Formalizing and Implementing Types in MSVL

X e D (✉)

Institute of Computing Theory and Technology and ISN Laboratory,
Xidian University, Xian 710071, People's Republic of China
xbwang@mail.xidian.edu.cn, zhenhua_duan@126.com, lzhao@xidian.edu.cn

Abstract. This paper investigates techniques for formalizing and implementing types in the temporal logic programming language MSVL, which is an executable subset of Projection Temporal Logic. To this end, the data domain of MSVL is extended to include typed values, and then typed functions and predicates concerning the extended data domain are defined. Based on these definitions, the statement for type declaration of program variables is formalized. The implementation mechanisms of the type declaration statement in the MSVL interpreter are also discussed, which is based on the notion of normal form of MSVL programs. To illustrate how to program with types, an example of in-place reversing an integer list is given.

Keywords: Type · Temporal logic programming · MSVL · Projection Temporal Logic

1 Introduction

This research is supported by the NSFC Grant Nos. 61272118, 61272117, 61133001, 61202038, 61373043, 973 Program Grant No. 2010CB328102, and the Fundamental Research Funds for the Central Universities Nos. K5051203014, K5051303020.

S. Liu and Z. Duan (Eds.): SOFL+MSVL 2013, LNCS 8332, pp. 62–75, 2014.
DOI: 10.1007/978-3-319-04915-1_5,

2 Preliminaries

2.1 Projection Temporal Logic

$$
\begin{array}{ll}
e :: & x \mid \bigcirc e \mid \bigcirc e \mid f(e_1, \ldots, e_m) \\
p :: & r \mid e_1 \quad e_2 \mid P(e_1, \ldots, e_m) \mid \neg p \mid p_1 \wedge p_2 \mid \exists v : p \mid \\
& \bigcirc p \mid (p_1, \ldots, p_m)\,\mathsf{prj}\,p
\end{array}
$$

(e) , es ✱ , , e , , s (e ◯ prj) e w se s , *temporal* , , (e)

A s , e s s , , , ss e s (I_v, I_p) w e e , e ✱ , , , e x e es s x I_v x , , , e ✱ , , s r e es s r I_p r I_v x s , , , e e , , , , , $\mathcal{D}$ *nil* w ✱ e s e e , , I_p r s , , , e $\mathbb{B}$ {true, false} A e , , σ $\langle s_0, s_1, \ldots \rangle$ s , , e , (ss , , e) se e ✱ e , s , es e e , , σ e e , $|\sigma|$ s e e , s ω , σ s , , e e w se s e e , s , es σ s e , , e , , , , , , , , , e , , , , e e , s we w se e e e e e s , s ✱ es , s we ✱ s e e se $\mathbb{N}$, , , , , e s , ω $\mathbb{N}_\omega$ $\mathbb{N} \cup \{\omega\}$, e e e ✱ , , s e , s , <, ≤ $\mathbb{N}_\omega$ ✱ s e ω ω , , , $i \in \mathbb{N}$ $i < \omega$ _ e e we e e $\preceq$, s $\leq -\{(\omega, \omega)\}$ s ✱ , , , , , $\sigma_{(i..j)}$($\leq i \preceq j \leq |\sigma|$) e es e s e , , $\langle s_i, \ldots, s_j \rangle$, $\sigma(k)$($\leq k \preceq |\sigma|$) e e es $\langle s_k, \ldots, s_{|\sigma|} \rangle$ e ✱ ✱ e , , , σ w , , e e , , (e s ,) σ' s e e $\sigma \bullet \sigma'$ e e e se , ✱s , e , e ✱ , , e , we ee , , , , , e , , , , e , s e σ $\langle s_0, s_1, \ldots \rangle$ e , , e , , , $n_1, \ldots, n_h$ e e e s ($h \geq$) s ✱ , , $\leq n_1 \leq n_2 \leq \ldots \leq n_h \preceq |\sigma|$ e , e ✱ , , , σ $n_1, \ldots, n_h$ s e e , , (✱ e , , e ✱ e , ,) $\sigma \downarrow (n_1, \ldots, n_h)$ $\langle s_{m_1}, s_{m_2}, \ldots, s_{m_l} \rangle$ w e e $m_1, \ldots, m_l$ s , , e , , , $n_1, \ldots, n_h$, e e , , , , , ✱ es _ e , , e

$$\langle s_0, s_1, s_2, s_3, s_4 \rangle \downarrow (\ , \ , \ , \ , \ , \ ) \quad \langle s_0, s_2, s_3 \rangle$$

