УДК 681.3

K.V. Shahbazyan, academician Yu.H. Shoukourian

Logically Definable Languages of Computations in one Class of Flow Event Structures[1]

(Submitted 5/III 2002)

1. Introduction. Our paper is concerned with one model for concurrency, sometimes qualified as "true concurrency" model, because it takes of events and causality as fundamental [1,2,3]. Our goal is to investigate languages of infinite concurrent processes in one class of flow event structures - infinite homogenous flow event structures.

In this paper we are interested in two known formalisms of describing the behaviour of flow event structures: configurations (i.e., sets of partial ordered events), and proving sequences (i.e., sequences of events).

First, we prove that the language of infinite configurations of homogenous flow event structure is ω-regular. There exists an algorithm that constructs a Büchi [4] automaton representing ω-language of configurations for any homogenous flow event structure. Thus the language of infinite configurations may be represented in monadic second-order logic.

Then we investigate languages of proving sequences in homogenous flow event structures. These languages can be represented by a class of sentences of Presburger logic complemented with a finite set of monadic predicates. There exists algorithm recognizing the emptiness of these languages. There exists an algorithm that for each sentence of mentioned class constructs a homogenous flow event structure representing the language defined by the sentence.

2. Languages of infinite configurations. In this section, we give basic definitions regarding flow event structures (FES) and then study languages of configurations.

Definition 1 (*flow-event structure*)[3]. A flow event structure (FES) is S = (E, #, $\prec$), where

- E is a denumerable set of events,
- $\# \subseteq E \times E$ is a symmetric *conflict* relation,
- $\prec \subseteq E \times E$ is an irreflexive *flow* relation.◊

Definition 2 (*configuration*)[3]. A configuration of flow event structure S = (E, #, $\prec$) is a subset $X \subseteq E$ such that

- X is conflict-free,
- X does not contain a causality cycle, i.e., the relation $\leq_X \underset{def}{=} \prec_X^*$ is an ordering ($\prec^*$ stands for the transitive closure),
- For all $e \in X$ the set $\{ e' | e' \in X \wedge e' \leq_X e\}$ is finite,
- Each $e \in X$ is *saturated* in X, i.e., if $e' \prec e$ and $e' \notin X$ then there exists $e'' \in X$ such that $e'' \prec e$ and $e' \# e''$.◊

We present below the definition of a class of infinite flow event structures. We call these FES - homogenous FES (HFES). This class will be the object of our study.

Let S = (E, #, $\prec$) be an arbitrary finite FES. Consider infinite sequence of FES's:

$$S_1 = (E_1, \#_1, \prec_1),\ S_2 = (E_2, \#_2, \prec_2), \ldots, \text{ where for } i = 1, 2, \ldots \qquad (1)$$

$$E_i = \{e_i \mid e \in E,\},$$
$$\#_i = \{(e_i, e_i') \mid (e, e') \in \#\},$$
$$\prec_i = \{(e_i, e_i') \mid (e, e') \in \prec\},$$

that is each S_i is isomorfic to S.

We assume that events of these FES are connected by conflict and causality relations in a regular way. More precisely, let L, R, P be three finite sets of binary relations on E :

$$L = \{L^1, \cdots, L^\rho\},\ L^l \subseteq E \times E,$$
$$R = \{R^1, \cdots, R^\rho\},\ R^l \subseteq E \times E,$$
$$P = \{P^1, \cdots, P^\rho\},\ P^l \subseteq E \times E,$$

where $0 < l \leq \rho$.

By means of these relations we construct three infinite relation′s families that connect the events from different FES′s of sequence (1). The construction of these families is as follows.

Set for $L^l \in L$ and i = 1, 2, …

$$L_i^l = \begin{cases} \{(e_{i-l}, e_i') \mid (e, e') \in L^l\} \text{ if } l < i, \\ \varnothing \text{ if } l \geq i. \end{cases}$$

Each relation L_i^l will play the role of causal relation between the events of S_i and S_{i-l}, i.e., if $(e_{i-l}, e_i') \in L_i^l$ then the event $e_{i-l} \in E_{i-l}$ will be one of the causes of the event $e_i' \in E_i$.

Set for $R^l \in R$ and i = 1, 2, …

$$R_i^l = \{(e_{i+l}, e_i') \mid (e, e') \in R^l\}.$$

Each relation R_i^l will play the role of causal relation between the events of S_i and S_{i+l}, i.e., if $(e_{i+l}, e_i') \in R_i^l$ then the event $e_{i+l} \in E_{i+l}$ is one of the causes of the event $e_i' \in E_i$.

