# THE COMPUTABLE $\Sigma$ -SPECIFICATIONS OF HIERARCHICAL KRIPKE WORLDS

*VALENTINA GLUSHKOVA*

The modal logical specification is proposed which makes possible to construct the computable Kripke worlds. The theory $Th$ includes the specific $\Delta_0$-formulas [1], that may include the unary $\Box$ and the binary $\Diamond$ connectives of some similarity type [2]. Let us consider $\Delta_0 T$-formulas [3] of the form

$$(\forall x_1 \dot{\in} t_1) \dots (\forall x_m \dot{\in} t_m)\ (y_1 \prec z_1) \dots (y_p \prec z_p)(\varphi(\bar{x}, \bar{t}) \to \psi(\bar{x}, \bar{t})),$$

$m \geq 1$, $p \geq 0$, $\bar{x}$, $\bar{t}-$ the sequences of the corresponding variables; $y_j$, $z_j \in (\bar{x}, \bar{y})$, $1 \leq j \leq m$.

The formula prefix has the hierarchy structure with the root $t_1$ correlated with the $CF-$grammar $G = (I, P)$; $I-$ the the alphabet, $P-$ the set of the rules. The tree is represented by the list; $\dot{\in}$ denotes the membership relation for list elements or its reflexive transitive closure, $\prec$ – is the relation "more left". The relation $\dot{\in}$ corresponds to hierarchy subordination on the tree nodes . The formula $\varphi$ ($\psi$) is the conjunctions of atomic formulas $p$, $\tau_1 = \tau_2$ ($f = \tau$) or their negations, where $p-$ predicate, $f-$ function, $\tau, \tau_1, \tau_2-$ the terms of the many-sorted signature $\sigma =< I \cup \{list\}, C, F, R >$. Here $I$ is the set of sorts, *list* the sort of lists from $CF-$set formed from constants ($C$) according to $P$; $F, R-$ the sets of functions and predicates respectively. Right part atomic formulas (or its negations) are able to contain a single connective $\Diamond$. Left part (negated) atomic formulas might have the operator $\Box$.

Let $\Im =< W, Rp >$ be a modal Kripke frame, $W-$ a nonempty set of worlds, $Rp-$ partial order on $W$. The worlds $w_i \in W$ (inductive computable models of classic logic) are constructed on the base of the "modus ponens" deduction rule and the generalization rule. The true formula $\varphi(\bar{c}) \to \Diamond(p(\bar{c}), q(\bar{c}))$ (in world $w_i$) leads to the formation of two new incomparable worlds: $w_{i1}$, $w_{i2}$. Formally the world $w$ is the values tables for the functions and the predicates (or its negations) in the form: $p(\bar{c})$ $(\neg p(\bar{c}))$, $f(\bar{c}) = c_i$, $\bar{c} \in C^*$. The logical consequences of the theory are calculated simultaneously with the construction of the derivation tree of grammar $G$. The constants $\bar{c}$ substituted to deduction rules are terminal symbols of this tree.

The arithmetic operations exploited in the theory are interpreted as embedded ones. The existence of the computable model is the key point for the given formulas class. Keeping the conception of the weak equality one can construct the inductive model from signature constants for those theories, that possess confluentness and Noetherian properties. Restricting the variables dependence it is possible to select the polynomially realized $\Delta_0 T-$formulas with respect to the amount of tree nodes. The $\Delta_0 T$-formulas may be applied for describing behaviour of compound technical systems with respect to continuous real time.