A , e , e , , , , e , , , , , s , , e $\mathcal{I}$ (σ, i, k, j) w e e σ $\langle s_0, s_1, \ldots \rangle$ s , , e , , , i , k , e , , , e , , e e e s , , , j s , , e e , , , ω s ✱ , , $i \leq k \preceq j \leq |\sigma|$ e se (σ, i, k, j) e , , , , e , , , , , , s , e , e e , , e , , s , e , , $\sigma_{(i..j)}$ w , , e ✱ , e s , e , e , s_k _ , e e , e , e e , , , , , , e , e , , , e , , , , $\mathcal{I}$ (σ, i, k, j) e , , e , s $\mathcal{I}$ e s , , , e , $\mathcal{D}$ *nil* s e , e , s , ✱ , , , ✱ , , s , w , _ ,

$$\begin{aligned}
\mathcal{I}[x] &= s_k[x] = I_v^k[x] \\
\mathcal{I}[\bigcirc e] &= \begin{cases} (\sigma, i, k+1, j)[e] & \text{if } k < j \\ \mathit{nil} & \text{otherwise} \end{cases} \\
\mathcal{I}[\ominus e] &= \begin{cases} (\sigma, i, k-1, j)[e] & \text{if } i < k \\ \mathit{nil} & \text{otherwise} \end{cases} \\
\mathcal{I}[f(e_1, \ldots, e_m)] &= \begin{cases} \mathcal{I}[f](\mathcal{I}[e_1], \ldots, \mathcal{I}[e_m]) & \text{if } \mathcal{I}[e_h] \neq \mathit{nil} \text{ for all } h \\ \mathit{nil} & \text{otherwise} \end{cases}
\end{aligned}$$

Fig. 1. Interpretation of PTL terms

e s , s , ✱ , e , , , , , , s | s , , ✱ , e e , e , s , , , ws

, $\mathcal{I}$ | r , s_k r I_p^k r true
, $\mathcal{I}$ | e_1 e_2 , $\mathcal{I}$ e_1 $\mathcal{I}$ e_2

$\mathcal{I} \mid P(e_1,\ldots,e_m)$ $\mathcal{I} e_h$ $/$ nil $\le h \le m$ $\mathcal{I} P$ $(\mathcal{I} e_1, \ldots, \mathcal{I} e_m)$ true

$\mathcal{I} \mid \neg p$ $\mathcal{I} \not\mid p$

$\mathcal{I} \mid p_1 \wedge p_2$ $\mathcal{I} \mid p_1$ $\mathcal{I} \mid p_2$

$\mathcal{I} \mid \exists v$ p σ' σ $(\sigma', i, k, j) \mid$ p ffe σ σ' v k

$\mathcal{I} \mid \bigcirc p$ $k < j$ $(\sigma, i, k$ $, j) \mid p$

$\mathcal{I} \mid (p_1, \ldots, p_m)$ prj q k $k_0 \le k_1 \le \ldots \le k_m \le j$ $(\sigma, i, k_0, k_1) \mid p_1$ $(\sigma, k_{l-1}, k_{l-1}, k_l) \mid p_l$ $(< l \le m)$ $(\sigma', , , |\sigma'|) \mid q$ σ'

( ) $k_m < j$ σ' $\sigma \downarrow (k_0, \ldots, k_m) \bullet \sigma_{(k_m+1..j)}$

( ) k_m j σ' $\sigma \downarrow (k_0, \ldots, k_h)$ $\le h \le m$

A p

satisfied σ $\sigma \mid p$ $(\sigma, , , |\sigma|) \mid p$

satisfiable $\sigma \mid p$ σ

valid $\mid p$ $\sigma \mid p$ σ

equivalent q $p \equiv q$ $\mid \Box(p \leftrightarrow q)$

$\vee$ $\rightarrow$ $\leftrightarrow$ true $\stackrel{\text{def}}{=} p \vee \neg p$ false $\stackrel{\text{def}}{=} p \wedge \neg p$ p

empty	$\stackrel{\text{def}}{=}$	$\neg \bigcirc$ true	more	$\stackrel{\text{def}}{=}$	$\neg$empty
$p; q$	$\stackrel{\text{def}}{=}$	(p, q) prj empty	$\Diamond p$	$\stackrel{\text{def}}{=}$	true; p
$\Box p$	$\stackrel{\text{def}}{=}$	$\neg \Diamond \neg p$	halt(p)	$\stackrel{\text{def}}{=}$	$\Box$(empty $\leftrightarrow p$)
keep(p)	$\stackrel{\text{def}}{=}$	$\Box(\neg$empty $\rightarrow p)$	fin(p)	$\stackrel{\text{def}}{=}$	$\Box$(empty $\rightarrow p$)
skip	$\stackrel{\text{def}}{=}$	$\bigcirc$empty	len(n)	$\stackrel{\text{def}}{=}$	$\bigcirc$len$(n-1)$ for $n > 0$
len(0)	$\stackrel{\text{def}}{=}$	empty			
p^*	$\stackrel{\text{def}}{=}$	empty $\vee (p; p^*) \vee p \wedge \Box$more			

