FORMALIZED MATHEMATICS
Volume 14, Number 4, Pages 171–186
University of Białystok, 2006

Model Checking. Part I

Kazuhisa Ishida
Shinshu University
Nagano, Japan

Summary. This text includes definitions of the Kripke structure, CTL (Computation Tree Logic), and verification of the basic algorithm for Model Checking based on CTL in [10].

MML identifier: MODELC_1, version: 7.8.03 4.75.958

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

Let x, S be sets and let a be an element of S. The functor $\mathrm{k.id}(x, S, a)$ yields an element of S and is defined by:

(Def. 1) $\quad \mathrm{k.id}(x, S, a) = \begin{cases} x, & \text{if } x \in S, \\ a, & \text{otherwise.} \end{cases}$

Let x be a set. The functor $\mathrm{k.nat}\, x$ yields an element of $\mathbb{N}$ and is defined by:

(Def. 2) $\quad \mathrm{k.nat}\, x = \begin{cases} x, & \text{if } x \in \mathbb{N}, \\ 0, & \text{otherwise.} \end{cases}$

Let f be a function and let x, a be sets. The functor $\mathrm{UnivF}(x, f, a)$ yielding a set is defined by:

(Def. 3) $\quad \mathrm{UnivF}(x, f, a) = \begin{cases} f(x), & \text{if } x \in \mathrm{dom}\, f, \\ a, & \text{otherwise.} \end{cases}$

Let a be a set. The functor $\mathrm{Castboolean}\, a$ yields a boolean set and is defined by:

(Def. 4) $\quad \mathrm{Castboolean}\, a = \begin{cases} a, & \text{if } a \text{ is a boolean set}, \\ \mathit{false}, & \text{otherwise.} \end{cases}$

Let X, a be sets. The functor $\mathrm{CastBool}(a, X)$ yielding a subset of X is defined as follows:

(Def. 5) $\quad \mathrm{CastBool}(a, X) = \begin{cases} a, & \text{if } a \subseteq X, \\ \emptyset, & \text{otherwise.} \end{cases}$

ISSN 1426–2630

For simplicity, we adopt the following rules: n denotes an element of $\mathbb{N}$, a denotes a set, D denotes a non empty set, and p, q denote finite sequences of elements of $\mathbb{N}$.

Let x be a variable. Then $\langle x \rangle$ is a finite sequence of elements of $\mathbb{N}$.

Let us consider n. The functor $\mathrm{atom}.\,n$ yields a finite sequence of elements of $\mathbb{N}$ and is defined by:

(Def. 6) $\mathrm{atom}.\,n = \langle 5 + n \rangle$.

Let us consider p. The functor $\neg p$ yielding a finite sequence of elements of $\mathbb{N}$ is defined by:

(Def. 7) $\neg p = \langle 0 \rangle \frown p$.

Let us consider q. The functor $p \wedge q$ yielding a finite sequence of elements of $\mathbb{N}$ is defined by:

(Def. 8) $p \wedge q = \langle 1 \rangle \frown p \frown q$.

Let us consider p. The functor $\mathrm{EX}\,p$ yielding a finite sequence of elements of $\mathbb{N}$ is defined as follows:

(Def. 9) $\mathrm{EX}\,p = \langle 2 \rangle \frown p$.

The functor $\mathrm{EG}\,p$ yielding a finite sequence of elements of $\mathbb{N}$ is defined by:

(Def. 10) $\mathrm{EG}\,p = \langle 3 \rangle \frown p$.

Let us consider q. The functor $p\,\mathrm{EU}\,q$ yields a finite sequence of elements of $\mathbb{N}$ and is defined as follows:

(Def. 11) $p\,\mathrm{EU}\,q = \langle 4 \rangle \frown p \frown q$.

The non empty set CTL-WFF is defined by the conditions (Def. 12).

(Def. 12) For every a such that $a \in$ CTL-WFF holds a is a finite sequence of elements of $\mathbb{N}$ and for every n holds $\mathrm{atom}.\,n \in$ CTL-WFF and for every p such that $p \in$ CTL-WFF holds $\neg p \in$ CTL-WFF and for all p, q such that $p \in$ CTL-WFF and $q \in$ CTL-WFF holds $p \wedge q \in$ CTL-WFF and for every p such that $p \in$ CTL-WFF holds $\mathrm{EX}\,p \in$ CTL-WFF and for every p such that $p \in$ CTL-WFF holds $\mathrm{EG}\,p \in$ CTL-WFF and for all p, q such that $p \in$ CTL-WFF and $q \in$ CTL-WFF holds $p\,\mathrm{EU}\,q \in$ CTL-WFF and for every D such that for every a such that $a \in D$ holds a is a finite sequence of elements of $\mathbb{N}$ and for every n holds $\mathrm{atom}.\,n \in D$ and for every p such that $p \in D$ holds $\neg p \in D$ and for all p, q such that $p \in D$ and $q \in D$ holds $p \wedge q \in D$ and for every p such that $p \in D$ holds $\mathrm{EX}\,p \in D$ and for every p such that $p \in D$ holds $\mathrm{EG}\,p \in D$ and for all p, q such that $p \in D$ and $q \in D$ holds $p\,\mathrm{EU}\,q \in D$ holds CTL-WFF $\subseteq D$.

Let I_1 be a finite sequence of elements of $\mathbb{N}$. We say that I_1 is CTL-formula-like if and only if:

(Def. 13) I_1 is an element of CTL-WFF.

Let us mention that there exists a finite sequence of elements of $\mathbb{N}$ which is CTL-formula-like.

A CTL-formula is a CTL-formula-like finite sequence of elements of $\mathbb{N}$.

One can prove the following proposition

(1) a is a CTL-formula iff $a \in$ CTL-WFF .

In the sequel F, G, H, H_1, H_2 denote CTL-formulae.

Let us consider n. One can verify that atom. n is CTL-formula-like.

Let us consider H. One can verify the following observations:

* $\neg H$ is CTL-formula-like,
* $\mathrm{EX}\, H$ is CTL-formula-like, and
* $\mathrm{EG}\, H$ is CTL-formula-like.

Let us consider G. One can verify that $H \wedge G$ is CTL-formula-like and $H \,\mathrm{EU}\, G$ is CTL-formula-like.

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

(Def. 14) There exists n such that $H = \mathrm{atom}.\, n$.

We say that H is negative if and only if:

(Def. 15) There exists H_1 such that $H = \neg H_1$.

