

Resource-based games

I A Sheremet¹

¹Russian Foundation for Basic Research, Leninskiy Prosp., 32a, Moscow, Russia, 119334

sheremet@rfbr.ru

Abstract. Article is dedicated to the multigrammatical modelling of games. Basic notions and definitions, concerning multisets and multiset grammars, are considered. So called resource-based games are introduced, and representation of their simplest class – antagonistic resource-based games – by filtering multiset grammars is considered in details. More sophisticated cooperative and coalitional resource-based games are proposed. Directions of further development of resource-based games by application of multigrammatical framework are discussed.

1. Introduction

Substantial background of any kind of conflict between sociotechnological systems (STS), every of which includes people and technological devices, is that both opposing sides possess some amounts of resources (called usually their “resource bases”, RB), and goals of both are to eliminate (destroy) or capture resources of the counterpart, spending for this purpose some adequate amounts of its own resources. So logics of any counterpart behavior at any moment is defined by current resources he possesses and possible actions he may perform, spending some part of his own RB for eliminating/capturing of some part of RB of the opponent. Thus conflict is implemented as a chain of sequentially made actions (mutual impacts) of opponents until it is possible. Problem of planning of behavior of opponents during conflict is to determine such sequence of actions (“strategy”), leading to the end of the conflict (i.e. final resource bases of the players), which meets goals of both or at least one of the opponents, who is declared winner. Such conflicts are called below resource-based (RBC).

Conflicts of any nature and scale are objects of the game theory (GT), which is one of the most valuable and usable parts of modern operations research, providing decision makers by clear, simply applied and effectively algorithmically implemented methodology of strict mathematical representation of logics of action of opposing subjects [1, 2, 3].

However, despite widespread and commonly understood essence of the described resource-based conflicts, GT state of the art is such, that available mathematical tools, used nowadays by GT developers, are not exactly suitable for completely adequate mathematical formalization of such conflicts, because application of these tools demands explicit representation of sets of possible implementations of conflicts in a form of trees, matrices or automata. By this, dimensionalities of practically interest representations of RBC are so large, and time, necessary for their construction, so long, that in most cases it is very hard or even impossible to apply known GT approaches to the adequate modelling of real RBC.

Alternative approach, which consideration in the context of RBC may have evident reasons, have raised from the artificial intelligence area. It differs from GT by basic representation of the potential

activities of the opposing sides and possible ways of application of their capabilities during conflict. This approach is called “multi-agent” [4, 5, 6], following that every STS is represented as a set of active entities, called agents, operating independently in accordance with their internal logics. Interactions between agents are implemented in full accordance with set of a priori established mutual agreements, providing resolution of current situations and constructing consequences of the aforementioned interactions. Agents form multi-agent system (MAS) [7, 8, 9].

In comparison with modern GT models, MAS do not demand explicit representation of action of the opposing sides during conflict. It is sufficient to define implicitly logics of action of all agents, entering MAS, and to construct the initial state of this MAS, i.e. set of initial states of its agents. All further behavior of the modelled STS until some final state will be explicated by MAS itself [13, 14, 15, 16].

Being in some sense more advanced approach in comparison with classical GT frameworks, multi-agent technologies (MAT) usually do not operate notions of strict optimality and optimal strategies in the sense of game theory, and in most cases implement “what-if” regimes, providing heuristic-based assessment of very limited number of all possible consequences of decisions made by the opposing sides.

In order to integrate best features of both GT- and MAT-centered technologies of modelling resource-based conflicts, a new approach to representation and solution of such conflicts is proposed in this article. Its basic notion is “resource-based game” (RBG), and its mathematical background is theory of recursive multisets, which basic mathematical toolkit are multiset grammars (MG) [17, 18, 19, 20, 21, 22]. This toolkit is result of deep integration in one multiset-based formalism key advantages of classical operations research (strict mathematical definition and correct algorithmic availability of provably optimal solutions) and modern knowledge engineering (generality, simplicity and flexibility of problems representation). This article is introduction to the theory of RBG, which main nowadays application is economical combinatorics.

Section 2 contains main definitions and notions of the theory of recursive multisets, necessary for further considerations. Resource-based antagonistic games are introduced in Section 3, while resource-based cooperative and coalitional games – in Section 4. Directions of further research in the RBG area are discussed in the conclusion.

2. Multisets, operations on multisets, and multiset grammars

A background of classical theory of sets is notion of set as an unordered collection of mutually distinguishable elements. Theory of multisets is based on the assumption, that aforementioned collection may contain identical (indistinguishable) elements:

$$v = \left\{ \underbrace{a_1, \dots, a_1}_{n_1 \text{ times}}, \dots, \underbrace{a_i, \dots, a_i}_{n_i \text{ times}}, \dots, \underbrace{a_m, \dots, a_m}_{n_m \text{ times}} \right\}, \quad (1)$$

and this is recorded as

$$v = \{n_1 \cdot a_1, \dots, n_m \cdot a_m\}, \quad (2)$$

where v is called multiset, a_i – objects, n_i – multiplicities of objects, and $n_i \cdot a_i$ – multiobjects (MO), $i = 1, \dots, m$. Number $|v| = m$, being quantity of MO in MS v , is called it’s dimensionality, while number

$$|v| = \sum_{i=1}^m n_i, \quad (3)$$

is called power of MS v . Following (2), multiset may be considered as a set of multiobjects, and, from the substantial point of view, multiset $\{1 \cdot a_1, \dots, 1 \cdot a_m\}$ and set $\{a_1, \dots, a_m\}$ mean one and the same collection. Set $\{a_1, \dots, a_m\}$ is called basis of MS v and is denoted $\beta(v)$. Empty multiset as well as empty set is denoted $\{\phi\}$.

Zero multiplicity of some object is identical to absence of this object in multiset, i.e.

$$\{n_1 \cdot a_1, \dots, n_m \cdot a_m, 0 \cdot a_{m+1}\} = \{n_1 \cdot a_1, \dots, n_m \cdot a_m\}. \quad (4)$$

Fact, that object a enters MS v , or, just the same, MS v includes object a , is denoted $a \in v$. Symbol “ \in ” is also used to denote, that MO $n \cdot a$ enters MS v (MS v includes MO $n \cdot a$): $n \cdot a \in v$. Structure of the left operand determines what kind of relation is denoted in every particular case. Similarly, symbol “ \notin ” is used to de-note, that object a (multiobject a) does not enter multiset v . Due to (4), $a \notin v$ and $0 \cdot a \in v$ are equivalent.

Two main relations on multisets – inclusion (“ \subseteq ”) and strict inclusion (“ \subset ”) – are defined as follows. MS v is included to MS v' , if

$$(\forall n \cdot a \in v)(\exists n' \cdot a \in v')n \leq n', \quad (5)$$

i.e. for every multiobject $n \cdot a$ entering MS v there exist multiobject $n' \cdot a$, which multiplicity n' is not less than n . There may be also $n' \cdot a' \in v'$ such that $a' \notin v$ (this does not contradict (5), because $0 \cdot a' \in v$ and $0 < n'$). If $v \subseteq v'$ and $v \neq v'$, then MS v is strictly included to MS v' , that is denoted $v \subset v'$. MS v , which is included to MS v' , is called submultiset of MS v' ; MS v , which is strictly included to MS v' , is called strict submultiset of MS v' .