Fig. 2. Derived formulas and composite predicates

A

Table 1. Precedence rules of PTL

1	$\neg$	2	$\bigcirc$ $\ominus$ $\Diamond$ $\Box$	3	$\wedge$	4	$\vee$
5	:= =	6	$\rightarrow$ $\leftrightarrow$	7	prj	8	;

2.2 MSVL

[illegible]

$$e ::= n \mid x \mid \bigcirc x \mid \bigcirc x \mid e_1 \; e_2 \mid e_1 - e_2 \mid e_1 * e_2 \mid e_1/e_2 \mid e_1 \, mod \, e_2$$
$$b ::= \mathsf{true} \mid \mathsf{false} \mid e_1 \; e_2 \mid e_1 > e_2 \mid e_1 \geq e_2 \mid e_1 < e_2 \mid e_1 \leq e_2 \mid \neg b \mid b_1 \wedge b_2$$

[illegible] p q [illegible]

Assignment:	$x = e$
P-I-Assignment:	$x \Leftarrow e \overset{\text{def}}{=} x = e \wedge r_x$
Unit Assignment:	$x := e \overset{\text{def}}{=} \mathsf{skip} \wedge \bigcirc x \Leftarrow e$
Sequential Composition:	$p; q$
Conditional Choice:	$\texttt{if } b \texttt{ then } p \texttt{ else } q \overset{\text{def}}{=} (b \rightarrow p) \wedge (\neg b \rightarrow q)$
While Loop:	$\texttt{while } b \texttt{ do } p \overset{\text{def}}{=} (b \wedge p)^* \wedge \Box(\mathsf{empty} \rightarrow \neg b)$
Conjunction:	$p \wedge q$
Selection:	$p \vee q$
Parallel Composition:	$p \parallel q \overset{\text{def}}{=} p \wedge (q; \mathsf{true}) \vee q \wedge (p; \mathsf{true})$
Next:	$\bigcirc p$
Always:	$\Box p$
Termination:	empty
Local variable:	$\exists x : p$
State Frame:	$\mathsf{lbf}(x) \overset{\text{def}}{=} \neg r_x \rightarrow \exists z : (\ominus x = z \wedge x = z)$
Interval Frame:	$\mathsf{frame}(x) \overset{\text{def}}{=} \Box(\mathsf{more} \rightarrow \bigcirc \mathsf{lbf}(x))$
Projection:	$(p_1, \ldots, p_m)\, \mathsf{prj}\, q$
Await:	$\mathsf{await}(b) \overset{\text{def}}{=} (\mathsf{frame}(x_1) \wedge \ldots \wedge \mathsf{frame}(x_h)) \wedge \Box(\mathsf{empty} \leftrightarrow b)$, where $x_1, \ldots, x_h$ are the variables that occur in b

[illegible] x [illegible] x [illegible] $x \Leftarrow e$ [illegible] empty [illegible] $\mathsf{lbf}(x)$ [illegible] $\mathsf{frame}(x)$ [illegible] x [illegible] x [illegible] x [illegible] x [illegible] $x \Leftarrow e$ [illegible] x [illegible] r_x [illegible] x [illegible] p q [illegible] p [illegible] q [illegible] $\texttt{if } b \texttt{ then } p \texttt{ else } q$ [illegible] $\texttt{while } b \texttt{ do } p$ [illegible] $p \wedge q$ [illegible] p [illegible] q [illegible] $p \vee q$ [illegible] p [illegible] q [illegible] $\mathsf{len}(\;) \parallel \mathsf{len}(\;)$ [illegible] $\mathsf{len}(\;) \wedge \mathsf{len}(\;)$ [illegible] $\bigcirc p$ [illegible] p [illegible]

$\Box p$ e s , p s , s , es e e e e , e e
s , e e empty e s , e e s , e s e , s , e , e e ,
A e s e , , , e ... $\exists x$, p e s e x w p A s , e e
lbf(x) e s e , e , x e e s , e e , s e , e , x e
e s s , e , , ss e x s e e e w e frame(x) e es
, , e , e , , , e x , w s ee s s , , e e , e , ,
, ss e x s e e e A e e s , e e e , s ,
s e e , , , e e e e , s e , e ffe e es e es e e
$(p_1, \ldots, p_m)$ prj q e s , q s e e e e w $p_1, \ldots, p_m$ e ,
e , , e , , e e s , e e , s e w e p_i s , e
e e e , , , e se e e , p_i s , q , e , e , ffe e
e , s _ , , , w s , e e await(b) s , w s , b e es
e w , , , , , , es
e e e e e e es , _ , s , e e s , e s e , , e w e e ,
e s , es , , , e s wes

Table 2. Precedence rules of MSVL

1		$\neg$	2	$\bigcirc$	$\ominus$	$\Box$	3	$*$	/	*mod*	4	$+$	$-$	5	$>$	$\geq$	$<$	$\leq$	6	$\exists$
7	$=$	$\Leftarrow$	:=	8	$\wedge$		9	$\vee$	$\|$		10	$\rightarrow$	$\leftrightarrow$	11	prj				12	;

3 Typed MSVL

3.1 Data Domain

e s e s (I_v, I_p) , , , , e x e e s $I_v\, x \in \mathcal{D}$,
, , , e y s e e , (e e e) s we w e $I_v\, y$ *nil* , ,
e e , $\mathcal{I}$, , , e x $\mathcal{I}\, x \in \mathcal{D} \cup \{nil\}$ e e e e
e e , I_v , , , es e , es we ee e , e , e , ,
D e e _ , se $\mathcal{T}$, es

, s es , , , ,
s es , $\langle\rangle$, $\langle\rangle$, $\langle\rangle$
, , es , , ,

e se , , es , e , s e e , e , e , s , ws

, $\mathbb{Z}$
, F $\stackrel{\text{def}}{=}$ $\{n.d_1d_2\cdots d_m \mid m \in \mathbb{N}, n \in \mathbb{Z}, d_i \in \{\ ,\ldots,\ \}for\ \leq i \leq m\}$
, C $\stackrel{\text{def}}{=}$ $\{'a', \ldots, 'z', 'A', \ldots, 'Z', '\ ', \ldots, '/', '!', '\#', '\$', \ldots\}$

s e e , , es (v, T) e , es , e e e es w e e $T \in \mathcal{T}$
e e e e se , , e , , es , e , s e s , e _ , se S S^n e es e se
, s s , e n S $(n \in \mathbb{N})$, S^* e es e se , s s S

$\stackrel{\text{def}}{\mathbb{Z} \times \{ \quad \}} \quad \langle\rangle \stackrel{\text{def}}{\mathbb{Z}^* \times \{ \quad \langle\rangle\}}$

$n \stackrel{\text{def}}{\mathbb{Z}^n \times \{ \quad \}} \; n \geq \quad \stackrel{\text{def}}{\bigcup_{n\geq 1} \quad n}$

$\stackrel{\text{def}}{\mathtt{F} \times \{ \quad \}} \quad \langle\rangle \stackrel{\text{def}}{\mathtt{F}^* \times \{ \quad \langle\rangle\}}$

$n \stackrel{\text{def}}{\mathtt{F}^n \times \{ \quad \}} \; n \geq \quad \stackrel{\text{def}}{\bigcup_{n\geq 1} \quad n}$

$\stackrel{\text{def}}{\mathtt{C} \times \{ \quad \}} \quad \langle\rangle \stackrel{\text{def}}{\mathtt{C}^* \times \{ \quad \langle\rangle\}}$

$n \stackrel{\text{def}}{\mathtt{C}^n \times \{ \quad \}} \; n \geq \quad \stackrel{\text{def}}{\bigcup_{n\geq 1} \quad n}$

$D \quad$ s

$\mathcal{D} \stackrel{\text{def}}{\quad \cup \quad \langle\rangle \cup \quad \cup \quad \cup \quad \langle\rangle \cup \quad \cup \quad \cup \quad \langle\rangle \cup}$

3.2 Typed Functions and Predicates

$f \quad m \quad T_1 \times \ldots \times T_m \to T \quad n \quad T_1 \times \ldots \times T_n \to \mathbb{B} \quad T_i \quad T \quad \mathcal{T} \quad \cdot \perp \cdot \quad \cdot > \cdot \quad \times \quad \to \quad \cdot > \cdot \quad \times \quad \to \mathbb{B}$

I_v

$f \quad T_1 \times \ldots \times T_m \to T \quad f(v_1, \ldots, v_m) \quad v \quad f((v_1, T_1), \ldots, (v_m, T_m)) \quad (v, T)$

$P \quad T_1 \times \ldots \times T_m \to \mathbb{B} \quad P(v_1, \ldots, v_m) \quad \mathsf{true} \; (\text{ } \mathsf{false}) \quad P((v_1, T_1), \ldots, (v_m, T_m)) \quad \mathsf{true} \; (\text{ } \mathsf{false})$