We say that H is conjunctive if and only if:

(Def. 16) There exist F, G such that $H = F \wedge G$.

We say that H is exist-next-formula if and only if:

(Def. 17) There exists H_1 such that $H = \mathrm{EX}\, H_1$.

We say that H is exist-global-formula if and only if:

(Def. 18) There exists H_1 such that $H = \mathrm{EG}\, H_1$.

We say that H is exist-until-formula if and only if:

(Def. 19) There exist F, G such that $H = F \,\mathrm{EU}\, G$.

Let us consider F, G. The functor $F \vee G$ yielding a CTL-formula is defined by:

(Def. 20) $F \vee G = \neg(\neg F \wedge \neg G)$.

One can prove the following proposition

(2) H is atomic, or negative, or conjunctive, or exist-next-formula, or exist-global-formula, or exist-until-formula.

Let us consider H. Let us assume that H is negative, or exist-next-formula, or exist-global-formula. The functor $\mathrm{Arg}(H)$ yielding a CTL-formula is defined as follows:

(Def. 21)(i) $\neg\, \mathrm{Arg}(H) = H$ if H is negative,

(ii) $\mathrm{EX}\, \mathrm{Arg}(H) = H$ if H is exist-next-formula,

(iii) $\mathrm{EG}\, \mathrm{Arg}(H) = H$, otherwise.

Let us consider H. Let us assume that H is conjunctive or exist-until-formula. The functor $\mathrm{LeftArg}(H)$ yields a CTL-formula and is defined as follows:

(Def. 22)(i) There exists H_1 such that $\mathrm{LeftArg}(H) \wedge H_1 = H$ if H is conjunctive,

(ii) there exists H_1 such that $\mathrm{LeftArg}(H)\,\mathrm{EU}\,H_1 = H$, otherwise.

The functor $\mathrm{RightArg}(H)$ yielding a CTL-formula is defined by:

(Def. 23)(i) There exists H_1 such that $H_1 \wedge \mathrm{RightArg}(H) = H$ if H is conjunctive,

(ii) there exists H_1 such that $H_1\,\mathrm{EU}\,\mathrm{RightArg}(H) = H$, otherwise.

Let x be a set. The functor $\mathrm{CastCTLformula}\,x$ yields a CTL-formula and is defined by:

(Def. 24) $\mathrm{CastCTLformula}\,x = \begin{cases} x, & \text{if } x \in \text{CTL-WFF}, \\ \mathrm{atom}.\,0, & \text{otherwise.} \end{cases}$

Let P_1 be a set. We consider Kripke structures over P_1 as systems

⟨ worlds, starts, possibilities, a label ⟩,

where the worlds constitute a set, the starts constitute a subset of the worlds, the possibilities constitute a total relation between the worlds and the worlds, and the label is a function from the worlds into 2^{P_1}.

We introduce CTL model structures which are systems

⟨ assignations, basic assignations, a conjunction, a negation, a next-operation, a global-operation, an until-operation ⟩,

where the assignations constitute a non empty set, the basic assignations constitute a non empty subset of the assignations, the conjunction is a binary operation on the assignations, the negation is a unary operation on the assignations, the next-operation is a unary operation on the assignations, the global-operation is a unary operation on the assignations, and the until-operation is a binary operation on the assignations.

Let V be a CTL model structure. An assignation of V is an element of the assignations of V.

The subset the atomic WFF of CTL-WFF is defined by:

(Def. 25) The atomic WFF $= \{x; x$ ranges over CTL-formulae: x is atomic$\}$.

Let V be a CTL model structure, let K_1 be a function from the atomic WFF into the basic assignations of V, and let f be a function from CTL-WFF into the assignations of V. We say that f is an evaluation for K_1 if and only if the condition (Def. 26) is satisfied.

(Def. 26) Let H be a CTL-formula. Then

(i) if H is atomic, then $f(H) = K_1(H)$,

(ii) if H is negative, then $f(H) = (\text{the negation of } V)(f(\mathrm{Arg}(H)))$,

(iii) if H is conjunctive, then $f(H) = (\text{the conjunction of } V)(f(\mathrm{LeftArg}(H)), f(\mathrm{RightArg}(H)))$,

(iv) if H is exist-next-formula, then $f(H) = (\text{the next-operation of } V)(f(\mathrm{Arg}(H)))$,

(v) if H is exist-global-formula, then $f(H) =$ (the global-operation of V)$(f(\mathrm{Arg}(H)))$, and

(vi) if H is exist-until-formula, then $f(H) =$ (the until-operation of V)$(f(\mathrm{LeftArg}(H)), f(\mathrm{RightArg}(H)))$.

Let V be a CTL model structure, let K_1 be a function from the atomic WFF into the basic assignations of V, let f be a function from CTL-WFF into the assignations of V, and let n be an element of $\mathbb{N}$. We say that f is a n-pre-evaluation for K_1 if and only if the condition (Def. 27) is satisfied.

(Def. 27) Let H be a CTL-formula such that $\mathrm{len}\, H \leq n$. Then

(i) if H is atomic, then $f(H) = K_1(H)$,

(ii) if H is negative, then $f(H) =$ (the negation of V)$(f(\mathrm{Arg}(H)))$,

(iii) if H is conjunctive, then $f(H) =$ (the conjunction of V)$(f(\mathrm{LeftArg}(H)), f(\mathrm{RightArg}(H)))$,

(iv) if H is exist-next-formula, then $f(H) =$ (the next-operation of V)$(f(\mathrm{Arg}(H)))$,

(v) if H is exist-global-formula, then $f(H) =$ (the global-operation of V)$(f(\mathrm{Arg}(H)))$, and

(vi) if H is exist-until-formula, then $f(H) =$ (the until-operation of V)$(f(\mathrm{LeftArg}(H)), f(\mathrm{RightArg}(H)))$.