Let us illustrate introduced notions by the following example, where objects will be represented by strings in brackets. This same notation will be used in all examples, having place in this chapter.

Example 1. Let

$$v = \{3 \cdot (\text{eur}), 5 \cdot (\text{usd}), 10 \cdot (\text{rur})\},$$

$$v' = \{6 \cdot (\text{eur}), 5 \cdot (\text{usd}), 15 \cdot (\text{rur})\}.$$

As seen, according to (5), $v \subseteq v'$ and $v \subset v'$. ■

As it is known from [23, 24], multiplicities may be not only non-negative integer, but also non-negative rational numbers. Three main operations on multisets are multiplication by a constant, addition, and subtraction, which are denoted by bold symbols $*$, $+$ and $-$ respectively. Semantics of these operations is defined by use of the well-known set-theoretical operations (join and intersection), as well as of the arithmetic operations:

$$n * \{n_1 \cdot a_1, \dots, n_m \cdot a_m\} = \{(n \times n_1) \cdot a_1, \dots, (n \times n_m) \cdot a_m\}, \quad (6)$$

where “ \times ” denotes multiplication of integer numbers;

$$v + v' = \bigcup_{\substack{a \in \beta(v) \cup \beta(v') \\ n \cdot a \in v \\ n' \cdot a \in v'}} \{(n + n') \cdot a\}, \quad (7)$$

$$v - v' = \bigcup_{n \cdot a \in v} \{(n - n') \cdot a\}. \quad (8)$$

$$\begin{aligned} n' \cdot a \in v' \\ n > n' \end{aligned}$$

Along with these operations, there will be used set-theoretical operations on multisets – join and intersection, – denoted respectively by bold symbols \cup and \cap , different from \cup and \cap :

$$\begin{aligned} v \cup v' = \bigcup_{\substack{\alpha \in \beta(v) \cup \beta(v') \\ n \cdot a \in v \\ n' \cdot a \in v}} \{\max\{n, n'\} \cdot a\}, \end{aligned} \quad (9)$$

$$\begin{aligned} v \cap v' = \bigcup_{\substack{\alpha \in \beta(v) \cup \beta(v') \\ n \cdot a \in v \\ n' \cdot a \in v}} \{\min\{n, n'\} \cdot a\} \end{aligned} \quad (10)$$

We have used the aforementioned equivalence between $a \notin v$ and $0 \cdot a \in v$ in (9)-(10).

Example 2. Let v and v' be as in the Example 1. Then

$$\begin{aligned} 3 * v &= \{9 \cdot (\text{eur}), 15 \cdot (\text{usd}), 30 \cdot (\text{eur})\}, \\ v + v' &= \{9 \cdot (\text{eur}), 10 \cdot (\text{usd}), 25 \cdot (\text{rur})\}, \\ v - v' &= \{\emptyset\}, \\ v' - v &= \{3 \cdot (\text{eur}), 5 \cdot (\text{rur})\}, \\ v \cup v' &= \{6 \cdot (\text{eur}), 5 \cdot (\text{usd}), 15 \cdot (\text{rur})\}, \\ v \cap v' &= \{3 \cdot (\text{eur}), 5 \cdot (\text{usd}), 10 \cdot (\text{rur})\} = v, \\ &\text{because } v \subset v'. \blacksquare \end{aligned}$$

All described operations are well known from theory of multisets [23, 24]; their common feature is that their operands are multisets. The following operation called “filtration” has set of multisets (SMS) as the first operand and set of conditions called “filter” as the second operand. Filtration is denoted as $V \downarrow F$, where V is filtrated SMS, “ \downarrow ” – symbol of operation, and F is filter.

Conditions entering filter F may be boundary and optimizing.

Boundary condition (BC) is written as $a\theta n$, where $\theta \in \{>, <, \geq, \leq, =\}$. Multiset v satisfies BC $a\theta n$, if $m \cdot a \in v$, and $m\theta n$ is true (as everywhere, $a \notin v$ is equivalent to $0 \cdot a \in v$).

Let $F_{\leq} = \{bc_1, \dots, bc_k\}$ be a filter, containing only boundary conditions.

Then

$$v \downarrow F_{\leq} = \bigcap_{i=1}^k (V \downarrow \{bc_i\}). \quad (11)$$

Example 3. Let $V = \{v_1, v_2, v_3\}$, where

$$\begin{aligned} v_1 &= \{3 \cdot (\text{eur}), 5 \cdot (\text{rur})\}, \\ v_2 &= \{7 \cdot (\text{usd}), 9 \cdot (\text{rur})\}, \\ v_3 &= \{4 \cdot (\text{eur}), 11 \cdot (\text{usd}), 15 \cdot (\text{rur})\}, \end{aligned}$$

and

$$F_{\leq} = \{(\text{usd}) > 3, (\text{rur}) \leq 10\}.$$

Then

$$V \downarrow F_{\leq} = (V \downarrow \{(\text{usd}) > 3\}) \cap (V \downarrow \{(\text{rur}) \leq 10\}) = \{v_2, v_3\} \cap \{v_1, v_2\} = \{v_2\}. \blacksquare$$

Optimizing condition (OC) is written $a = opt$, where $opt \in \{max, min\}$. Multiset $v \in V$ satisfies OC $a = max$, if $n \cdot a \in v$, and all $v' \in V - \{v\}$ satisfy boundary condition $a \leq n$. Similarly, multiset $v \in V$ satisfies OC $a = min$, if $n \cdot a \in v$, and all $v' \in V - \{v\}$ satisfy boundary condition $a \geq n$.

Let $F_{opt} = \{oc_1, \dots, oc_l\}$ be a filter, containing only optimizing conditions. Then, similarly to (6),

$$v \downarrow F_{opt} = \bigcap_{i=1}^l (v \downarrow \{oc_i\}). \quad (12)$$

Example 4. Let V be the same as in the Example 3, and

$$F_{opt} = \{(usd) = max, (eur) = min\}.$$

Then

$$v \downarrow F_{opt} = v \downarrow \{(usd) = max\} (v \downarrow \{(eur) = min\}) = \{v_3\} \cap \{v_1\} = \{\phi\}. \blacksquare$$

If filter F contains both boundary and optimizing conditions, i.e.

$$F = F_{\leq} \cup F_{opt}, \quad (13)$$

then

$$V \downarrow F = (V \downarrow F_{\leq}) \downarrow F_{opt}, \quad (14)$$

i.e. result of filtering set of multisets V by filter F is obtained by application of boundary subfilter F_{\leq} to V , and then resulting SMS is filtered by optimizing subfilter F_{opt} . This substantial interpretation of SMS filtering is a background for multiset representation of classical problems of operations research [17, 18]. On the other hand, it is clear, that language of filters is very close to query languages, providing access to the relational and relational-like databases [25, 26].

Introduced operations on multisets and sets of multisets make it possible to define syntax and semantics of family of multiset grammars. It's background is notion of multiset grammar as a couple $S = \langle v_0, R \rangle$, where v_0 is multiset, and R is a finite set of rules; v_0 is called kernel, and R is called scheme. The set of all objects used in kernel and scheme of MG S is denoted A_S . Rule $r \in R$ is a construction

$$v \rightarrow v', \quad (15)$$

where multisets v and v' are called left part and right part of the rule r respectively, $v \neq \{\phi\}$, and “ \rightarrow ” is divider.