$nil \quad \mathsf{false} \quad \mathcal{I} \; e \in \mathcal{D} \cup \{nil\} \quad e \quad \mathcal{I} \quad (\quad , \quad) / (\quad , \quad) \quad nil \quad hd((\langle\rangle, \quad \langle\rangle)) \quad nil$

$\mathcal{I} \; (\quad , \quad) \quad \mathcal{I} \; , \quad (\quad , \quad , \quad)$

$\mathcal{I} \; \langle x', y' \rangle \quad (\langle x', y' \rangle, \quad \langle\rangle)$

$\mathcal{I} \; \langle\rangle_i \quad (\langle\rangle, \quad \langle\rangle) \quad \mathcal{I} \; \langle\rangle_f \quad (\langle\rangle, \quad \langle\rangle) \quad \mathcal{I} \; \langle\rangle_c \quad (\langle\rangle, \quad \langle\rangle)$

$(\langle\rangle_i) \quad (\langle\rangle_f) \quad (\langle\rangle_c)$

$\perp, -, *, /, mod \quad >, \geq, <, \leq \quad (\quad - \quad * \quad /)$

$\cdot \perp_f \cdot \quad \times \quad \to$

[illegible]

$$\cdot \llcorner \cdot : (\quad \times \quad \to \quad) \cup (\quad \times \quad \to \quad)$$

[illegible] f [illegible] P) [illegible] f (P) [illegible]

Arithmetic operators. [illegible] $-$, $*$, $/$ [illegible]

$$\cdot \llcorner \cdot, \cdot - \cdot, \cdot * \cdot, \cdot / \cdot : (\quad \times \quad \to \quad) \cup (\quad \times \quad \to \quad)$$

[illegible] mod : $\times \to$ [illegible] $>, \geq, <, \leq$ [illegible]

$$\cdot > \cdot, \cdot \geq \cdot, \cdot < \cdot, \cdot \leq \cdot : (\quad \times \quad \to \mathbb{B}) \cup (\quad \times \quad \to \mathbb{B})$$

Type cast. [illegible]

$$\begin{array}{ll} (\quad) \cdot : \quad \to & (n.d_1 d_2 \cdots d_m, \quad) \mapsto (n, \quad) \\ (\quad) \cdot : \quad \to & (n, \quad) \mapsto (n., \quad) \end{array}$$

Array and list operations. [illegible] a [illegible] $a\,i$ [illegible]

$$\cdot \cdot : (\quad \times \quad \to \quad) \cup (\quad \times \quad \to \quad) \cup (\quad \times \quad \to \quad)$$
$$(\langle c_0, \ldots, c_k \rangle, T\ ), (i, \quad) \mapsto (c_i, T) \qquad i \in \{\ , \ldots, k\}$$
$$(\langle c_0, \ldots, c_k \rangle, T\ ), (i, \quad) \mapsto nil \qquad i \notin \{\ , \ldots, k\}$$
$$k \in \mathbb{N}, T \in \{\ ,\ ,\ ,\ \}$$

[illegible] l [illegible] $|l|$ [illegible] l

$$|\cdot| : (\quad \langle\rangle \to \quad) \cup (\quad \langle\rangle \to \quad) \cup (\quad \langle\rangle \to \quad)$$
$$(\langle c_1, \ldots, c_k \rangle, T) \mapsto (k, \quad) \qquad k \in \mathbb{N}, T \in \{\ \langle\rangle, \quad \langle\rangle, \quad \langle\rangle\}$$

[illegible] $hd(l)$, $tl(l)$ [illegible] l [illegible]

$$hd : (\quad \langle\rangle \to \quad) \cup (\quad \langle\rangle \to \quad) \cup (\quad \langle\rangle \to \quad)$$
$$(\langle\rangle, T\langle\rangle) \mapsto nil$$
$$(\langle c_0, c_1, \ldots, c_k \rangle, T\langle\rangle) \mapsto (c_0, T) \qquad k \in \mathbb{N}, T \in \{\ ,\ ,\ \}$$
$$tl : (\quad \langle\rangle \to \quad \langle\rangle) \cup (\quad \langle\rangle \to \quad \langle\rangle) \cup (\quad \langle\rangle \to \quad \langle\rangle)$$
$$(\langle\rangle, T) \mapsto nil$$
$$(\langle c_0, c_1, \ldots, c_k \rangle, T) \mapsto (\langle c_1, \ldots, c_k \rangle, T) \quad k \in \mathbb{N}, T \in \{\ \langle\rangle, \quad \langle\rangle, \quad \langle\rangle\}$$