Let V be a CTL model structure, let K_1 be a function from the atomic WFF into the basic assignations of V, let f, h be functions from CTL-WFF into the assignations of V, let n be an element of $\mathbb{N}$, and let H be a CTL-formula. The functor $\mathrm{GraftEval}(V, K_1, f, h, n, H)$ yields a set and is defined as follows:

(Def. 28) $\mathrm{GraftEval}(V, K_1, f, h, n, H) =$

$$\begin{cases} f(H), \text{ if } \mathrm{len}\, H > n+1, \\ K_1(H), \text{ if } \mathrm{len}\, H = n+1 \text{ and } H \text{ is atomic}, \\ (\text{the negation of } V)(h(\mathrm{Arg}(H))), \text{ if } \mathrm{len}\, H = n+1 \text{ and } H \text{ is negative}, \\ (\text{the conjunction of } V)(h(\mathrm{LeftArg}(H)), h(\mathrm{RightArg}(H))), \\ \quad \text{if } \mathrm{len}\, H = n+1 \text{ and } H \text{ is conjunctive}, \\ (\text{the next-operation of } V)(h(\mathrm{Arg}(H))), \text{ if } \mathrm{len}\, H = n+1 \text{ and } H \text{ is} \\ \quad \text{exist-next-formula}, \\ (\text{the global-operation of } V)(h(\mathrm{Arg}(H))), \text{ if } \mathrm{len}\, H = n+1 \text{ and } H \text{ is} \\ \quad \text{exist-global-formula}, \\ (\text{the until-operation of } V)(h(\mathrm{LeftArg}(H)), h(\mathrm{RightArg}(H))), \\ \quad \text{if } \mathrm{len}\, H = n+1 \text{ and } H \text{ is exist-until-formula}, \\ h(H), \text{ if } \mathrm{len}\, H < n+1, \\ \emptyset, \text{ otherwise.} \end{cases}$$

We follow the rules: V is a CTL model structure, K_1 is a function from the atomic WFF into the basic assignations of V, and f, f_1, f_2 are functions from CTL-WFF into the assignations of V.

Let V be a CTL model structure, let K_1 be a function from the atomic

WFF into the basic assignations of V, and let n be an element of $\mathbb{N}$. The functor $\mathrm{EvalSet}(V, K_1, n)$ yields a non empty set and is defined by:

(Def. 29) $\mathrm{EvalSet}(V, K_1, n) = \{h; h$ ranges over functions from CTL-WFF into the assignations of V: h is a n-pre-evaluation for $K_1\}$.

Let V be a CTL model structure, let v_0 be an element of the assignations of V, and let x be a set. The functor $\mathrm{CastEval}(V, x, v_0)$ yielding a function from CTL-WFF into the assignations of V is defined by:

(Def. 30) $\mathrm{CastEval}(V, x, v_0) = \begin{cases} x, & \text{if } x \in (\text{the assignations of } V)^{\text{CTL-WFF}}, \\ \text{CTL-WFF} \longmapsto v_0, & \text{otherwise.} \end{cases}$

Let V be a CTL model structure and let K_1 be a function from the atomic WFF into the basic assignations of V. The functor $\mathrm{EvalFamily}(V, K_1)$ yielding a non empty set is defined by the condition (Def. 31).

(Def. 31) Let p be a set. Then $p \in \mathrm{EvalFamily}(V, K_1)$ if and only if the following conditions are satisfied:

(i) $p \in 2^{(\text{the assignations of } V)^{\text{CTL-WFF}}}$, and

(ii) there exists an element n of $\mathbb{N}$ such that $p = \mathrm{EvalSet}(V, K_1, n)$.

We now state two propositions:

(3) There exists f which is an evaluation for K_1.

(4) If f_1 is an evaluation for K_1 and f_2 is an evaluation for K_1, then $f_1 = f_2$.

Let V be a CTL model structure, let K_1 be a function from the atomic WFF into the basic assignations of V, and let H be a CTL-formula. The functor $\mathrm{Evaluate}(H, K_1)$ yields an assignation of V and is defined by:

(Def. 32) There exists a function f from CTL-WFF into the assignations of V such that f is an evaluation for K_1 and $\mathrm{Evaluate}(H, K_1) = f(H)$.

Let V be a CTL model structure and let f be an assignation of V. The functor $\neg f$ yields an assignation of V and is defined as follows:

(Def. 33) $\neg f = (\text{the negation of } V)(f)$.

Let V be a CTL model structure and let f, g be assignations of V. The functor $f \wedge g$ yielding an assignation of V is defined by:

(Def. 34) $f \wedge g = (\text{the conjunction of } V)(f, g)$.

Let V be a CTL model structure and let f be an assignation of V. The functor $\mathrm{EX}\, f$ yields an assignation of V and is defined by:

(Def. 35) $\mathrm{EX}\, f = (\text{the next-operation of } V)(f)$.

The functor $\mathrm{EG}\, f$ yielding an assignation of V is defined as follows:

(Def. 36) $\mathrm{EG}\, f = (\text{the global-operation of } V)(f)$.

Let V be a CTL model structure and let f, g be assignations of V. The functor $f\, \mathrm{EU}\, g$ yields an assignation of V and is defined as follows:

(Def. 37) $f\, \mathrm{EU}\, g = (\text{the until-operation of } V)(f, g)$.

The functor $f \vee g$ yielding an assignation of V is defined as follows:

(Def. 38) $f \vee g = \neg(\neg f \wedge \neg g)$.

Next we state several propositions:

(5) $\mathrm{Evaluate}(\neg H, K_1) = \neg\,\mathrm{Evaluate}(H, K_1)$.

(6) $\mathrm{Evaluate}(H_1 \wedge H_2, K_1) = \mathrm{Evaluate}(H_1, K_1) \wedge \mathrm{Evaluate}(H_2, K_1)$.

(7) $\mathrm{Evaluate}(\mathrm{EX}\, H, K_1) = \mathrm{EX}\, \mathrm{Evaluate}(H, K_1)$.

(8) $\mathrm{Evaluate}(\mathrm{EG}\, H, K_1) = \mathrm{EG}\, \mathrm{Evaluate}(H, K_1)$.

(9) $\mathrm{Evaluate}(H_1\, \mathrm{EU}\, H_2, K_1) = \mathrm{Evaluate}(H_1, K_1)\, \mathrm{EU}\, \mathrm{Evaluate}(H_2, K_1)$.

(10) $\mathrm{Evaluate}(H_1 \vee H_2, K_1) = \mathrm{Evaluate}(H_1, K_1) \vee \mathrm{Evaluate}(H_2, K_1)$.

Let f be a function and let n be an element of $\mathbb{N}$. We introduce f^n as a synonym of f^n.

Let S be a set, let f be a function from S into S, and let n be an element of $\mathbb{N}$. Then f^n is a function from S into S.

We use the following convention: S is a non empty set, R is a total relation between S and S, and s, s_0, s_1 are elements of S.