Application of the rule r to the multiset \bar{v} is defined as follows. If $v \subseteq \bar{v}$, then the result of r application to \bar{v} is multiset

$$\bar{v}' = \bar{v} - v + v' \quad (16)$$

i.e., speaking informally, MS v , which is submultiset of MS \bar{v} , is replaced by MS v' . The result of application of rule r to multiset \bar{v} is denoted

$$\bar{v} \xrightarrow{r} \bar{v}' \quad (17)$$

and it is said, that MS \bar{v}' is generated from MS \bar{v} by application of rule r . If left part v is not submultiset of MS \bar{v} , result of application r to \bar{v} is assumed empty MS $\{\phi\}$.

Recursive definition of set of multisets, generated by application of multigrammar $S = \langle v_0, R \rangle$, is as follows:

$$V_{(0)} = \{v_0\}, \quad (18)$$

$$V_{(i+1)} = V_{(i)} \cup \left(\bigcup_{\bar{v} \in V_{(i)}} \bigcup_{r \in R} \{\bar{v}' \mid \bar{v} \xrightarrow{r} \bar{v}'\} \right) \quad (19)$$

$$V_S = V_{(\infty)}. \quad (20)$$

As seen, V_S includes all multisets, which may be generated from MS v_0 by sequential application of rules $r \in R$, and V_S is nothing but fixed point of the sequence $V_{(0)}, V_{(1)}, \dots, V_{(i)}, \dots$, so

$$V_S = \bigcup_{i=0}^{\infty} V_{(i)}, \quad (21)$$

and in general case V_S may be infinite.

If MS \bar{v}' may be generated from MS \bar{v} by application of scheme R , it is denoted

$$\bar{v} \xrightarrow{R} \bar{v}', \quad (22)$$

so, from this point of view,

$$V_S = \{\bar{v} \mid v_0 \xrightarrow{R} \bar{v}\}. \quad (23)$$

Multiset $\bar{v} \in V_S$ is called terminal multiset (TMS), if

$$(\forall r \in R) \bar{v} \not\xrightarrow{r} \{\phi\}, \quad (24)$$

i.e. no any rule $r \in R$ may be applied to \bar{v} . Set of terminal multisets (STMS) generated by application of multiset grammar S is denoted \bar{V}_S . Obviously,

$$\bar{V}_S \subseteq V_S. \quad (25)$$

Example 5. Let $S = \langle v_0, R \rangle$, where $v_0 = \{5 \cdot (\text{eur}), 7 \cdot (\text{usd}), 4 \cdot (\text{rur})\}$, and $R = \{r_1, r_2\}$, where r_1 is $\{3 \cdot (\text{eur})\} \rightarrow \{4 \cdot (\text{usd})\}$, and r_2 is $\{3 \cdot (\text{usd}), 2 \cdot (\text{rur})\} \rightarrow \{2 \cdot (\text{eur})\}$.

According to (18) – (19),

$$V_{(0)} = \{\{5 \cdot (\text{eur}), 7 \cdot (\text{usd}), 4 \cdot (\text{rur})\}\},$$

$$V_{(1)} = V_{(0)} \cup \{\{2 \cdot (\text{eur}), 11 \cdot (\text{usd}), 4 \cdot (\text{rur})\}, \{7 \cdot (\text{eur}), 4 \cdot (\text{usd}), 2 \cdot (\text{rur})\}\},$$

$$V_{(2)} = V_{(1)} \cup \{\{4 \cdot (\text{eur}), 8 \cdot (\text{usd}), 2 \cdot (\text{rur})\}, \{4 \cdot (\text{eur}), 8 \cdot (\text{usd}), 2 \cdot (\text{rur})\}, \{9 \cdot (\text{eur}), 1 \cdot (\text{usd})\}\}, \dots$$

As seen, this MG provides generation of all possible collections of euros, dollars, and rubles, which may be obtained from the initial collection v_0 by sequential currency exchanges, which parameters are fixed by rules r_1 and r_2 (3 euros may be exchanged to 4 dollars, while 3 dollars and 2 rubles may be exchanged to 2 euros). ■

Generalizing introduced notions, we may affirm, that multiset grammars have the same conceptual background, as classical string-operating grammars, proposed by N. Chomsky [27]. Despite primary elements, which are processed (multisets and strings), differ, the key logical construction of both formalisms – generation of new elements from already generated by replacement of their parts according to the applicable rules – is the same. Chomsky grammars are used for generation of sets of strings (“sentential forms”) until sentences of language, while multiset grammars provide generation of sets of multisets until terminal MS. Due to the fact, that multiobjects contain both numerical and symbolic components (multiplicities and object names), generation of multisets incorporates numerical computation, that provides a lot of new opportunities for simple formalizing and effective solution of various practical problems with combinatorial background. To implement such opportunities, so called filtering multiset grammars were proposed in [17, 18]. FMG are such generalization of MG, that integrate two basic concepts – generation of set of multisets and selection from it such MS, that satisfy some logical conditions, joined to filter.

Filtering multiset grammar is a triple $S = \langle v_0, R, F \rangle$, where v_0 and R are kernel and scheme, while F is a filter, including boundary and optimizing conditions, defining multisets, which would be selected from set of terminal MS, generated by MG $\langle v_0, R \rangle$, i.e.

$$V_S = V_{\langle v_0, R \rangle} \downarrow F. \quad (26)$$

Verbally, V_S is subset of $V_{\langle v_0, R \rangle}$, which includes only such elements of this set, that satisfy filter F .

Example 6. Consider $S = \langle v_0, R, F \rangle$, where v_0 and R are the same, as in the Example 5, and filter $F = \{(\text{eur}) > 7, (\text{rur}) = \min\}$, that means V_S would include collections, created by all possible chains of currency exchange, that contain more than 7 euros, and minimal sum of rubles. According to (9), (18) – (20) and (26), STMS, generated by application of FMG S , would contain, at least, multiset $\{9 \cdot (\text{eur}), 1 \cdot (\text{usd})\}$, because it includes multiobject $9 \cdot (\text{eur})$ satisfying boundary condition $(\text{eur}) > 7$, as well as $0 \cdot (\text{rur})$ with absolutely minimal multiplicity 0, thus satisfying optimizing condition $(\text{rur}) = \min$. (Remind, that $(\text{rur}) \notin v$ is equivalent to $0 \cdot (\text{rur}) \in v$). ■

As may be seen, generation of set of terminal multisets by application of filtering multiset grammar is nothing but knowledge- and goal- driven computation.

By this we finish brief introduction to the mathematical toolkit, which will be used below for representation of the proposed resource-based games.

3. Resource-based antagonistic games

We shall begin from the simplest kind of game, where two participating sides (players A and B), each owning his resource base, make their moves sequentially, one after another. Each move (“impact”) provides extraction (elimination) of some amount of resources from the RB of the moving player as well as from the RB of his opponent, i.e., to reduce opponent’s resource base, player must spend some part of his own RB. This situation is typical for conflicts of various nature. Along with resource base, every player has his moves base (MB), i.e. set of his possible moves; each move may be made only in the case, when player’s resource base contains all resources, necessary for this move. Also, every player may stop his activity, and after that, as well as in the case, where next move can’t be done by lack of resources, game is also stopped. So, game continues until next move may be done. Game result is join of players resource bases at the moment, when game is stopped. It is postulated, that every player owns full information about RB and MB of his opponent. Along with RB and MB every player at the beginning of the game formulates his goal, i.e. what conditions would satisfy both players resource bases when the game stops. Participating the game, any player wishes to reach his goal. If there exists such final state of the game, that both players goals are reached, game is called bi-winners; if only one player’s goal is reachable – one-winner; otherwise – no-winner, or empty. According to the background of game theory [1, 2], described game may be related as antagonistic and with sequential moves.