Set for $P^l \in P$ and i = 1, 2, …

$$P_i^l = \begin{cases} \{(e_{i-l}, e_i'), (e_i', e_{i-l}) \mid (e, e') \in P^l\} \text{ if } l < i, \\ \varnothing \text{ if } l \geq i. \end{cases}$$

Each relation P_i^l will play the role of conflict relation between the events of S_i and S_{i-l}, i.e., if $(e_{i-l}, e_i') \in P_i^l$ then the events $e_{i-l} \in E_{i-l}$ and $e_i' \in E_i$ are in conflict.

Definition 3 (*homogenous flow–event structure*). An infinite FES $S_\infty = (E_\infty, \#_\infty, \prec_\infty)$ is called homogenous flow-event structure (HFES) if it is generated by the quadruple (S, L, R, P), where $S = (E, \#, \prec)$ is an arbitrary finite FES, L, R, P are finite sets of binary relations over E, and

$$E_\infty = \bigcup_{i=1}^{\infty} E_i,$$
$$\prec_\infty = \bigcup_{i=1}^{\infty} \left(\prec_i \cup \bigcup_{l=1}^{\rho} L_i^l \cup \bigcup_{l=1}^{\rho} R_i^l \right),$$
$$\#_\infty = \bigcup_{i=1}^{\infty} \left(\#_i \cup \bigcup_{l=1}^{\rho} P_i^l \right). \Diamond$$

Notations to be used throughout the paper are as follows.

Let $Q \subseteq E$ be an arbitrary subset of E. We denote by Q_i the corresponding subset of E_i, i.e., $Q_i = \{e_i | e \in Q\}$.

Further we shall be concerned about languages of configurations of HFES. Let a quadruple ($S = (E, \#, \prec)$, L, R, P) be given and HFES $S_\infty = (E_\infty, \#_\infty, \prec_\infty)$ is generated by the quadruple.

We consider the alphabet 2^E and words over 2^E, i.e., finite sequences of sets

$$Q^1, Q^2, \ldots, Q^n, \tag{2}$$

where $Q^i \subseteq E$ and the corresponding sequence (3):

$$Q_1^1, Q_2^2, \ldots, Q_n^n, \tag{3}$$

where $Q_i^i = \{e_i | e \in Q^i\} \subseteq E_i$. Remark that $\bigcup_{i=1}^{n} Q_i^i \subseteq E_\infty$ and each finite configuration of S_∞ can be uniquely represented as $\bigcup_{i=1}^{n} Q_i^i$. In this case, we say that the word $Q^1\, Q^2 \ldots Q^n$ defines a configuration $\bigcup_{i=1}^{n} Q_i^i$ of S_∞.

We say that a set $\mathbf{C}(S_\infty)$ of finite words over the alphabet 2^E is a language of finite configurations of HFES S_∞ if $\mathbf{C}(S_\infty) = \{Q^1\, Q^2 \ldots Q^n \mid Q^1\, Q^2 \ldots Q^n$ defines the configuration of $S_\infty\}$.

Theorem 1. *For each* HFES S_∞ *there exists a finite state automaton over alphabet* 2^E *accepting a word* $Q^1 \ldots Q^n$ *iff it defines a configuration of* S_∞.

We denote by $\mathbf{C}^\omega(S_\infty)$ the ω-language Büchi-recognizable by automaton $\mathbf{B}(S_\infty) = (\Sigma, \Delta, F, \langle\varnothing, \varnothing\rangle)$ over 2^E.

Evidently, ω-word over 2^E belongs to the ω-language $\mathbf{C}^\omega(S_\infty)$ iff for each $n > 0$ exists a $n_0 > n$ such that $Q^1 \ldots Q^{n_0}$ defines a configuration of S_∞. This gives us the Theorem 2.

Theorem 2. *The language* $\mathbf{C}^\omega(S_\infty)$ *of all infinite configurations of any HFES* S_∞ *is* ω*-regular. There exists an algorithm that constructs for any* S_∞ *a Büchi automaton* $\mathbf{B}(S_\infty)$ *representing the* ω*-language* $\mathbf{C}^\omega(S_\infty)$ *of configurations. So the* ω*-language* $\mathbf{C}^\omega(S_\infty)$ *can be represented in monadic second-order logic.*

3. Languages of proving sequences. In this section, we study the languages of proving sequences, i.e., linearizations of configurations.

Definition 4 (*proving sequence* for S_∞)[3]. Given S_∞, a proving sequence in S_∞ is a (finite or infinite) sequence $\alpha = \alpha(1)\alpha(2)\ldots$ of distinct nonconflicting events $\alpha(i) \in E_\infty$ (i.e., $i \neq j \Rightarrow \alpha(i) \neq \alpha(j)$ and $\neg(\alpha(i) \#_\infty \alpha(j))$ for all i, j) satisfying

$$\forall n\, \forall e : e \prec_\infty \alpha(n) \Rightarrow \exists m < n : (\alpha(m) = e) \quad \vee \quad (e \#_\infty \alpha(m)) \wedge (\alpha(m) \prec_\infty \alpha(n)). \lozenge$$

We denote by $\mathbf{P}(S_\infty)$ a language of all infinite proving sequences in S_∞. We shall say that the language $\mathbf{P}(S_\infty)$ is represented by S_∞. It is known that $\mathbf{P}(S_\infty)$ can not be accepted by finite automaton [6].