The scheme *ExistPath* deals with a non empty set $\mathcal{A}$, a total relation $\mathcal{B}$ between $\mathcal{A}$ and $\mathcal{A}$, an element $\mathcal{C}$ of $\mathcal{A}$, and a unary functor $\mathcal{F}$ yielding a set, and states that:

> There exists a function f from $\mathbb{N}$ into $\mathcal{A}$ such that $f(0) = \mathcal{C}$ and for every element n of $\mathbb{N}$ holds $\langle f(n), f(n+1)\rangle \in \mathcal{B}$ and $f(n+1) \in \mathcal{F}(f(n))$

provided the following requirement is met:

- For every element s of $\mathcal{A}$ holds $\mathcal{B}^\circ\{s\} \cap \mathcal{F}(s)$ is a non empty subset of $\mathcal{A}$.

Let S be a non empty set and let R be a total relation between S and S. A function from $\mathbb{N}$ into S is said to be an infinity path of R if:

(Def. 39) For every element n of $\mathbb{N}$ holds $\langle \mathrm{it}(n), \mathrm{it}(n+1)\rangle \in R$.

Let S be a non empty set. The functor $\mathrm{ModelSP}\, S$ yields a non empty set and is defined by:

(Def. 40) $\mathrm{ModelSP}\, S = \mathit{Boolean}^S$.

Let S be a non empty set. Observe that $\mathrm{ModelSP}\, S$ is non empty.

Let S be a non empty set and let f be a set. The functor $\mathrm{Fid}(f, S)$ yielding a function from S into *Boolean* is defined by:

(Def. 41) $\mathrm{Fid}(f, S) = \begin{cases} f, & \text{if } f \in \mathrm{ModelSP}\, S, \\ S \longmapsto \mathit{false}, & \text{otherwise.} \end{cases}$

Now we present several schemes. The scheme *Func1EX* deals with a non empty set $\mathcal{A}$, a function $\mathcal{B}$ from $\mathcal{A}$ into *Boolean*, and a binary functor $\mathcal{F}$ yielding a boolean set, and states that:

> There exists a set g such that $g \in \mathrm{ModelSP}\, \mathcal{A}$ and for every set s such that $s \in \mathcal{A}$ holds $\mathcal{F}(s, \mathcal{B}) = \mathit{true}$ iff $(\mathrm{Fid}(g, \mathcal{A}))(s) = \mathit{true}$

for all values of the parameters.

The scheme *Func1Unique* deals with a non empty set $\mathcal{A}$, a function $\mathcal{B}$ from $\mathcal{A}$ into *Boolean*, and a binary functor $\mathcal{F}$ yielding a boolean set, and states that:

Let g_1, g_2 be sets. Suppose that
(i) $g_1 \in \mathrm{ModelSP}\,\mathcal{A}$,
(ii) for every set s such that $s \in \mathcal{A}$ holds $\mathcal{F}(s, \mathcal{B}) = \mathit{true}$ iff $(\mathrm{Fid}(g_1, \mathcal{A}))(s) = \mathit{true}$,
(iii) $g_2 \in \mathrm{ModelSP}\,\mathcal{A}$, and
(iv) for every set s such that $s \in \mathcal{A}$ holds $\mathcal{F}(s, \mathcal{B}) = \mathit{true}$ iff $(\mathrm{Fid}(g_2, \mathcal{A}))(s) = \mathit{true}$.
Then $g_1 = g_2$

for all values of the parameters.

The scheme *UnOpEX* deals with a non empty set $\mathcal{A}$ and a unary functor $\mathcal{F}$ yielding an element of $\mathcal{A}$, and states that:

There exists a unary operation o on $\mathcal{A}$ such that for every set f such that $f \in \mathcal{A}$ holds $o(f) = \mathcal{F}(f)$

for all values of the parameters.

The scheme *UnOpUnique* deals with a non empty set $\mathcal{A}$, a non empty set $\mathcal{B}$, and a unary functor $\mathcal{F}$ yielding an element of $\mathcal{B}$, and states that:

Let o_1, o_2 be unary operations on $\mathcal{B}$. Suppose for every set f such that $f \in \mathcal{B}$ holds $o_1(f) = \mathcal{F}(f)$ and for every set f such that $f \in \mathcal{B}$ holds $o_2(f) = \mathcal{F}(f)$. Then $o_1 = o_2$

for all values of the parameters.

The scheme *Func2EX* deals with a non empty set $\mathcal{A}$, a function $\mathcal{B}$ from $\mathcal{A}$ into *Boolean*, a function $\mathcal{C}$ from $\mathcal{A}$ into *Boolean*, and a ternary functor $\mathcal{F}$ yielding a boolean set, and states that:

There exists a set h such that $h \in \mathrm{ModelSP}\,\mathcal{A}$ and for every set s such that $s \in \mathcal{A}$ holds $\mathcal{F}(s, \mathcal{B}, \mathcal{C}) = \mathit{true}$ iff $(\mathrm{Fid}(h, \mathcal{A}))(s) = \mathit{true}$

for all values of the parameters.

The scheme *Func2Unique* deals with a non empty set $\mathcal{A}$, a function $\mathcal{B}$ from $\mathcal{A}$ into *Boolean*, a function $\mathcal{C}$ from $\mathcal{A}$ into *Boolean*, and a ternary functor $\mathcal{F}$ yielding a boolean set, and states that:

Let h_1, h_2 be sets. Suppose that
(i) $h_1 \in \mathrm{ModelSP}\,\mathcal{A}$,
(ii) for every set s such that $s \in \mathcal{A}$ holds $\mathcal{F}(s, \mathcal{B}, \mathcal{C}) = \mathit{true}$ iff $(\mathrm{Fid}(h_1, \mathcal{A}))(s) = \mathit{true}$,
(iii) $h_2 \in \mathrm{ModelSP}\,\mathcal{A}$, and
(iv) for every set s such that $s \in \mathcal{A}$ holds $\mathcal{F}(s, \mathcal{B}, \mathcal{C}) = \mathit{true}$ iff $(\mathrm{Fid}(h_2, \mathcal{A}))(s) = \mathit{true}$.
Then $h_1 = h_2$

for all values of the parameters.

Let S be a non empty set and let f be a set. The functor $\mathrm{Not}_0(f, S)$ yielding an element of $\mathrm{ModelSP}\,S$ is defined as follows:

(Def. 42) For every set s such that $s \in S$ holds $\neg\,\mathrm{Castboolean}(\mathrm{Fid}(f, S))(s) = \mathit{true}$ iff $(\mathrm{Fid}(\mathrm{Not}_0(f, S), S))(s) = \mathit{true}$.