Let us consider multigrammatical representation of the described games, which since now will be strict mathematical definition of the verbally described earlier resource-based games.

First of all, we shall fix structure of lexemes, denoting resources; namely, the last would be composite objects of the form (a, x) , where a is name of the resource, $x \in \{A, B\}$ – owner of the resource, and “(,)” and “,” are dividers. So multiobject $n \cdot (a, x)$ describes the fact, that n units of resource a belong to (are in ownership of) player x . Thus, initial resource base of player A is multiset,

$$v_A = \{n_1^A \cdot (a_1, A), \dots, n_m^A \cdot (a_m, A)\}, \quad (27)$$

while initial RB of player B is multiset

$$v_B = \{n_1^B \cdot (a'_1, B), \dots, n_l^B \cdot (a'_l, B)\}. \quad (28)$$

Multiset

$$v_{A,B} = v_A \cup v_B \quad (29)$$

will be called initial resource base of the game.

For moves representation we shall use rules having following form:

$$\begin{aligned} &\{1 \cdot x, l_1 \cdot (a_{i_1}, x), \dots, b_k \cdot (a_{i_k}, x)\} \rightarrow \\ &\{-m_1 \cdot (a_{j_1}, y), \dots, -m_g \cdot (a_{j_g}, y), 1 \cdot y\}, \end{aligned} \quad (30)$$

where $x, y \in \{A, B\}$, and $x \neq y$. According to FMG semantics, application of rule (30) is possible, if left part of this rule is submultiset of the current multiset \bar{v} , generated by execution of previous steps.

The new and principal thing in (30) are negative multiplicities $-m_1, \dots, -m_g$, which from the substantial point of view provide elimination of regarding numbers of proper amounts of resources from the RB of opposite player y . To apply such kind of rules we shall redefine (16) as

$$\bar{v}' = \bar{v} - v + v'_+ - (-v'_-), \quad (31)$$

where

$$v'_+ = v'_+ + v'_- \quad (32)$$

$$-\{-m_1 \cdot a_1, \dots, -m_l \cdot a_l\} = \{m_1 \cdot a_1, \dots, m_l \cdot a_l\} \quad (33)$$

Let us illustrate (31)-(33) by following example

Example 7. Let

$$\bar{v} = \{1 \cdot A, 3 \cdot (\text{eur}, A), 4 \cdot (\text{usd}, A), 1 \cdot (\text{eur}, B), 3 \cdot (\text{usd}, B)\},$$

and rule r is

$$\{1 \cdot A, 2 \cdot (\text{eur}, A), 2 \cdot (\text{usd}, A)\} \rightarrow \{-3 \cdot (\text{eur}, B), 1 \cdot B\}.$$

according to (32)

$$v'_+ = \{1 \cdot B\},$$

$$v'_- = -3 \cdot (\text{eur}, B),$$

$$-v'_- = \{3 \cdot (\text{eur}, B)\},$$

and result of application of rule r to multiset \bar{v} is MS

$$\begin{aligned} \bar{v}' &= \{1 \cdot (\text{eur}, A), 2 \cdot (\text{usd}, A), 1 \cdot (\text{eur}, B), 3 \cdot (\text{usd}, B)\} + \{1 \cdot B\} - \{3 \cdot (\text{eur}, B)\} = \\ &= \{1 \cdot B, 1 \cdot (\text{eur}, A), 2 \cdot (\text{usd}, A), 3 \cdot (\text{usd}, B)\}. \blacksquare \end{aligned}$$

As it is easy to see, replacement of (16) by (31)-(32) is necessary to avoid negative multiplicities in generated multisets. This is done by transformation (31), which provides application of MS subtraction (8), which excludes appearance of such multiplicities, i.e. negative amounts of resources.

Now it is clear, that result of application of rule (30) is appearance of multioject $1 \cdot y$, which presence in the generated multiset provides possibility of the next move of player $y \neq x$. As seen, such form of rules exactly corresponds to the verbally described above logic and dynamic of the considered games.

Now we may construct filtering multigrammar representing game in full accordance with its verbal description.

This FMG $S = \langle v_0, R, F \rangle$ is such, that multiset

$$v_0 = \{1 \cdot A\} + v_{A,B}, \quad (34)$$

is kernel, corresponding to the initial state of the game, which starts from the player's A move with initial resource base of the game, defined by (27)-(29). Scheme R is join of moves bases of both players:

$$R = R_A \cup R_B, \quad (35)$$

where

$$R_A = \{r \mid \{1 \cdot A\} + v \rightarrow v' \in R\}, \quad (36)$$

$$R_B = \{r \mid \{1 \cdot B\} + v \rightarrow v' \in R\}, \quad (37)$$

and

$$F = F_A \cup F_B \quad (38)$$

is filter, being join of subfilters F_A and F_B , representing goals of participation of players in the game. Filter F provides selection of all terminal multisets which satisfy both players goals. These TMS form set \bar{V}_S , representing all possible results of the game, which satisfy both players. If $\bar{V}_S = \{\phi\}$, game is no-winner. If $\bar{V}_{S_A} \neq \{\emptyset\}$ and $\bar{V}_{S_B} = \{\emptyset\}$, winner is player A; if $\bar{V}_{S_A} = \{\emptyset\}$ and $\bar{V}_{S_B} \neq \{\emptyset\}$, winner is player B. Here $S_A = \langle v_0, R, F_A \rangle$, $S_B = \langle v_0, R, F_B \rangle$.

Let us illustrate described technique of RBG multiset modelling by example.

Example 7. Let initial resource base of player A is $\{4 \cdot a, 8 \cdot b, 9 \cdot c\}$, RB of player B – $\{6 \cdot a, 10 \cdot b, 3 \cdot c, 12 \cdot d\}$, and their capabilities (possible mutual impacts) are presented in the following tables.

Table 1.

№	Spent by A	Eliminated from B
1	$2 \cdot a, 1 \cdot b$	$1 \cdot a, 1 \cdot d$
2	$1 \cdot a, 3 \cdot b$	$2 \cdot b$
3	$1 \cdot a, 2 \cdot b, 3 \cdot c$	$1 \cdot b, 4 \cdot d$

Table 2.

№	Spent by B	Eliminated from A
1	$3 \cdot a, 2 \cdot b$	$2 \cdot a, 1 \cdot b$
2	$1 \cdot b, 1 \cdot c, 4 \cdot d$	$4 \cdot c$

Player's A goal is that, after game stops, his resource base would contain no less than 2 objects a , no less than 4 objects b and maximal possible number of objects d , while resource base of player B - minimal possible number of objects d , as well as no more than 2 objects b . Player's B goal is that, after game stops, his resource base would contain no less than 3 objects a , no less than 5 objects b , while resource base of player A would contain minimal possible number of objects c and no more than 3 objects b .

By this, game, which first move will be done by player A , may be represented by multigrammar $S = \langle v_0, R, F \rangle$, where

$$v_0 = \{4 \cdot (a, A), 8 \cdot (b, A), 9 \cdot (c, A), 6 \cdot (a, B), 10 \cdot (b, B), 3 \cdot (c, B), 12 \cdot (d, B), 1 \cdot A\}.$$

According to tables 1 and 2, scheme R will contain following rules marked r_i^A and r_j^B :

$$(r_1^A): \{1 \cdot A, 2 \cdot (a, A), 1 \cdot (b, A)\} \rightarrow \{-1 \cdot (a, B), -1 \cdot (d, B), 1 \cdot B\},$$

$$(r_2^A): \{1 \cdot A, 1 \cdot (a, A), 3 \cdot (b, A)\} \rightarrow \{-2 \cdot (a, B), 1 \cdot B\},$$

$$(r_3^A): \{1 \cdot A, 1 \cdot (a, A), 2 \cdot (b, A), 3 \cdot (c, A)\} \rightarrow \{-1 \cdot (b, B), -4 \cdot (d, B), 1 \cdot B\},$$

$$(r_1^B): \{1 \cdot B, 3 \cdot (a, B), 2 \cdot (b, B)\} \rightarrow \{-2 \cdot (a, A), -1 \cdot (b, A), 1 \cdot A\},$$

$$(r_2^B): \{1 \cdot B, 3 \cdot (a, B), 2 \cdot (b, B)\} \rightarrow \{-4 \cdot (c, A), 1 \cdot A\}.$$

Filter F contains following conditions:

$$(C_1^A): (a, A) \geq 2,$$

$$(C_2^A): (b, A) \geq 4,$$

$$(C_3^A): (c, A) = \max,$$

$$(C_4^A): (d, B) = \min,$$

$$(C_5^A): (b, B) \leq 2,$$

$$(C_1^B): (a, B) \geq 3,$$

$$(C_2^B): (b, B) \geq 5,$$

$$(C_3^B): (c, A) = \min,$$

$$(C_4^B): (b, A) \leq 3.$$

As seen, subfilter $F_A = \{C_1^A, \dots, C_5^A\}$ defines goal of player A , while subfilter $F_B = \{C_1^B, \dots, C_4^B\}$ - goal of player B .

Let us apply R to v_0 . To simplify representation of generation chains, we shall use table, which columns correspond to objects having place in the generated multisets, each represented by string of the table. Number n having place on the intersection of column (x, X) and string i is multiplicity of object (x, X) in multiset v_i , i.e. $n \cdot (x, X) \in v_i$. Each string, representing multiset v_i , is followed by string, representing rule, applied to v_i , in a form of multiplicities of objects having place in this rule; that, which are in the left part of rule, are negative numbers, while multiplicities from the right part are represented by positive numbers. Table corresponds to one generation chain, and two lowest strings represents subfilters F_A and F_B in such a way, that each boundary condition $(x, X) \theta n$ is represented by θn , and each optimizing condition $(x, X) = opt$ is represented by opt , both having place in column (x, X) .

Generation chain $v_0 \xRightarrow{r_1^A} v_1 \xRightarrow{r_1^B} v_2 \xRightarrow{r_4^A} v_3$ is represented by Table 3; as may be seen, multiset v_3 does not satisfy filter F .

We shall not continue generation by reason this game is no-winner (empty); it is obvious, if to consider conditions entering filter - some of them (for example, $(b, A) \geq 4 \in F_A$ and $(b, A) \leq 3 \in F_B$) are contradictory, i.e. no one final multiset generation by scheme R application to kernel v_0 will not satisfy them. ■

Let us consider some variations of the proposed basic techniques of multigrammatical modelling of resource-driven games.

Table 3.

$\frac{v}{r}$	A			B				A	B
	a	b	c	a	b	c	d		
v_0	4	8	9	6	10	3	12	1	—
r_1^A	-2	-3	—	-1	—	—	-1	-1	1
v_1	2	5	9	5	10	3	11	—	1
r_1^B	-2	-1	—	-3	-2	—	—	1	-1
v_2	—	4	9	2	8	3	11	1	—
r_4^A	—	—	—	—	—	—	—	-1	—
v_3	—	4	9	2	8	3	11	—	—
F_A	≥ 2	≥ 4	Max	—	≤ 2	—	min	—	—
F_B	—	≤ 3	min	≥ 3	≥ 5	—	—	—	—

More complicated case may include not only elimination of resources of the opposing player by spending own resources, but also change of their ownership (“their capturing”). Such moves may be represented by rules having following form:

$$\begin{aligned} & \{1 \cdot x, l_1 \cdot (a_{i_1}, x), \dots, l_k \cdot (a_{i_k}, x), \\ & m_1 \cdot (a_{j_1}, y), \dots, m_q \cdot (a_{j_q}, y)\} \rightarrow \\ & \{1 \cdot y, m_1 \cdot (a_{j_1}, x), \dots, m_q \cdot (a_{j_q}, x)\}. \end{aligned} \quad (39)$$

As seen, player x , spending l_1 objects a_{i_1}, \dots, l_k objects a_{i_k} , removes m_1 objects a_{j_1}, \dots, m_p objects a_{j_p} to his ownership from the opposing player. Unlike previous case (30), where result of every move was elimination of resources from both players resource bases, here occurs possibility of keeping resources reusable, but their owner becomes another player. (Note, that according to MG semantics, rule (39) will be applied only if all removed resources in necessary amounts have place in player’s y ownership).

Example 8. Let us join to the moves base (i.e. scheme R) of the Example 7 two new moves (rules), providing capturing resources of the opposing player:

$$(r_4^A): \{1 \cdot A, 3 \cdot (a, A), 1 \cdot (c, A), 1 \cdot (b, B), 1 \cdot (d, B)\} \rightarrow \{1 \cdot B, 1 \cdot (b, A), 1 \cdot (d, A)\},$$

$$(r_3^B): \{1 \cdot B, 1 \cdot (a, B), 3 \cdot (d, B), 1 \cdot (a, A), 2 \cdot (b, A)\} \rightarrow \{1 \cdot A, 1 \cdot (a, B), 2 \cdot (b, A)\}.$$

As seen, r_5^A application provides capturing one object b and one object d , belonging to player B , by player A , spending for this purpose three objects a and one object c . Similarly, r_4^B application provides removal of one object a and two objects b from A to B , the last spending for this action one object a and three objects d . ■

In general case there may be combination of two previous cases, when one move provides both elimination (destruction) of some resources of the opposing player as well as capturing of another his resources remaining reusable. Such “combined” moves have following form:

$$\begin{aligned} & \{1 \cdot x, l_1 \cdot (a_{i_1}, x), \dots, l_k \cdot (a_{i_k}, x), m_1 \cdot (a_{j_1}, y), \dots, m_p \cdot (a_{j_p}, y)\} \rightarrow \\ & \{1 \cdot y, -l'_1 \cdot (a'_{j_1}, y), \dots, -l'_q \cdot (a'_{j_q}, y), m_1 \cdot (a_{j_1}, x), \dots, m_p \cdot (a_{j_p}, x)\}. \end{aligned} \quad (40)$$

As seen, this move provides elimination of l'_1 objects a'_{j_1}, \dots, l'_q objects a'_{j_q} , belonging to player y , and removal of m_1 objects a_{j_1}, \dots, m_p objects a_{j_p} from his ownership to player x , who would spend for this purpose l_1 objects a_{i_1}, \dots, l_k objects a_{i_k} . Note, that (40) would be applicable only in the case when resource base of player y before move of player x includes no less than l'_1 objects a'_{j_1}, \dots, l'_q objects a'_{j_q} . Otherwise such operation would not be implemented.

The next level of complexity presumes that some player is capable to produce new objects from being already in his ownership, and aftermath to use produced resources during further steps of the game (conflict). To implement such feature, we shall introduce so-called internal moves, which are defined by rules, containing “control” multiobjects $1 \cdot x$ in both parts of rule:

$$\{1 \cdot x, l_1 \cdot (a_{i_1}, x), \dots, l_k \cdot (a_{i_k}, x)\} \rightarrow \{1 \cdot x, l'_1 \cdot (a'_{i_1}, x), \dots, l'_q \cdot (a'_{i_q}, x)\}. \quad (41)$$

As seen, after this rule application, player x remains active and will do the next move, and so until there will be applied rule with multiobject $1 \cdot y$ instead of $1 \cdot x$ in the right part. Moves described by (41) will be called internal.

Let us now describe, how player’s producing capabilities may be defined by means of internal moves techniques. For this purpose, we shall use intermediate “black box” representation for description of mentioned capabilities. Namely, we shall assume, that player x possesses set of producing (manufacturing) devices, each represented by “black box” with m inputs and one output (Fig.1). Every i -th input is marked

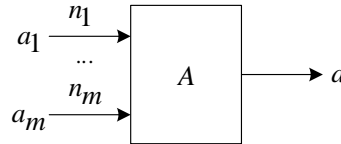


Figure 1.

by a_i – name of object (item, resource) – and n_i – amount (volume, quantity) of such objects. So graphical representation having place at Fig. 1 describes, that n_i objects a_1, \dots, n_m objects a_m are required for producing (manufacturing) one object a .

Each object produced by means of any device may be used in a “manufacturing chains” providing creation of new objects. So, the whole “manufacturing potential” (or “technological base”) of player x may be represented by set of rules like

$$\{1 \cdot x, n_1 \cdot a_1, \dots, n_m \cdot a_m\} \rightarrow \{1 \cdot x, 1 \cdot a\}, \quad (42)$$

each corresponding its own producing (manufacturing) device.

Some of produced objects may be used in moves, providing impacts on the opposing player, along with objects, used in these moves directly. Graphical representation of such game has place at Fig. 2, where players interaction (i.e. moves like (40)) is implemented through the depicted ellipse G.

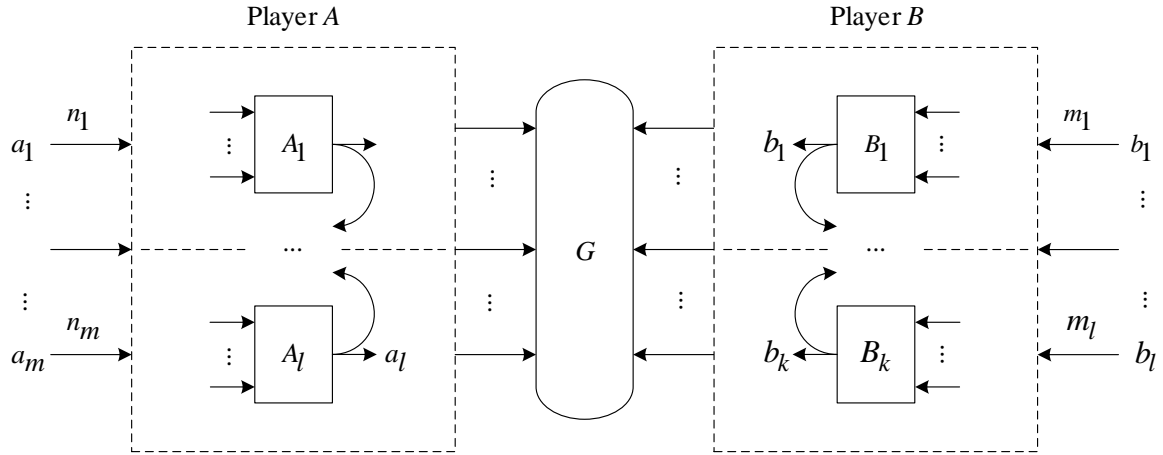


Figure 2.

In general case every or some of “black boxes” may have not one but $q > 1$ outputs, and then rules, representing it, have the following form:

$$\{1 \cdot x, n_1 \cdot a_1, \dots, n_m \cdot a_m\} \rightarrow \{1 \cdot x, 1 \cdot a'_1, \dots, 1 \cdot a'_q\}. \quad (43)$$

Let us illustrate introduced generalization by example.

Example 9. Let us join to the moves base (scheme R), obtained in the previous example 8, two new moves (rules), providing manufacturing objects, which may be used for preparing impacts:

$$(r_6^A): \{1 \cdot A, 1 \cdot (a, A), 2 \cdot (c, A)\} \rightarrow \{1 \cdot A, 1 \cdot (b, A)\},$$

$$(r_5^B): \{1 \cdot B, 1 \cdot (a, B), 1 \cdot (d, B)\} \rightarrow \{1 \cdot B, 1 \cdot (c, B)\}.$$

Since these rules appearance in the scheme R , generation chains (i.e. moves sequences) will implement manufacturing operations, providing inclusion to the player's A resource base additional objects b , each produced by spending one object a and two objects c , as well as to the player's B resource base additional objects c , each produced by spending one object a and one object d . ■

Let us note, that resources spent for manufacturing of new object may be its spare parts as well as consumables necessary for operation of manufacturing device (electrical energy, fuels etc.). Among such resources there would be obligatory “active state of the device”, represented by special multiobject $1 \cdot e$, where e is name of device. Such multiobjects may be, as all other presenting in the resource bases, eliminated (destroyed) and captured as a result of some moves of the opposing side. Such techniques provide simple representation of actions providing destructive impacts not only on resource base but also on technological base.

All the said concerned antagonistic RBG with sequential moves.

Simultaneous moves may be represented as follows. Namely, it is sufficient to use rules of the following form:

$$\{1 \cdot A, 1 \cdot B, l_1 \cdot (a_{i_1}, A), \dots, l_k \cdot (a_{i_k}, A), m_1 \cdot (a_{j_1}, B), \dots, m_q \cdot (a_{j_q}, B)\} \rightarrow \{1 \cdot A, 1 \cdot B\} \quad (44)$$

Of course, kernel of the corresponding multigrammar will be in this case $\{1 \cdot A, 1 \cdot B\} + v_A + v_B$, where v_A and v_B are initial resource bases of both players.

Goals of game for both players are, as higher, represented by subfilters F_A and F_B joined to one filter $F = F_A \cup F_B$ of corresponding filtering multigrammar.

Along with simultaneous moves, made by both players, there may be internal moves of the same form as (41). By this moves each player may prepare to simultaneous moves, made by both of them. All the said about such moves higher in this section is applicable to this kind of antagonistic RBG. As may be seen, multigrammatical representation provides easy use of both sequential and parallel moves in one game.

As it is known from game theory, there may be also cooperative games, where players interact without causing mutual damage, but, on the contrary, helping one another in order to provide their goals reachability. This kind of RBG is considered in the following section.

4. Resource-based cooperative and coalitional games

Let us begin from cooperative RBG.

Cooperation means, that there may be resources exchange between players, or one of them may deliver his own resources to another without any compensation. Also, of course, any player may produce new resources from his own, and also from resources, belonging to the second player.

Resources exchange may be represented by rule of the following form (in a simultaneous moves version):

$$\left\{1 \cdot A, 1 \cdot B, l_1 \cdot (a_{i_1}, A), \dots, l_k \cdot (a_{i_k}, A), m_1 \cdot (a_{j_1}, B), \dots, m_q \cdot (a_{j_q}, B)\right\} \rightarrow \left\{1 \cdot A, 1 \cdot B, m_1 \cdot (a_{j_1}, A), \dots, m_q \cdot (a_{j_q}, A), l_1 \cdot (a_{i_1}, B), \dots, l_k \cdot (a_{i_k}, B)\right\} \quad (45)$$

As may be seen, application of this rule results in transfer of l_1 units of resource a_{i_1}, \dots, l_k units of resource a_{i_k} , all belonging to player A , to the player B ownership. Instead of mentioned transferred resources player A will get m_1 units of resource a_{j_1}, \dots, m_q units of resource a_{j_q} , which until exchange belonged to player B .

In the “altruistic” case, when one player does not demand on any compensation from his partner, corresponding rule is as follows:

$$\left\{1 \cdot A, 1 \cdot B, l_1 \cdot (a_{i_1}, A), \dots, l_k \cdot (a_{i_k}, A)\right\} \rightarrow \left\{1 \cdot A, 1 \cdot B, l_1 \cdot (a_{i_1}, B), \dots, l_k \cdot (a_{i_k}, B)\right\}. \quad (46)$$

Cooperation inside producing (manufacturing) process presumes possible use of resources, that belong to one player, for production of resources, which, being manufactured, may belong to another player.

Structure of rules, representing producing (manufacturing) capabilities of players, is the same, as (42), but, for generality, multiobjects are extended by information about ownership, which, as was said higher, may be of both sides. Such rules, implementing internal moves of players, may be as follows:

$$\left\{1 \cdot x, n_1 \cdot (a_{i_1}, x_1), \dots, n_m \cdot (a_{i_m}, x_m)\right\} \rightarrow \left\{1 \cdot x, 1 \cdot (a, x_{m+1})\right\}. \quad (47)$$

This means, that player $x \in \{A, B\}$, doing his move, produces one unit of resource a and belonging to player $x_{m+1} \in \{A, B\}$, using for this purpose n_1 units of resource a_{i_1} and belonging to player $x_1 \in \{A, B\}, \dots, n_m$ units of resource a_{i_m} and belonging to player $x_m \in \{A, B\}$.

Of course, there may be multiobject $1 \cdot y$, where $y \neq x$, in the right part of rule (47). In this case move is not internal, because the next move may be done only by another player.

Filtering multiset grammar $S = \langle v_0, R, F \rangle$, which kernel v_0 is initial resource base joined with “starting” multiobject(s), scheme R contains rules of form (45) – (47), and filter F contains conditions, defining goals of both players, is representation of cooperative game with two players.

Further generalization of cooperative games with two players are cooperative games with arbitrary number of players $m \geq 2$ (multi-player games).

As may be seen, described techniques of multigrammatical representation of games provides simple and natural description of games with any number of players. It is sufficient to use multiobjects of the form $1 \cdot x$, where x is name of the player, in any combinations, in the right and left parts of the rules. Such games are easily represented by multigrammars, which schemes contain rules like

$$\{1 \cdot x_1, \dots, 1 \cdot x_m, l_1 \cdot (a_{i_1}, x_{i_1}), \dots, l_k \cdot (a_{i_k}, x_{i_k})\} \rightarrow \{1 \cdot x_1, \dots, 1 \cdot x_m, l'_1 \cdot (a_{j_1}, x_{j_1}), \dots, l'_{k'} \cdot (a_{j_{k'}}, x_{j_{k'}})\} \quad (48)$$

and kernel is

$$\{1 \cdot x_1, \dots, 1 \cdot x_m\} + v_1 + \dots + v_m, \quad (49)$$

where x_i is name of i -th player, and v_i is his initial resource base. As seen, rules like (47) represent cooperation of players in order to create such resource base, which would satisfy filter

$$F = F_1 \cup \dots \cup F_m, \quad (50)$$

where F_i is subfilter, defining goal of i -th player. Of course, along with rules (38) there may be rules, representing internal moves of any player, but also there may be moves, done simultaneously by any subset of players $\{x_{i_1}, \dots, x_{i_q}\} \subset \{x_1, \dots, x_m\}$. Such kind of cooperation may be implemented not only in the case, when players do help one another, but also in the case when some of them join in order to oppose another. Following terminology of classical game theory, such stable sets (groups) of players, synchronizing their moves and matching their goals, may be called coalitions, and such games – coalitional resource-based games. As may be seen, these games are antagonistic, regarding opposing coalitions, but cooperative inside any coalition.

Multigrammatical representation of games makes it easy to describe logic of behavior of any number of coalitions and players entering them.

5. Conclusions

Presented paper introduces a new class of games, covering a wide area of interactions of sociotechnological systems, and its basic novelty is consideration of interacting STS as entities, which consume, produce, destroy and capture various resources. Secondly, a novel multiset-based mathematical toolkit, integrating descriptonal properties of classical mathematical programming and modern knowledge engineering, is used for strict formalizing of the introduced games, and, thus, for solving problems in the fully automated mode, saving to the decision makers and their staff a minimal work on the MG-represented knowledge base creation and maintenance. “What – if” mode here may be used without any difficulties.

However, described approach in his present form is not free of some limitations.

First of them is lack of time in rules, representing resources motion and use. This makes impossible description of STS operation on the time scale, that is highly important for precise modelling of any kind of conflict and cooperation. An adequate tool for elimination of this drawback are temporal multiset grammars (TMG), introduced in brief in [19]. TMG operate so called temporal multisets having form $\{n_1 \cdot a_1, \dots, n_m \cdot a_m, n \cdot t\}$, where t is special object (“time”), and multiplicity n means, that MS $\{n_1 \cdot a_1, \dots, n_m \cdot a_m\}$ is actual at moment n since STS have begun its operation. Scheme of TMG contains so called temporal rules like

$$\{n_1 \cdot a_1, \dots, n_m \cdot a_m\} \rightarrow \{n'_1 \cdot a'_1, \dots, n'_l \cdot a'_l, k' \cdot \Delta t\}, \quad (51)$$

which application to temporal MS $\bar{v} + \{k \cdot t\}$ is as follows. If

$\{n_1 \cdot a_1, \dots, n_m \cdot a_m\} \subseteq \bar{v} \in V_S$
then temporal MS

$$\bar{v}' = \bar{v} - \{n_1 \cdot a_1, \dots, n_m \cdot a_m\} + \{n'_1 \cdot a'_1, \dots, n'_l \cdot a'_l, (k + k') \cdot t\} \in V_S \quad (52)$$

From the substantial point of view (52) means, that new temporal MS \bar{v}' will be actual at moment $k + k'$. In (51) Δt is one more special object (“time interval”), which multiplicity k' defines duration of replacement of MS $\{n_1 \cdot a_1, \dots, n_m \cdot a_m\}$ by MS $\{n'_1 \cdot a'_1, \dots, n'_l \cdot a'_l\}$ (in the case of producing devices/facilities k' means duration of manufacturing of collection of resources $\{n'_1 \cdot a'_1, \dots, n'_l \cdot a'_l\}$ given input collection of resources $\{n_1 \cdot a_1, \dots, n_m \cdot a_m\}$).