We present below a logic **L** and a class of well-defined sentences of **L** which describes the languages $\mathbf{P}(S_\infty)$ of proving sequences for arbitrary HFES S_∞.

Given HFES S_∞, consider the alphabet $E_\infty = \{e_j \mid e \in E, j > 0\}$. We shall represent an ω-word $\alpha = \alpha(0)\alpha(1)\ldots$, where $\alpha(i) \in E_\infty$ by the structure $\alpha = (\omega, 0, +1, <, \{q_e\}_{e\in E}, \psi)$. Here $(\omega, 0, +1, <)$ is the structure of natural numbers with zero, successor function and the natural ordering and q_e is monadic predicate such as

$q_e = \mathbf{T}$ if $i \in \omega$ and $\alpha(i) = e_j$ for some j, and

$\psi(i)$ is one-place function defined on ω, i.e., $\psi(i) = j$ if $\alpha(i) = e_j$ for some $e \in E$, i.e., $\psi(i)$ is the *index* of event which stands in i-th position in $\underline{\alpha}$.

We allow also variables i, j for natural numbers, i.e., for positions of ω-words and for indexes of events. Terms are constructed from the constant 0, variables i, j by application of "+1" (successor function) and ψ (index function). Atomic formulas are of the form $q_e(i)$, $t = t'$, $t < t'$, $\psi(i) = t$, where t, t′ are terms. **L**-formulas are constructed from atomic formulas using the connectives ∨, ∧, ¬, → and quantifiers ∃, ∀ acting on variables. Formulas without free variables are called *sentences*.

Consider two predicate families $\{q_e(i, j)\}_{e\in E}$ and $\{u_e(i, j)\}_{e\in E}$. They can be expressed by fundamental notions of **L** as follows.

$$
\begin{aligned}
q_e(i,j) &\underset{\text{def}}{=} q_e(i) \wedge (\psi(i) = j),\\
u_e(i,j) &\underset{\text{def}}{=} \exists\, i' < i \wedge q_e(i') \wedge (\psi(i') = j).
\end{aligned}
\tag{4}
$$

The predicate $q_e(i, j)$ denotes the statement "$\alpha(i) = e_j$". The predicate $u_e(i, j)$ denotes the statement "there exists $i' < i$ such that $\alpha(i') = e_j$".

Now we consider one class of propositional formulas. Each formula of this class generates a sentence of **L** that determines an ω-language representable in a HFES.

Let us consider a formula $\Theta(i, j)$ without quantification constructed from elementary statements of the set

$$\{q_e(i, j) \mid e \in E\} \cup \{u_e(i, j + t) \mid e \in E,\ t = 1, 2, \ldots\}$$

using the connectives ∨, ∧, ¬, → and let $\Theta^*(i, j)$ be its disjunctive normal form.

Formula $\Theta(i, j)$ is *well-defined* if its disjunctive normal form $\Theta^*(i, j)$ satisfies the following conditions:

- For each summand of formula $\Theta^*(i, j)$ there exists unique $e \in E$ such that elementary statement $q_e(i, j)$ is contained in the summand without negation. A summand of $\Theta^*(i, j)$ is called a e-*summand* if it contains $q_e(i, j)$.
- If $\Theta^*(i, j)$ contains $u_e(i, j + t)$ or $\neg u_e(i, j + t)$ then $\Theta^*(i, j)$ contains e-summand too.
- If a e-summand contains $\neg u_{e'}(i, j + t)$ then each e-summand contains $\neg u_{e'}(i, j + t)$ and e′-summand contains $\neg u_e(i, j - t)$.
- If $\Theta^*(i, j)$ contains two e-summands θ_1 and θ_2 then for each $u_{e'}(i, j + t) \in \theta_1 \setminus \theta_2$ there is a $u_{e''}(i, j + t') \in \theta_2 \setminus \theta_1$ such that e″ -summand contains $\neg u_{e'}(i, j - t' + t)$, and each e′-summand contains $\neg u_{e''}(i, j + t' - t)$.

Starting from arbitrary well-defined formula $\Theta(i, j)$, we construct now a correspondent sentence $\Theta_{\mathbf{L}}$ of logic **L**.