Let S be a non empty set. The functor $\mathrm{Not}\,S$ yields a unary operation on $\mathrm{ModelSP}\,S$ and is defined by:

(Def. 43) For every set f such that $f \in \mathrm{ModelSP}\,S$ holds $(\mathrm{Not}\,S)(f) = \mathrm{Not}_0(f, S)$.

Let S be a non empty set, let R be a total relation between S and S, let f be a function from S into *Boolean*, and let x be a set. The functor $\mathrm{EneXt}_{\mathrm{univ}}(x, f, R)$ yielding an element of *Boolean* is defined by:

(Def. 44) $$\mathrm{EneXt}_{\mathrm{univ}}(x, f, R) = \begin{cases} \mathit{true}, & \\ \quad \text{if } x \in S \text{ and there exists an infinity path } p_1 & \\ \quad\; \text{of } R \text{ such that } p_1(0) = x \text{ and } f(p_1(1)) = \mathit{true}, & \\ \mathit{false}, \text{ otherwise.} & \end{cases}$$

Let S be a non empty set, let R be a total relation between S and S, and let f be a set. The functor $\mathrm{EneXt}_0(f, R)$ yielding an element of $\mathrm{ModelSP}\,S$ is defined as follows:

(Def. 45) For every set s such that $s \in S$ holds $\mathrm{EneXt}_{\mathrm{univ}}(s, \mathrm{Fid}(f, S), R) = \mathit{true}$ iff $(\mathrm{Fid}(\mathrm{EneXt}_0(f, R), S))(s) = \mathit{true}$.

Let S be a non empty set and let R be a total relation between S and S. The functor $\mathrm{EneXt}\,R$ yields a unary operation on $\mathrm{ModelSP}\,S$ and is defined by:

(Def. 46) For every set f such that $f \in \mathrm{ModelSP}\,S$ holds $(\mathrm{EneXt}\,R)(f) = \mathrm{EneXt}_0(f, R)$.

Let S be a non empty set, let R be a total relation between S and S, let f be a function from S into *Boolean*, and let x be a set. The functor $\mathrm{EGlobal}_{\mathrm{univ}}(x, f, R)$ yielding an element of *Boolean* is defined by:

(Def. 47) $$\mathrm{EGlobal}_{\mathrm{univ}}(x, f, R) = \begin{cases} \mathit{true}, & \\ \quad \text{if } x \in S \text{ and there exists an infinity path} & \\ \quad p_1 \text{ of } R \text{ such that } p_1(0) = x \text{ and for every} & \\ \quad \text{element } n \text{ of } \mathbb{N} \text{ holds } f(p_1(n)) = \mathit{true}, & \\ \mathit{false}, \text{ otherwise.} & \end{cases}$$

Let S be a non empty set, let R be a total relation between S and S, and let f be a set. The functor $\mathrm{EGlobal}_0(f, R)$ yielding an element of $\mathrm{ModelSP}\,S$ is defined as follows:

(Def. 48) For every set s such that $s \in S$ holds $\mathrm{EGlobal}_{\mathrm{univ}}(s, \mathrm{Fid}(f, S), R) = \mathit{true}$ iff $(\mathrm{Fid}(\mathrm{EGlobal}_0(f, R), S))(s) = \mathit{true}$.

Let S be a non empty set and let R be a total relation between S and S. The functor $\mathrm{EGlobal}\,R$ yields a unary operation on $\mathrm{ModelSP}\,S$ and is defined as follows:

(Def. 49) For every set f such that $f \in \mathrm{ModelSP}\,S$ holds $(\mathrm{EGlobal}\,R)(f) = \mathrm{EGlobal}_0(f, R)$.

Let S be a non empty set and let f, g be sets. The functor $\mathrm{And}_0(f,g,S)$ yields an element of ModelSP S and is defined as follows:

(Def. 50) For every set s such that $s \in S$ holds $\mathrm{Castboolean}(\mathrm{Fid}(f,S))(s) \wedge \mathrm{Castboolean}(\mathrm{Fid}(g,S))(s) = \mathit{true}$ iff $(\mathrm{Fid}(\mathrm{And}_0(f,g,S),S))(s) = \mathit{true}$.

Let S be a non empty set. The and S yielding a binary operation on ModelSP S is defined by:

(Def. 51) For all sets f, g such that $f \in \mathrm{ModelSP}\,S$ and $g \in \mathrm{ModelSP}\,S$ holds (the and S)$(f, g) = \mathrm{And}_0(f,g,S)$.

Let S be a non empty set, let R be a total relation between S and S, let f, g be functions from S into *Boolean*, and let x be a set. The functor $\mathrm{EUntill}_{\mathrm{univ}}(x,f,g,R)$ yielding an element of *Boolean* is defined as follows:

(Def. 52) $$\mathrm{EUntill}_{\mathrm{univ}}(x,f,g,R) = \begin{cases} \mathit{true}, \text{if } x \in S \text{ and there exists an infinity path } p_1 \text{ of } R \text{ such that } p_1(0) = x \text{ and there exists an element } m \text{ of } \mathbb{N} \text{ such that for every element } j \text{ of } \mathbb{N} \text{ such that } j < m \text{ holds } f(p_1(j)) = \mathit{true} \text{ and } g(p_1(m)) = \mathit{true}, \\ \mathit{false}, \text{otherwise.} \end{cases}$$

Let S be a non empty set, let R be a total relation between S and S, and let f, g be sets. The functor $\mathrm{EUntill}_0(f,g,R)$ yields an element of ModelSP S and is defined by:

(Def. 53) For every set s such that $s \in S$ holds $\mathrm{EUntill}_{\mathrm{univ}}(s,\mathrm{Fid}(f,S),\mathrm{Fid}(g,S),R) = \mathit{true}$ iff $(\mathrm{Fid}(\mathrm{EUntill}_0(f,g,R),S))(s) = \mathit{true}$.

Let S be a non empty set and let R be a total relation between S and S. The functor EUntill R yields a binary operation on ModelSP S and is defined as follows:

(Def. 54) For all sets f, g such that $f \in \mathrm{ModelSP}\,S$ and $g \in \mathrm{ModelSP}\,S$ holds $(\mathrm{EUntill}\,R)(f, g) = \mathrm{EUntill}_0(f,g,R)$.