[illegible] $l_1 \bullet l_2$, $l_1 \circ l_2$ [illegible]

$$\cdot \bullet \cdot : (\quad \langle\rangle \times \quad \langle\rangle \to \quad \langle\rangle) \cup (\quad \langle\rangle \times \quad \langle\rangle \to \quad \langle\rangle)$$
$$\cup (\quad \langle\rangle \times \quad \langle\rangle \to \quad \langle\rangle)$$
$$(\langle c_1, \ldots, c_j \rangle, T), (\langle d_1, \ldots, d_k \rangle, T) \mapsto (\langle c_1, \ldots, c_j, d_1, \ldots, d_k \rangle, T)$$
$$j, k \in \mathbb{N}, T \in \{\ \langle\rangle, \quad \langle\rangle, \quad \langle\rangle\}$$

$$\cdot \circ \cdot : (\ \langle\rangle \times \ \langle\rangle \to \ \langle\rangle) \cup (\ \langle\rangle \times \ \langle\rangle \to \ \langle\rangle)$$
$$\cup(\ \langle\rangle \times \ \langle\rangle \to \ \langle\rangle)$$
$$(\langle\rangle, T), (\langle d_1, \ldots, d_k\rangle, T) \mapsto (\langle d_1, \ldots, d_k\rangle, T)$$
$$(\langle c_1, \ldots, c_j, c\rangle, T), (\langle\rangle, T) \mapsto (\langle c_1, \ldots, c_j, c\rangle, T)$$
$$(\langle c_1, \ldots, c_j, c\rangle, T), (\langle c, d_1, \ldots, d_k\rangle, T) \mapsto (\langle c_1, \ldots, c_j, c, d_1, \ldots, d_k\rangle, T)$$
$$(\langle c_1, \ldots, c_j, c\rangle, T), (\langle d, d_1, \ldots, d_k\rangle, T) \mapsto nil \quad d \ / \ c$$
$$j, k \in \mathbb{N}, T \in \{\ \langle\rangle, \ \langle\rangle, \ \langle\rangle\}$$

Other. ... we ... De () ... e De (e) ... *nil*

$$\mathrm{De} : (\ \to \mathbb{B}) \cup (\ \to \mathbb{B}) \cup (\ \to \mathbb{B}) \cup (\ \langle\rangle \to \mathbb{B}) \cup (\ \langle\rangle \to \mathbb{B})$$
$$\cup(\ \langle\rangle \to \mathbb{B})$$
$$(c, T) \mapsto \mathsf{true} \quad T \in \{\ ,\ ,\ ,\ \langle\rangle, \ \langle\rangle, \ \langle\rangle\}$$

... *nil* ... $e_1 \leq e_2 \equiv \neg(e_1 > e_2)$ ... $e_1 \leq e_2 \equiv \mathrm{De}(e_1) \wedge \mathrm{De}(e_2) \wedge \neg(e_1 > e_2)$ ... De () ... $e_1 \leq e_2$ ... $\mathrm{De}(e_1) \wedge \mathrm{De}(e_2) \wedge (e_1 < e_2 \vee e_1 \ \ e_2)$

3.3 Type Declaration Statement

... we ... we ... *a* ... $\mathcal{T}_d$ ... ffe ... $\mathcal{T}$

$$\mathcal{T}_d \ \stackrel{\text{def}}{} \ \{\ ,\ ,\ ,\ \langle\rangle, \ \langle\rangle, \ \langle\rangle, \ \ldots, \ \ldots, \ \ldots\}.$$

... $s_T(\cdot)$ w ... T ... $T \in \mathcal{T}_d$

- ... $T \in \{\ ,\ ,\ \}$ ... $s_T : T \to \mathbb{B}$ w ... $(c, T) \mapsto \mathsf{true}$
- ... $T\langle\rangle \in \{\ \langle\rangle, \ \langle\rangle, \ \langle\rangle\}$ ... $s_{T\langle\rangle} : T\langle\rangle \to \mathbb{B}$ w ... $(c, T\langle\rangle) \mapsto \mathsf{true}$ ...
- ... $T\ n \in \{\ ,\ \ldots,\ \ldots,\ \ldots\}$ ... $s_{T[n]} : T \to \mathbb{B}$ w ... $(c, T) \mapsto \mathsf{true}$ ff $|c| \ \ n$

... we ...

$$T\ x \ \stackrel{\text{def}}{} \ \Box\, s_T(x)$$

4 Implementation Mechanisms

... w ...

4.1 Normal Form of Programs

Definition 1. *A typed MSVL program q is in normal form if*

$$q \stackrel{\text{def}}{=} (\bigvee_{i=1}^{k} q_{ei} \wedge \mathsf{empty}) \vee (\bigvee_{j=1}^{h} q_{cj} \wedge \bigcirc q_{fj})$$

where $k \geq 0$, $h \geq 0$ and the following hold:

1. *each q_{ei} and q_{cj} is either true or a state formula of the form $p_1 \wedge \ldots \wedge p_m$ ($m \geq 1$) such that each p_l ($1 \leq l \leq m$) is either $s_T(x)$ with $x \in \mathcal{V}$, $T \in \mathcal{T}_d$, or $x = e$ with $e \in D$, or r_x, or $\neg r_x$.*
2. *q_{fj} is an internal program, that is, one in which variables may refer to the previous states but not beyond the first state of the current interval over which the program is executed.*

[illegible] $k \geq 0 \wedge h > 0$ [illegible] $\bigvee_{i=1}^{k} q_{ei} \wedge empty \bigvee_{j=1}^{h} q_{cj} \wedge \bigcirc q_{fj}$ [illegible]

$$\begin{aligned}
T\,x &\equiv \Box\, s_T(x) \\
&\equiv \Box\, s_T(x) \wedge (\mathsf{empty} \vee \neg\mathsf{empty}) && (i)\\
&\equiv \Box\, s_T(x) \wedge \mathsf{empty} \vee \Box\, s_T(x) \wedge \neg\mathsf{empty} && (ii)\\
&\equiv s_T(x) \wedge \mathsf{empty} \vee \Box\, s_T(x) \wedge \neg\mathsf{empty} && (iii)\\
&\equiv s_T(x) \wedge \mathsf{empty} \vee \Box\, s_T(x) \wedge \mathsf{more} && (iv)\\
&\equiv s_T(x) \wedge \mathsf{empty} \vee s_T(x) \wedge \bigcirc\Box\, s_T(x) && (v)
\end{aligned}$$

4.2 MSVL Interpreter

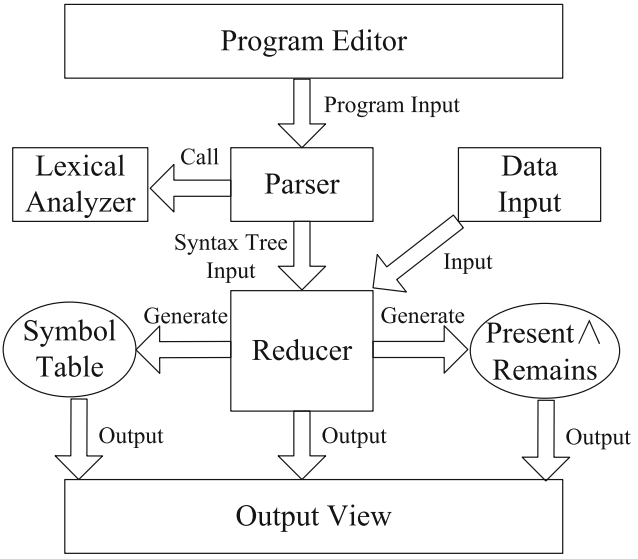
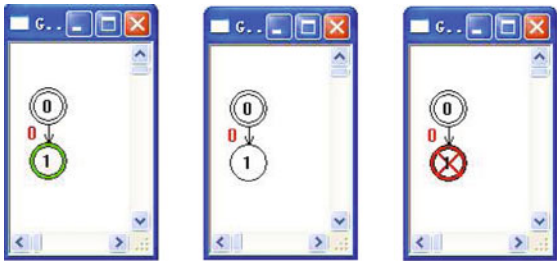


Fig. 3. Interpreter structure



(a) (b) (c)

Fig. 4. Three types of nodes. (a)modeling: a path. (b)verification: a satisfiable path. (c)verification: an unsatisfiable path.

5 An Application

```
frame(node1, node2, node3, p, q, r, head, tail) and (
    int[2] node1, node2, node3;
    pointer p,q,r;
    node1[0] = 10 and node1[1] := -1;
    node2[0] = node1[0] + 10 and node2[1] := -1;
    node3[0] = (int)30.0 and node3[1] := -1;
    node2[1] := &node3;     node1[1] := &node2;
    q :=& node1;    head := *q[0];
    while(q != -1){ tail := *q[0];      q := *q[1]  };
    p :=& node1;    q := -1;
```

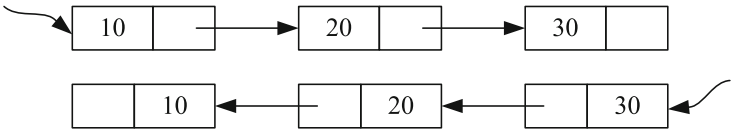
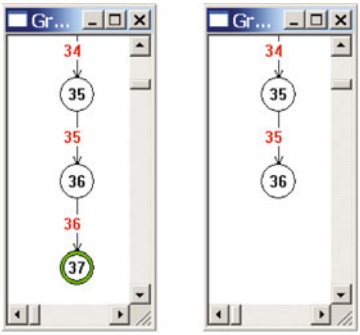


Fig. 5. In-place reversal of an integer list

```
    while(p != -1){ r := *p[1]; *p[1] := q; q := p; p := r   };
    head := *q[0];
    while(q != -1){ tail := *q[0];      q := *q[1]  }
)
```

As we [illegible] prop1 is head = 10 [illegible] prop2 is tail = 30 [illegible] prop3 [illegible] prop4 [illegible] $\Diamond(prop_3 \wedge prop_4) \wedge \Box(\mathsf{empty} \rightarrow prop_3 \wedge prop_4)$

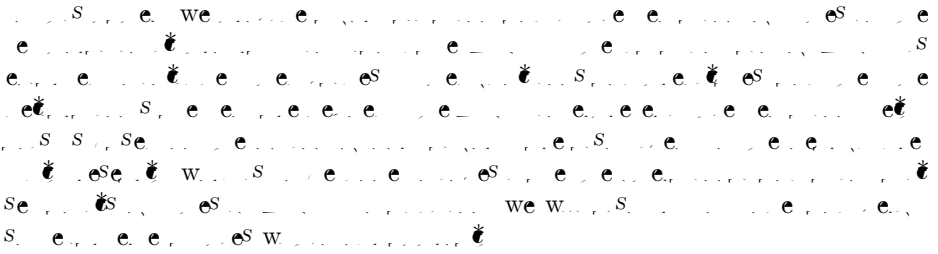
```
</
    define prop1: head = 10;    define prop2: tail = 30;
    define prop3: head = 30;    define prop4: tail = 10;
    som(prop1 and prop2) and always(empty -> prop3 and prop4)
/>
```



(a) (b)

Fig. 6. Program execution. (a) modeling result. (b) verification result.

6 Conclusions



References

1. Allen Emerson, E.: Temporal and modal logic. In: Handbook of Theoretical Computer Science, pp. 995–1072. Elsevier, Amsterdam (1995)
2. Clarke, E.M., Grumberg, O., Peled, D.: Model Checking. MIT Press, Cambridge (1999)
3. Baier, C., Katoen, J.-P.: Principles of Model Checking. MIT Press, Cambridge (2008)
4. Duan, Z., Maciej, K.: A framed temporal logic programming language. J. Comput. Sci. Technol. **19**, 341–351 (2004)
5. Duan, Z., Tian, C.: A unified model checking approach with projection temporal logic. In: Liu, S., Maibaum, T., Araki, K. (eds.) ICFEM 2008. LNCS, vol. 5256, pp. 167–186. Springer, Heidelberg (2008)
6. Tian, C., Duan, Z.: Expressiveness of propositional projection temporal logic with star. Theor. Comput. Sci. **412**, 1729–1744 (2011)
7. Cau, A., Moszkowski, B., Zedan, H.: Itl and tempura home page on the web. http://www.cse.dmu.ac.uk/STRL/ITL/ (2013)
8. Tang, Z.: Temporal Logic Program Designing and Engineering. Science Press, Beijing (1999)
9. Lamport, L.: The TLA Home Page. http://research.microsoft.com/en-us/um/people/lamport/tla/tla.html (2013)
10. Fisher, M.: MetateM: the story so far. In: Bordini, R.H., Dastani, M.M., Dix, J., El Fallah Seghrouchni, A. (eds.) PROMAS 2005. LNCS (LNAI), vol. 3862, pp. 3–22. Springer, Heidelberg (2006)
11. Zhou, S., Zedan, H., Cau, A.: Run-time analysis of time-critical systems. J. Syst. Archit. **51**, 331–345 (2005)
12. Duan, Z., Yang, X., Koutny, M.: Framed temporal logic programming. Sci. Comput. Program. **70**, 31–61 (2008)
13. Duan, Z.: Temporal Logic and Temporal Logic Programming. Science Press, Beijing (2006)
14. Yang, X., Duan, Z.: Operational semantics of framed tempura. J. Logic Algebraic Program. **78**, 22–51 (2008)
15. Duan, Z., Wang, X.: Implementing pointer in temporal logic programming languages. In: Proceedings of SBMF 2006, pp. 171–184 (2006)