As example one can describe lift simplified behaviour. The grammar $G_l$ specifies the lift actions sequence. Assume that lift has the single switch $Sw(t)$ depending explicitly on time; $Sw(t_0)$ is equal to the tuple containing numbers of involved floors; $t_0$ is time initial moment. The values from $Sw(t)$ are only removed but not added with time variation. Nonterminal symbols are the predicates of theory $Th_l$ describing the lift functioning. Symbol $St$ generates the couples $< m, time >$ where m is floor number. The time is assigned by "continuous" segment $< t_1, t_2 >$ or momentary time $t$. Nonterminal $Act$ denotes the actions: "up movement" ($MU \subseteq St \times St$) or "down movement" ($MD \subseteq St \times St$), "door opening" ($Dop \subseteq St$), "door closing" ($Dcl$), "lift standing" ($S$). Symbols $BrCd$, $BrOd$ designate two types of lift breakdowns: "the door is not closed", "the door is not opened"; $N-$ the floors quantity. Consider the grammar $G_l$ rules:

$$
\begin{aligned}
&1.LL \to \{Act\}^* \\
&2.Act \to MU(St, St) \mid MD(St, St) \mid S(St) \mid Dop(St) \mid Dcl(St) \mid \\
&BrCd(St) \mid BrOd(St) \\
&3.St \to Loc\, T \\
&4.Loc \to 1 \mid 2 \mid \ldots \mid N \\
&5.T \to t \mid (t, t) \mid [t, t) \mid (t, t] \mid [t, t]
\end{aligned}
$$

The theory $Th_l$ formalizes the knowledge on the simulated system behaviour. The predicate $Goal \subset St \times Loc$, the functions $h(< x_1, ..., x_n >) = x_1$, $e(< x_1, ..., x_n >) = x_n$ (regardless of the segment type) are used in the theory. All variables $st$ are connected to bounded quantifier $\forall st \in Act$; $m-$ $\forall m \in [1 - -N]$; $n = st[1]$; $t = st[2]$; $n_1 = st_1[1]$; $t_1 = st_1[2]$; $t_l = |m - n| l \diagup v$. Some signature constants are given below: $\delta-$ the time of door closing (opening); $l-$ the nearest floors distance; $v-$ the lift speed. The conjunction is represented by ",". The sequence of grammar productions is written to the left of the formula, where $sp = [3, 4, 5]$. These productions utilize the constants from $\bar{c}$ (as terminal symbols) that make right part of formula (1) true.

Let us represent the theory $Th_l$:

$$
\begin{aligned}
&1.Dcl(st), \neg Br(st) \to Goal(< n, e(t) >, m), \text{ where m=h(Sw(t))} \\
&2.Goal(st, m), m = n \to \Diamond(Dop(< m, [t, t + \delta) >), \neg Dop(< m, [t, t + \delta) >), \\
&3.Goal(st, m), m = n, \Box Dop(st) \to Sw(t + \delta) = Sw(t - \delta)/1 \\
&4.Goal(st, m), m = n, \Box\neg Dop(st) \to BrOp(st); [2.7, sp] \\
&5.\Box Dop(st) \to \Diamond(Dcl(< n, [e(t), e(t) + \delta) >), \neg Dcl(< n, [e(t), e(t) + \delta) >); [2.4, sp] \\
&6.\Box Dop(st), \Box\neg Dcl(st) \to BrCd(st); [2.6, sp] \\
&7.Goal(st, m), n < m \to MU(st, < m, [t + t_l) >), \\
&\Diamond(Dop(m, [t + t_1, t + t_1 + \delta)), \neg Dop(m, [t + t_1, t + t_1 + \delta)); [2.1, sp, sp] \\
&8.MU(st, st_1), \Box Dop(n_1, [t_1, t_1 + \delta)) \to Sw(t_1 + \delta) = Sw(t - \delta)/1; [2.4, sp]
\end{aligned}
$$

$9. Goal(st, m), n > m \rightarrow MD(st, < m, [t + t_l) >),$
$\diamondsuit(Dop(m, [t + t_1, t + t_1 + \delta)), \neg Dop(m, [t + t_1, t + t_1 + \delta)); [2.2, sp, sp]$
$10. MD(st, st_1), \Box Dop(n_1, [t_1, t_1 + \delta)) \rightarrow Sw(t_1 + \delta) = Sw(t - \delta)/1; [2.4, sp]$
$11. MU(st, st_1), \Box \neg Dop(n_1, [t_1, t_1 + \delta)) \rightarrow BrOp(st); [2.7, sp]$
$12. MD(st, st_1), \Box \neg Dop(n_1, [t_1, t_1 + \delta)) \rightarrow BrOp(st); [2.7, sp].$

