Multiset Storage Automata

Manfred Kudlek, Patrick Totzke, and Georg Zetzsche
{kudlek,3totzke,3zetzsch}@informatik.uni-hamburg.de

Department Informatik, MIN Fakultät, Universität Hamburg
http://www.informatik.uni-hamburg.de/TGI

Abstract. Two kinds of multiset automata with a storage attached, varying only in their ability of testing the storage for emptiness, are introduced, as well as normal forms. Their accepting power and relation to other multiset languages classes is investigated.

1 Introduction

Multiset languages and their characterizations play an important role in various areas of Theoretical Computer Science, as in Concurrency Theory or Membrane Computing. Generative models have been investigated in [4]. For word languages it is well known that they can be characterized by grammars as well as by automata. Analogous automata can be introduced for multiset languages. Multiset finite automata (MFA) have been considered in [1, 4]. Furthermore, [1] defines models of automata that are more expressive than MFA and correspond to type 1 and 0 multiset grammars.

Naturally one would first think of extending MFA by introducing a push down store. However, since $\oplus$ is commutative, the multiset language classes corresponding to regular and context-free word languages, coincide. Therefore one can show that extending MFA with a word stack accessible only from one end does not suffice to result in a more powerful model.

This paper focuses on the commutative equivalent of the PDA which we call multiset storage automaton (MSA) since it features a multiset storage. