes}_{\mathrm{LTL}}\, v$ is defined by the condition (Def. 36).

(Def. 36) Let given x. Suppose $x \in \mathrm{Nodes}_{\mathrm{LTL}}\, v$. Then (the U-choice successor function w.r.t. w, v)(x) = the U-chosen successor of $\mathrm{CastNode}(x, v)$ w.r.t. w, v.

We now state the proposition

(62) The U-choice successor function w.r.t. w, v is a successor homomorphism from v to w.

2. Negation Inner most LTL

Let us consider H. We say that H is negation-inner-most if and only if:

(Def. 37) For every LTL-formula G such that G is a subformula of H holds if G is negative, then $\mathrm{Arg}(G)$ is atomic.

Let us observe that there exists an LTL-formula which is negation-inner-most.

Let us consider H. We say that H is sub-atomic if and only if:

(Def. 38) H is atomic or there exists an LTL-formula G such that G is atomic and $H = \neg G$.

Next we state several propositions:

(63) If H is negation-inner-most and F is a subformula of H, then F is negation-inner-most.

(64) H is sub-atomic iff H is atomic or H is negative and $\mathrm{Arg}(H)$ is atomic.

(65) Suppose H is negation-inner-most. Then H is either sub-atomic, or conjunctive, or disjunctive, or has *next* operator, or *until* operator, or *release* operator.

(66) If H is negation-inner-most and has *next* operator, then $\mathrm{Arg}(H)$ is negation-inner-most.

(67) Suppose that
  (i) H is conjunctive, or
  (ii) H is disjunctive, or
  (iii) H is negation-inner-most.

Then LeftArg(H) is negation-inner-most and RightArg(H) is negation-inner-most.

3. Definition of Buchi Automaton and Verification of the Main Theorem

Let W be a non empty set. We consider Buchi automatons over W as systems
$\langle$ a carrier, a transition, an initial state, final states $\rangle$,
where the carrier is a set, the transition is a relation between the carrier$\times W$ and the carrier, the initial state is an element of $2^{\text{the carrier}}$, and the final states constitute a subset of $2^{\text{the carrier}}$.

Let W be a non empty set, let B be a Buchi automaton over W, and let w be an element of the infinite sequences of W. We say that w is accepted by B if and only if the condition (Def. 39) is satisfied.

(Def. 39) There exists a sequence r_1 of the carrier of B such that
  (i) $r_1(0) \in$ the initial state of B, and
  (ii) for every natural number i holds $\langle\langle r_1(i), (\mathrm{CastSeq}(w, W))(i)\rangle, r_1(i+1)\rangle \in$ the transition of B and for every set F_1 such that $F_1 \in$ the final states of B holds $\{k \in \mathbb{N}: r_1(k) \in F_1\}$ is an infinite set.

For simplicity, we use the following convention: v denotes a negation-inner-most LTL-formula, U denotes a choice function of $2_+^{\text{Subformulae}\, v}$, N denotes a strict LTL-node over v, and s, s_1 denote elementary strict LTL-nodes over v.

Let us consider v and let us consider N. The functor $\mathrm{atomic}_{\mathrm{LTL}}\, N$ yields a subset of $\mathrm{WFF}_{\mathrm{LTL}}$ and is defined by:

(Def. 40) $\mathrm{atomic}_{\mathrm{LTL}}\, N = \{x; x$ ranges over LTL-formulae: x is atomic $\wedge\ x \in$ the old-component of $N\}$.

The functor $\mathrm{NegAtomic}_{\mathrm{LTL}}\, N$ yields a subset of $\mathrm{WFF}_{\mathrm{LTL}}$ and is defined as follows:

(Def. 41) $\mathrm{NegAtomic}_{\mathrm{LTL}}\, N = \{x; x$ ranges over LTL-formulae: x is atomic $\wedge\ \neg x \in$ the old-component of $N\}$.