Let $st_0 =< 5, [t_0, t_0 + \delta) >$ is the initial state and the predicates $Dcl(st_0)$, $\neg Br(st_0)$ are realized, $Sw(t_0) =< 7, 10, 10, 1 >$. If the lift is not broken then the action tree (world $w_{16}$) has the frontier: $MU\ (< 5, t_0+\delta >\ , < 7, t_0 + \delta + 2\nu >)\ Dop(< 7, [t_0+\delta+2\nu, t_0+2\delta+2\nu) >)$ $Dcl\ (< 7, [t_0 + 2\delta + 2\nu, t_0 + 3\delta + 2\nu) >)\ MU(< 7, t_0 + 3\delta + 2\nu >, < 10, t_0 + 3\delta + 5\nu >)$ $Dop(< 10, [t_0 + 3\delta + 5\nu, t_0 + 4\delta + 5\nu) >)\ Dcl\ (< 10, [t_0 + 4\delta + 5\nu, t_0 + 5\delta + 5\nu) >)$ $Dop\ (< 10, [t_0 + 5\delta + 5\nu, t_0 + 6\delta + 5\nu) >)\ Dcl\ (< 10\ , [t_0 + 6\delta + 5\nu, t_0 + 7\delta + 5\nu) >)$ $MD(< 10, t_0 + 7\delta + 5\nu >, < 1, t_0 + 7\delta + 14\nu >)\ Dop(< 1, [t_0 + 7\delta + 14\nu, t_0 + 8\delta + 14\nu) >)$ $Dcl(< 1, [t_0 + 8\delta + 14\nu, t_0 + 9\delta + 14\nu) >)$; $Sw(t_0 + 8\delta + 14\nu) =<>$.

For the frame different constraints may be stated expressed by arbitrary $\Delta_0 T-$formulas. In particular $BrCd(st) \rightarrow S(st)$, $BrOd(st) \rightarrow S(st)$ mean that the lift wreck implies its stopping. To prove these formulas it is necessary to specify the predicate $S(st)$. $\Delta_0 T-$formulas are polynomially realized and polynomial degree depends on grammar kind [4].

As time is explicit variable, the functions (predicates) may be continuosly continued. Suppose (in world $w_i$) $Sw(t_0) =< n_1, n_2, ..., n_k >$, $Sw(t_1) =< n_2, ..., n_k >$, $t_0, t_1 \in C$, $\neg \exists t \in [t_0, t_1)(Sw(t) \neq Sw(t_0))$, then the statement $\forall t \in [t_0, t_1)(Sw(t) = Sw(t_0))$ is set as declarative knowledge. It simplifies the specification problem of the constraints and raises the expressive capabilites of modeling language.

REFERENCES

1. S.S. Goncharov and D.I. Sviridenko. Theoretical aspects of Σ- programming. **Mathematical Methods of Specification and Synthesis of Software Systems' 85**. Proceed. of the Internat.Spring School. Springer-Verlag, April, 1985. pp. 169-179.

2. P. Blackburn,M.de Rijke,Y.Venema. Modal Logic.Cambridge Tracts in Theoretical Computer Science, 53, 2004.

3. V.N.Glushkova. Logical specifications as productions for transformation of program graphs. **The Bulletin of Symbolic Logic**, V.6, N.1, March 2000, pp. 133-134.

4. V. Glushkova. The execution complexity of logical formulas with restricted quantifiers based on CF-grammars. http://www.univ-paris12.fr/lacl/LCCS2001/accepted.html