Let a well-defined formula $\Theta(i, j)$ be given and $\rho = \max\{|t| \text{ such that } u(i, j + t) \text{ enters } \Theta(i, j)\}$. Consider the formula $\Theta'(i) = \Theta'(i, 1) \vee \Theta'(i, 2) \vee \ldots \vee \Theta'(i, \rho)$, where $\Theta'(i, k)$ is obtained from $\Theta^*(i, k)$ by substitution of value **T** instead of all occurences of $u(i, k - t)$ and $\neg u(i, k - t)$ for $k \le t$. Then we get formulas $\Theta''(i)$ and $\Theta''(i, j)$ from the formulas $\Theta'(i)$ and $\Theta(i, j)$ by replacing all occurences of q(i, j), u(i, j + t) by correspondent expressions (4). Finally, we construct the following sentence $\Theta_{\mathbf{L}}$ of logic **L** :

$$\Theta_{\mathbf{L}} \underset{\mathrm{def}}{=} \forall i \left(\Theta''(i) \vee \underset{j>p}{\exists} \Theta''(i, j) \right)$$

Thus to each well-defined formula $\Theta(i, j)$ we associate a sentence $\Theta_{\mathbf{L}}$ of logic **L**. We call it *sentence of* **L** *generated by* $\Theta(i, j)$.

Theorem 3. *Suppose* $\mathcal{L}$ *is an* ω*-language over infinite alphabeth* $E_\infty = \{e_i \mid e \in E, i \in N\}$, *where* E *is a finite set. Then the following conditions are equivalent:*

1) $\mathcal{L}$ *is definable by a sentence* $\Theta_{\mathbf{L}}$ *generated by well-defined formula.*

2) $\mathcal{L} = \mathbf{P}(S_\infty)$ *for some* HFES S_∞ .

Corollary 1. *The emptiness problem of* ω*-language* $\mathbf{P}(S_\infty)$ *is decidable.*

Consider logic $\mathbf{L}_+$ which differs from **L** by the following items:

- A finite alphabet $E \cup \{|\}$ is considered instead of infinite E_∞ ;
- A predicate $q_|$ is added such that $q_|(i) = \mathbf{T}$ iff $\alpha(i) = |$ in the word $\alpha = \alpha(1)\alpha(2) \ldots$;
- The function ψ is excluded. It is replaced by operation of addition of arbitrary natural numbers.

Thus $\mathbf{L}_+$ is the Presburger logic with additional monadic predicates $\{q_e\}_{e \in E} \cup \{q_|\}$. Evidently, the function ψ can be computed in $\mathbf{L}_+$.

Theorem 4. *The* ω*-language* $\mathbf{P}(S_\infty)$ *of proving sequences for arbitrary* FES S_∞ *can be represented by a sentence of Presburger logic extended by a finite number of monadic predicates.*

Corollary 2. *The emptiness problem is decidable for each sentence* $\Theta_{\mathbf{L}_+}$ *generated by well-defined formula.*

Acknowledgment. Prof. I.D.Zaslavsky has given very helpful comments on a draft version of this paper.

Institute for Informatics and Automation Problems
National Academy of Sciences of Armenia

References

1. *Winskel G. and Nielsen M.* - In Handbook of Logic in Computer Science, Oxford Univ. Press. 1995. V 4.
2. *Winskel G.* - LNCS. 1988. V. 354.
3. *Boudol G.* - LNCS. 1990. V. 469.
4. *Büchi J.R.* - Z. math. logic und Grundlag. Math. 1960. V. 6. N1.
5. *Thomas W.* - In Handbook of Theoretical Computer Science, Elsevier Sci. Publ. B.V. 1990.
6. *Shakhbazyan K.V., Shoukourian Yu.H.* - Journal of Math. Sci. 1999. V. 101. N4.

Կ.Վ. Շահբազյան, ակադեմիկոս Յու.Հ. Շուքուրյան

Հոսքային պատահարային կառուցվածքների հաշվարկների լեզուների տրամաբանական ներկայացումը

Դիտարկվում են բախշված համակարգերի մոդելներից մեկի՝ համասեռ պատահարային կառուցվածքների, հաշվարկների լեզուները և նրանց տրամաբանական ներկայացումը:

Ապացուցվում է, որ անվերջ կոնֆիգուրացիաների լեզուները ներկայացվում են մեկտեղանի երկրորդ կարգի պրեդիկատների տրամաբանության մեջ:

Ապացուցող հաջորդականությունների լեզուները ներկայացվում են վերջավոր մեկտեղանի պրեդիկատներով ընդլայնված Պրեսբուրգերի տրամաբանությունում:

К.В. Шахбазян, академик Ю.Г. Шукурян

Представление языков вычислений в одном классе событийных структур в логических языках

Рассматриваются языки вычислений в одном классе моделей распределенных вычислений - однородных событийных структур - и их представимость в логических языках.

Языки бесконечных конфигураций представимы в логике одноместных предикатов второго порядка.

Языки доказывающих последовательностей представимы в логике Пресбургера, дополненной конечным множеством одноместных предикатов.