Let S be a non empty set, let X be a non empty subset of ModelSP S, and let s be a set. The functor F-LABEL(s,X) yields a subset of X and is defined as follows:

(Def. 55) For every set x holds $x \in \text{F-LABEL}(s,X)$ iff $x \in X$ and there exists a function f from S into *Boolean* such that $f = x$ and $f(s) = \mathit{true}$.

Let S be a non empty set and let X be a non empty subset of ModelSP S. The functor Label X yields a function from S into 2^X and is defined by:

(Def. 56) For every set x such that $x \in S$ holds $(\mathrm{Label}\,X)(x) = \text{F-LABEL}(x,X)$.

Let S be a non empty set, let S_0 be a subset of S, let R be a total relation between S and S, and let P_1 be a non empty subset of ModelSP S. The functor $\mathrm{KModel}(R,S_0,P_1)$ yields a Kripke structure over P_1 and is defined as follows:

(Def. 57) $\mathrm{KModel}(R,S_0,P_1) = \langle S,S_0,R,\mathrm{Label}\,P_1\rangle$.

Let S be a non empty set, let S_0 be a subset of S, let R be a total relation between S and S, and let P_1 be a non empty subset of ModelSP S. One can check that the worlds of KModel(R, S_0, P_1) is non empty.

Let S be a non empty set, let S_0 be a subset of S, let R be a total relation between S and S, and let P_1 be a non empty subset of ModelSP S. One can verify that ModelSP (the worlds of KModel(R, S_0, P_1)) is non empty.

Let S be a non empty set, let R be a total relation between S and S, and let B_1 be a non empty subset of ModelSP S. The functor CTLModel(R, B_1) yielding a CTL model structure is defined as follows:

(Def. 58) $\mathrm{CTLModel}(R, B_1) = \langle \mathrm{ModelSP}\, S, B_1,$ the and S, $\mathrm{Not}\, S, \mathrm{EneXt}\, R, \mathrm{EGlobal}\, R, \mathrm{EUntill}\, R\rangle$.

In the sequel B_1 is a non empty subset of ModelSP S and k_1 is a function from the atomic WFF into the basic assignations of CTLModel(R, B_1).

Let S be a non empty set, let R be a total relation between S and S, let B_1 be a non empty subset of ModelSP S, let s be an element of S, and let f be an assignation of CTLModel(R, B_1). The predicate $s \models f$ is defined by:

(Def. 59) $(\mathrm{Fid}(f, S))(s) = \mathit{true}$.

Let S be a non empty set, let R be a total relation between S and S, let B_1 be a non empty subset of ModelSP S, let s be an element of S, and let f be an assignation of CTLModel(R, B_1). We introduce $s \not\models f$ as an antonym of $s \models f$.

Next we state several propositions:

(11) For every assignation a of CTLModel(R, B_1) such that $a \in B_1$ holds $s \models a$ iff $a \in (\mathrm{Label}\, B_1)(s)$.

(12) For every assignation f of CTLModel(R, B_1) holds $s \models \neg f$ iff $s \not\models f$.

(13) For all assignations f, g of CTLModel(R, B_1) holds $s \models f \wedge g$ iff $s \models f$ and $s \models g$.

(14) For every assignation f of CTLModel(R, B_1) holds $s \models \mathrm{EX}\, f$ iff there exists an infinity path p_1 of R such that $p_1(0) = s$ and $p_1(1) \models f$.

(15) Let f be an assignation of CTLModel(R, B_1). Then $s \models \mathrm{EG}\, f$ if and only if there exists an infinity path p_1 of R such that $p_1(0) = s$ and for every element n of $\mathbb{N}$ holds $p_1(n) \models f$.

(16) Let f, g be assignations of CTLModel(R, B_1). Then $s \models f\, \mathrm{EU}\, g$ if and only if there exists an infinity path p_1 of R such that $p_1(0) = s$ and there exists an element m of $\mathbb{N}$ such that for every element j of $\mathbb{N}$ such that $j < m$ holds $p_1(j) \models f$ and $p_1(m) \models g$.

(17) For all assignations f, g of CTLModel(R, B_1) holds $s \models f \vee g$ iff $s \models f$ or $s \models g$.

Let S be a non empty set, let R be a total relation between S and S, let B_1 be a non empty subset of ModelSP S, let k_1 be a function from the atomic

WFF into the basic assignations of CTLModel(R, B_1), let s be an element of S, and let H be a CTL-formula. The predicate $s \models_{k_1} H$ is defined by:

(Def. 60) $s \models \text{Evaluate}(H, k_1)$.

Let S be a non empty set, let R be a total relation between S and S, let B_1 be a non empty subset of ModelSP S, let k_1 be a function from the atomic WFF into the basic assignations of CTLModel(R, B_1), let s be an element of S, and let H be a CTL-formula. We introduce $s \not\models_{k_1} H$ as an antonym of $s \models_{k_1} H$.

The following propositions are true:

(18) If H is atomic, then $s \models_{k_1} H$ iff $k_1(H) \in (\text{Label}\, B_1)(s)$.

(19) $s \models_{k_1} \neg H$ iff $s \not\models_{k_1} H$.

(20) $s \models_{k_1} H_1 \wedge H_2$ iff $s \models_{k_1} H_1$ and $s \models_{k_1} H_2$.

(21) $s \models_{k_1} H_1 \vee H_2$ iff $s \models_{k_1} H_1$ or $s \models_{k_1} H_2$.

(22) $s \models_{k_1} \text{EX}\, H$ iff there exists an infinity path p_1 of R such that $p_1(0) = s$ and $p_1(1) \models_{k_1} H$.

(23) $s \models_{k_1} \text{EG}\, H$ iff there exists an infinity path p_1 of R such that $p_1(0) = s$ and for every element n of $\mathbb{N}$ holds $p_1(n) \models_{k_1} H$.

(24) $s \models_{k_1} H_1\, \text{EU}\, H_2$ if and only if there exists an infinity path p_1 of R such that $p_1(0) = s$ and there exists an element m of $\mathbb{N}$ such that for every element j of $\mathbb{N}$ such that $j < m$ holds $p_1(j) \models_{k_1} H_1$ and $p_1(m) \models_{k_1} H_2$.

(25) For every s_0 there exists an infinity path p_1 of R such that $p_1(0) = s_0$.

(26) Let R be a relation between S and S. Then R is total if and only if for every set x such that $x \in S$ there exists a set y such that $y \in S$ and $\langle x, y \rangle \in R$.