Let us consider v and let us consider N. The functor $\mathrm{Label}\, N$ yielding a set is defined by:

(Def. 42) $\mathrm{Label}\, N = \{x \subseteq \mathrm{atomic}_{\mathrm{LTL}}$: $\mathrm{atomic}_{\mathrm{LTL}}\, N \subseteq x\ \wedge\ \mathrm{NegAtomic}_{\mathrm{LTL}}\, N$ misses $x\}$.

Let us consider v. The functor $\mathrm{Tran}_{\mathrm{LTL}}\, v$ yields a relation between $\mathrm{States}_{\mathrm{LTL}}\, v \times$ AtomicFamily and $\mathrm{States}_{\mathrm{LTL}}\, v$ and is defined as follows:

(Def. 43) $\mathrm{Tran}_{\mathrm{LTL}}\, v = \{y \in \mathrm{States}_{\mathrm{LTL}}\, v \times \mathrm{AtomicFamily} \times \mathrm{States}_{\mathrm{LTL}}\, v : \bigvee_{s,s_1,x} (y = \langle\langle s,\, x\rangle,\, s_1\rangle \wedge s_1 \text{ is next to } s \wedge x \in \mathrm{Label}\, s_1)\}$.

The functor $\mathrm{InitS}_{\mathrm{LTL}}\, v$ yielding an element of $2^{\mathrm{States}_{\mathrm{LTL}}\, v}$ is defined as follows:

(Def. 44) $\mathrm{InitS}_{\mathrm{LTL}}\, v = \{\mathrm{init}\, v\}$.

Let us consider v and let us consider F. The functor $\mathrm{FinalS}_{\mathrm{LTL}}(F, v)$ yields an element of $2^{\mathrm{States}_{\mathrm{LTL}}\, v}$ and is defined as follows:

(Def. 45) $\mathrm{FinalS}_{\mathrm{LTL}}(F, v) = \{x \in \mathrm{States}_{\mathrm{LTL}}\, v : F \notin$ the old-component of $\mathrm{CastNode}(x, v) \vee \mathrm{RightArg}(F) \in$ the old-component of $\mathrm{CastNode}(x, v)\}$.

Let us consider v. The functor $\mathrm{FinalS}_{\mathrm{LTL}}\, v$ yields a subset of $2^{\mathrm{States}_{\mathrm{LTL}}\, v}$ and is defined by:

(Def. 46) $\mathrm{FinalS}_{\mathrm{LTL}}\, v = \{x \in 2^{\mathrm{States}_{\mathrm{LTL}}\, v} : \bigvee_F (F$ is a subformula of $v \wedge F$ has *until* operator $\wedge\, x = \mathrm{FinalS}_{\mathrm{LTL}}(F, v))\}$.

Let us consider v. The functor $\mathrm{BAutomaton}\, v$ yields a Buchi automaton over AtomicFamily and is defined as follows:

(Def. 47) $\mathrm{BAutomaton}\, v = \langle \mathrm{States}_{\mathrm{LTL}}\, v, \mathrm{Tran}_{\mathrm{LTL}}\, v, \mathrm{InitS}_{\mathrm{LTL}}\, v, \mathrm{FinalS}_{\mathrm{LTL}}\, v\rangle$.

The following proposition is true

(68) If w is accepted by $\mathrm{BAutomaton}\, v$, then $w \models v$.

Let us consider w, let us consider v, let us consider U, and let us consider N. Let us assume that N is non elementary and $w \models \cdot N$. The U-chosen successor end number of N w.r.t. w, v yields an element of $\mathbb{N}$ and is defined by the conditions (Def. 48).

(Def. 48)(i) For every i such that $i <$ the U-chosen successor end number of N w.r.t. w, v holds $\mathrm{CastNode}((\text{the } U\text{-choice successor function w.r.t. } w,\, v)^i(N), v)$ is non elementary and $\mathrm{CastNode}((\text{the } U\text{-choice successor function w.r.t. } w,\, v)^{i+1}(N), v)$ is a successor of $\mathrm{CastNode}((\text{the } U\text{-choice successor function w.r.t. } w,\, v)^i(N), v)$,

(ii) $\mathrm{CastNode}((\text{the } U\text{-choice successor function w.r.t. } w,\, v)^{\text{the } U\text{-chosen successor end number of } N \text{ w.r.t. } w,\ v}(N), v)$ is elementary, and

(iii) for every i such that $i \leq$ the U-chosen successor end number of N w.r.t. w, v holds $w \models \cdot \mathrm{CastNode}((\text{the } U\text{-choice successor function w.r.t. } w,\, v)^i(N), v)$.