Another limitation flows out of absence in rules, representing moves, information about locations of players. This feature restricts family of the considered games by those, which have place at one compact location. However, it is not so difficult to represent distributed games, which represent conflicts and cooperation, ongoing in some wide geospatial areas. Techniques of such representation is based on inclusion to the introduced higher composite lexemes like (a, A) the third component p being symbolic or numeric description of point or area, where collection of n units of resource a belonging to player A would be located. So multiobjects used in rules are as $n \cdot (a, p, A)$. Such techniques, proposed in [19], was applied in [20], dedicated to the assessment of resilience of critical infrastructures to the destructive impacts, multiplied by cascading (chain) effects.

One more limitation of the proposed framework is that it is not directly applicable to the incomplete information games [28, 29, 30], which are most useful for practice. We have considered all kinds of games in presumption, that all information about resources and moves at all steps of the game (or generation steps while rules application) is available to all players. However, this presumption is far from reality, where every player at every moment is aware only about some part of resource bases and moves bases of other players. Moreover, this information in general case may be incomplete, ambiguous, unreliable, contradictive etc.; saying shortly, imperfect.

To handle such games, further generalization of multiset grammars, called multiset metagrammars (MMG), may be applied effectively.

MMG differ from MG by existence of multiplicities-variables and containing them metarules (MR), which provide natural implanting to the multigrammatical framework powerful toolkit of interval analysis [31, 32]. One valuable simplification of MMG called unitary multiset metagrammars is fully described in [17, 18].

The most general and powerful class of multiset grammars, providing natural representation and efficient solution of various games, are temporal multiset metagrammars (TMMG). This direction of the development of the multigrammatical framework and its implementation will be considered in future publications.

Acknowledgments

Author is grateful to Acad. Igor Bychkov, Acad. Yuriy Shokin, Prof. Fred Roberts, and Prof. Don Saari for useful discussions.

References

- [1] Bierman H S and Fernandez I F 1997 *Game theory for applied economists. 2d edition* (Addison-Wesley) p 480
- [2] Papayouanou P 2010 *Game Theory for Business: A Primer in Strategic Gaming* (Probabilistic Publishing) p 192
- [3] Harrington J E 2014 *Games, strategies, and decision making. 2d edition* (Worth Publishers) p 736
- [4] Shoham Y and Leyton-Brown K 2009 *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations* (New-York: Cambridge University Press) p 532
- [5] Belardinelli F and Argento F 2017 *Multi-Agent Systems and Agreements Technologies* (15th European Conf., EUMAS 2017, and 5th Int. Conf., AT 2017. Revised Selected Papers) p 554
- [6] Waldrop M M 2018 Free Agents: Monumentally complex models are gaming out disaster scenarios with millions of simulated people (*Science*) **360 (6385)** 144-147
- [7] Sycara K P 1998 Multiagent Systems *AI Magazine* **19(2)** 79-92
- [8] Yeoh W and Yokoo M 2012 Distributed Problem Solving *AI Magazine* **33(3)** 53-65
- [9] Pena D, Tschernykh A, Nesmachnow S, Massobrio R, Feoktistov A, Bychkov I, Radchenko G, Drozdov A Y, and Garichev S N 2019 Operating cost and quality of service optimization for multi-vehicle-type timetabling for urban bus systems *J. of Parallel and Distributed Computing* **133** 272-285
- [10] Adam N R and Dogramaci A 1979 *Current Issues in Computer Simulation* (New-York: Academic Press) p 292
- [11] Pooley R J 1987 *An Introduction to Programming in Simula* (Alfred Waller Ltd) p 344
- [12] 2007 *SIMSCRIPT III Users Manual* (San Diego, CA : CACI Products Company) p 152
- [13] Naraynasamy V, Wong K W , Rai S and Chiou A 2010 Complex Game Design Modeling (*Cultural Computing: Second IFIP TC 14, Entertainment Computing Symp., ECS2010*) 65-74
- [14] Sukhbatar S, Szlam A and Fergus R 2016 Learning Multiagent Communication with Backpropagation *Advances in Neural Information Processing Systems* 2244-2252
- [15] Havrylov S and Titov I 2017 Emergence of Language with Multi-agent Games: Learning to Communicate with Sequences of Symbols *Preprint cs-lg/1705.11192*
- [16] Rodriguez J 2018 Modern Game Theory and Multi-Agent Reinforcement Learning Systems *Towards Data Science*.
- [17] Sheremet I A 2010 *Recursive Multisets and their Applications* (Moscow: Nauka) p 292
- [18] Sheremet I A 2011 *Recursive Multisets and their Applications* (Berlin: NG Verlag) p 249
- [19] Sheremet I A 2018 Multiset Analysis of Consequences of Natural Disasters Impacts on Large-Scale Industrial Systems (*Data Science J.*) **17:4** 1-17
- [20] Sheremet I 2019 Multiset-based assessment of resilience of sociotechnological systems to natural hazards *Natural Hazards - Risks, Exposure, Response, and Resilience* ed J Tiefenbacher (London: Intech Open)
- [21] Sheremet I 2019 Multiset-Based Knowledge Representation for the Assessment and Optimization of Large-Scale Sociotechnical Systems *Enhanced Expert Systems* ed P Vizureanu (London: Intech Open)
- [22] Sheremet I 2019 Unitary Multiset Grammars and Metagrammars Algorithmics and Applications *Enhanced Expert Systems* ed P Vizureanu (London: Intech Open)
- [23] Petrovskiy A B 2003 *Spaces of sets and multisets* (Moscow: URSS) p 248
- [24] Petrovskiy A B 2018 *Theory of measured sets and multisets* (Moscow: Nauka) p 360
- [25] Date C I 2012 *An Introduction to Database Systems. 8th edition* (London: Pearson) p 247
- [26] Darwen H 2012 *An Introduction to Relational Database Theory* (London: bookboon.com) p 239
- [27] Chomsky N 2002 *Syntactic Structures* (The Hague: Mouton de Gruyter) p 118
- [28] Aumann R J and Maschler R B 1995 *Repeated Games with Incomplete Information* (Cambridge, MA: MIT Press) p 360
- [29] Jjiang D Y 2010 *Situation Analysis of Double Action Games with Entropy* (New-York: Science Press)

- [30] Sandomirsky F 2014 Repeated games of incomplete information with large sets of states (*Int. J. of Game Theory*) **43(4)** 767-789
- [31] Shokin Yu I 1996 On interval problems, interval algorithms and their complexity (*Computational Technologies*) **1(1)** 3
- [32] Hansen E and Walster G W 2004 *Global Optimization Using Interval Analysis* (New-York: Marcel Dekker) p 530