Let S be a non empty set, let R be a total relation between S and S, let s_0 be an element of S, let p_1 be an infinity path of R, and let n be a set. The functor PrePath(n, s_0, p_1) yielding an element of S is defined as follows:

(Def. 61) $\text{PrePath}(n, s_0, p_1) = \begin{cases} s_0, & \text{if } n = 0, \\ p_1(\text{k.nat}(\text{k.nat}\, n - 1)), & \text{otherwise.} \end{cases}$

The following propositions are true:

(27) If $\langle s_0, s_1 \rangle \in R$, then there exists an infinity path p_1 of R such that $p_1(0) = s_0$ and $p_1(1) = s_1$.

(28) For every assignation f of CTLModel(R, B_1) holds $s \models \text{EX}\, f$ iff there exists an element s_1 of S such that $\langle s, s_1 \rangle \in R$ and $s_1 \models f$.

Let S be a non empty set, let R be a total relation between S and S, and let H be a subset of S. The functor Pred(H, R) yields a subset of S and is defined by:

(Def. 62) $\text{Pred}(H, R) = \{s; s \text{ ranges over elements of } S\colon \bigvee_{t\,:\,\text{element of } S} (t \in H \wedge \langle s, t \rangle \in R)\}$.

Let S be a non empty set, let R be a total relation between S and S, let B_1 be a non empty subset of ModelSP S, and let f be an assignation of CTLModel(R, B_1). The functor SIGMA f yields a subset of S and is defined as follows:

(Def. 63) SIGMA $f = \{s; s$ ranges over elements of S: $s \models f\}$.

One can prove the following proposition

(29) For all assignations f, g of CTLModel(R, B_1) such that SIGMA f = SIGMA g holds $f = g$.

Let S be a non empty set, let R be a total relation between S and S, let B_1 be a non empty subset of ModelSP S, and let T be a subset of S. The functor Tau(T, R, B_1) yielding an assignation of CTLModel(R, B_1) is defined as follows:

(Def. 64) For every set s such that $s \in S$ holds $(\mathrm{Fid}(\mathrm{Tau}(T, R, B_1), S))(s) = \chi_{T,S}(s)$.

The following propositions are true:

(30) For all subsets T_1, T_2 of S such that $\mathrm{Tau}(T_1, R, B_1) = \mathrm{Tau}(T_2, R, B_1)$ holds $T_1 = T_2$.

(31) For every assignation f of CTLModel(R, B_1) holds Tau(SIGMA $f, R, B_1) = f$.

(32) For every subset T of S holds SIGMA Tau$(T, R, B_1) = T$.

(33) For all assignations f, g of CTLModel(R, B_1) holds SIGMA $\neg f = S \setminus$ SIGMA f and SIGMA$(f \wedge g)$ = SIGMA $f \cap$ SIGMA g and SIGMA$(f \vee g)$ = SIGMA $f \cup$ SIGMA g.

(34) For all subsets G_1, G_2 of S such that $G_1 \subseteq G_2$ and for every element s of S such that $s \models \mathrm{Tau}(G_1, R, B_1)$ holds $s \models \mathrm{Tau}(G_2, R, B_1)$.

(35) For all assignations f_1, f_2 of CTLModel(R, B_1) such that for every element s of S such that $s \models f_1$ holds $s \models f_2$ holds SIGMA $f_1 \subseteq$ SIGMA f_2.

Let S be a non empty set, let R be a total relation between S and S, let B_1 be a non empty subset of ModelSP S, and let f, g be assignations of CTLModel(R, B_1). The functor Fax(f, g) yielding an assignation of CTLModel(R, B_1) is defined by:

(Def. 65) $\mathrm{Fax}(f, g) = f \wedge \mathrm{EX}\, g$.

Next we state the proposition

(36) Let f, g_1, g_2 be assignations of CTLModel(R, B_1). Suppose that for every element s of S such that $s \models g_1$ holds $s \models g_2$. Let s be an element of S. If $s \models \mathrm{Fax}(f, g_1)$, then $s \models \mathrm{Fax}(f, g_2)$.

Let S be a non empty set, let R be a total relation between S and S, let B_1 be a non empty subset of ModelSP S, let f be an assignation of CTLModel(R, B_1), and let G be a subset of S. The functor SigFaxTau(f, G, R, B_1) yielding a subset of S is defined by:

(Def. 66) $\mathrm{SigFaxTau}(f, G, R, B_1) = \mathrm{SIGMA}\,\mathrm{Fax}(f, \mathrm{Tau}(G, R, B_1))$.

One can prove the following proposition

(37) For every assignation f of $\mathrm{CTLModel}(R, B_1)$ and for all subsets G_1, G_2 of S such that $G_1 \subseteq G_2$ holds $\mathrm{SigFaxTau}(f, G_1, R, B_1) \subseteq \mathrm{SigFaxTau}(f, G_2, R, B_1)$.

Let S be a non empty set, let R be a total relation between S and S, let p_1 be an infinity path of R, and let k be an element of $\mathbb{N}$. The functor $\mathrm{PathShift}(p_1, k)$ yielding an infinity path of R is defined as follows:

(Def. 67) For every element n of $\mathbb{N}$ holds $(\mathrm{PathShift}(p_1, k))(n) = p_1(n + k)$.

Let S be a non empty set, let R be a total relation between S and S, let p_2, p_3 be infinity paths of R, and let n, k be elements of $\mathbb{N}$. The functor $\mathrm{PathChange}(p_2, p_3, k, n)$ yielding a set is defined by:

(Def. 68) $\mathrm{PathChange}(p_2, p_3, k, n) = \begin{cases} p_2(n), & \text{if } n < k, \\ p_3(n - k), & \text{otherwise.} \end{cases}$

Let S be a non empty set, let R be a total relation between S and S, let p_2, p_3 be infinity paths of R, and let k be an element of $\mathbb{N}$. The functor $\mathrm{PathConc}(p_2, p_3, k)$ yielding a function from $\mathbb{N}$ into S is defined as follows:

(Def. 69) For every element n of $\mathbb{N}$ holds $(\mathrm{PathConc}(p_2, p_3, k))(n) = \mathrm{PathChange}(p_2, p_3, k, n)$.

We now state four propositions:

(38) Let p_2, p_3 be infinity paths of R and k be an element of $\mathbb{N}$. If $p_2(k) = p_3(0)$, then $\mathrm{PathConc}(p_2, p_3, k)$ is an infinity path of R.