Let us consider w, let us consider v, let us consider U, and let us consider N. Let us assume that $w \models \cdot \mathcal{X}\, N$. The U-chosen next node to N w.r.t. w, v yielding an elementary strict LTL-node over v is defined by:

(Def. 49) The U-chosen next node to N w.r.t. w, v

$$= \begin{cases} \mathrm{CastNode}((\text{the } U\text{-choice successor function w.r.t. } w, \\ \quad v)^{\text{the } U\text{-chosen successor end number of } \mathcal{X}\, N \text{ w.r.t. } w,\ v}(\mathcal{X}\, N), v), \\ \quad \text{if } \mathcal{X}\, N \text{ is non elementary}, \\ \mathrm{FinalNode}\, v, \text{ otherwise}. \end{cases}$$

One can prove the following proposition

(69) Suppose $w \models \cdot \mathcal{X}\, s$. Then the U-chosen next node to s w.r.t. w, v is next to s and $w \models \cdot$(the U-chosen next node to s w.r.t. w, v).

Let us consider w, let us consider v, and let us consider U. The U-chosen run w.r.t. w, v yields a sequence of $\mathrm{States}_{\mathrm{LTL}}\, v$ and is defined by the conditions (Def. 50).

(Def. 50)(i) (The U-chosen run w.r.t. w, v)$(0) = \mathrm{init}\, v$, and

(ii) for every n holds (the U-chosen run w.r.t. w, v)$(n+1)$ = the U-chosen next node to CastNode((the U-chosen run w.r.t. w, v)$(n), v)$ w.r.t. Shift(w, n), v.

The following propositions are true:

(70) If $w \models \cdot N$, then Shift$(w, 1) \models \cdot \mathcal{X}\, N$.

(71) If $w \models \mathcal{X}\, v$, then $w \models \cdot \mathrm{init}\, v$.

(72) $w \models v$ iff $w \models \cdot \mathcal{X}\, \mathrm{init}\, v$.

(73) Suppose $w \models v$. Let given n. Then

(i) CastNode((the U-chosen run w.r.t. w, v)$(n+1), v)$ is next to CastNode((the U-chosen run w.r.t. w, v)$(n), v)$, and

(ii) Shift$(w, n) \models \cdot \mathcal{X}$ CastNode((the U-chosen run w.r.t. w, v)$(n), v)$.

(74) Suppose $w \models v$. Let given i. Suppose $H \in$ the old-component of CastNode((the U-chosen run w.r.t. w, v)$(i+1), v)$ and H has *until* operator and Shift$(w, i) \models$ RightArg(H). Then RightArg$(H) \in$ the old-component of CastNode((the U-chosen run w.r.t. w, v)$(i+1), v)$.

(75) w is accepted by BAutomaton v iff $w \models v$.

References

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

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

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

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

[5] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.

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

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

[8] Kazuhisa Ishida. Model checking. Part I. *Formalized Mathematics*, 14(**4**):171–186, 2006.

[9] Kazuhisa Ishida. Model checking. Part II. *Formalized Mathematics*, 16(**3**):231–245, 2008.

[10] Jarosław Kotowicz. Real sequences and basic operations on them. *Formalized Mathematics*, 1(**2**):269–272, 1990.

[11] Konrad Raczkowski and Andrzej Nędzusiak. Series. *Formalized Mathematics*, 2(**4**):449–452, 1991.

[12] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.

[13] Wojciech A. Trybulec. Partially ordered sets. *Formalized Mathematics*, 1(**2**):313–319, 1990.

[14] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.

[15] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[16] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

Received August 19, 2008