(39) For every assignation f of $\mathrm{CTLModel}(R, B_1)$ and for every element s of S holds $s \models \mathrm{EG}\,f$ iff $s \models \mathrm{Fax}(f, \mathrm{EG}\,f)$.

(40) Let g be an assignation of $\mathrm{CTLModel}(R, B_1)$ and s_0 be an element of S. Suppose $s_0 \models g$. Suppose that for every element s of S such that $s \models g$ holds $s \models \mathrm{EX}\,g$. Then there exists an infinity path p_1 of R such that $p_1(0) = s_0$ and for every element n of $\mathbb{N}$ holds $p_1(n) \models g$.

(41) Let f, g be assignations of $\mathrm{CTLModel}(R, B_1)$. Suppose that for every element s of S holds $s \models g$ iff $s \models \mathrm{Fax}(f, g)$. Let s be an element of S. If $s \models g$, then $s \models \mathrm{EG}\,f$.

Let S be a non empty set, let R be a total relation between S and S, let B_1 be a non empty subset of $\mathrm{ModelSP}\,S$, and let f be an assignation of $\mathrm{CTLModel}(R, B_1)$. The functor $\mathrm{TransEG}\,f$ yielding a $\subseteq$-monotone function from 2^S into 2^S is defined as follows:

(Def. 70) For every subset G of S holds $(\mathrm{TransEG}\,f)(G) = \mathrm{SigFaxTau}(f, G, R, B_1)$.

One can prove the following two propositions:

(42) Let f, g be assignations of $\mathrm{CTLModel}(R, B_1)$. Then for every element s of S holds $s \models g$ iff $s \models \mathrm{Fax}(f, g)$ if and only if $\mathrm{SIGMA}\,g$ is a fixpoint of

TransEG f.

(43) For every assignation f of CTLModel(R, B_1) holds SIGMA EG f = gfp$(S, \text{TransEG}\, f)$.

Let S be a non empty set, let R be a total relation between S and S, let B_1 be a non empty subset of ModelSP S, and let f, g, h be assignations of CTLModel(R, B_1). The functor Foax(g, f, h) yields an assignation of CTLModel(R, B_1) and is defined as follows:

(Def. 71) $\text{Foax}(g, f, h) = g \vee \text{Fax}(f, h)$.

We now state the proposition

(44) Let f, g, h_1, h_2 be assignations of CTLModel(R, B_1). Suppose that for every element s of S such that $s \models h_1$ holds $s \models h_2$. Let s be an element of S. If $s \models \text{Foax}(g, f, h_1)$, then $s \models \text{Foax}(g, f, h_2)$.

Let S be a non empty set, let R be a total relation between S and S, let B_1 be a non empty subset of ModelSP S, let f, g be assignations of CTLModel(R, B_1), and let H be a subset of S. The functor SigFoaxTau(g, f, H, R, B_1) yields a subset of S and is defined as follows:

(Def. 72) $\text{SigFoaxTau}(g, f, H, R, B_1) = \text{SIGMA}\, \text{Foax}(g, f, \text{Tau}(H, R, B_1))$.

Next we state three propositions:

(45) For all assignations f, g of CTLModel(R, B_1) and for all subsets H_1, H_2 of S such that $H_1 \subseteq H_2$ holds $\text{SigFoaxTau}(g, f, H_1, R, B_1) \subseteq \text{SigFoaxTau}(g, f, H_2, R, B_1)$.

(46) For all assignations f, g of CTLModel(R, B_1) and for every element s of S holds $s \models f\, \text{EU}\, g$ iff $s \models \text{Foax}(g, f, f\, \text{EU}\, g)$.

(47) Let f, g, h be assignations of CTLModel(R, B_1). Suppose that for every element s of S holds $s \models h$ iff $s \models \text{Foax}(g, f, h)$. Let s be an element of S. If $s \models f\, \text{EU}\, g$, then $s \models h$.

Let S be a non empty set, let R be a total relation between S and S, let B_1 be a non empty subset of ModelSP S, and let f, g be assignations of CTLModel(R, B_1). The functor TransEU(f, g) yields a $\subseteq$-monotone function from 2^S into 2^S and is defined by:

(Def. 73) For every subset H of S holds $(\text{TransEU}(f, g))(H) = \text{SigFoaxTau}(g, f, H, R, B_1)$.

One can prove the following propositions:

(48) Let f, g, h be assignations of CTLModel(R, B_1). Then for every element s of S holds $s \models h$ iff $s \models \text{Foax}(g, f, h)$ if and only if SIGMA h is a fixpoint of TransEU(f, g).

(49) For all assignations f, g of CTLModel(R, B_1) holds $\text{SIGMA}(f\, \text{EU}\, g) = \text{lfp}(S, \text{TransEU}(f, g))$.

(50) For every assignation f of CTLModel(R, B_1) holds SIGMA EX f = Pred(SIGMA f, R).

(51) For every assignation f of CTLModel(R, B_1) and for every subset X of S holds (TransEG f)(X) = SIGMA $f \cap$ Pred(X, R).

(52) For all assignations f, g of CTLModel(R, B_1) and for every subset X of S holds (TransEU(f, g))(X) = SIGMA $g \cup$ SIGMA $f \cap$ Pred(X, R).

References

[1] Grzegorz Bancerek. A model of ZF set theory language. *Formalized Mathematics*, 1(**1**):131–145, 1990.

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

[3] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.

[4] Czesław Byliński. Basic functions and operations on functions. *Formalized Mathematics*, 1(**1**):245–254, 1990.

[5] Czesław Byliński. Binary operations. *Formalized Mathematics*, 1(**1**):175–180, 1990.

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

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

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

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

[10] Grumberg, O. and Clarke, E. M. and D. Peled. *Model Checking.* MIT Press, 2000.

[11] Library Committee of the Association of Mizar Users. Binary operations on numbers. *To appear in Formalized Mathematics.*

[12] Piotr Rudnicki and Andrzej Trybulec. Abian's fixed point theorem. *Formalized Mathematics*, 6(**3**):335–338, 1997.

[13] Piotr Rudnicki and Andrzej Trybulec. Fixpoints in complete lattices. *Formalized Mathematics*, 6(**1**):109–115, 1997.

[14] Andrzej Trybulec. Subsets of complex numbers. *To appear in Formalized Mathematics.*

[15] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.

[16] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.

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

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

[19] Edmund Woronowicz. Many–argument relations. *Formalized Mathematics*, 1(**4**):733–737, 1990.

[20] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

[21] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

Received November 14, 2006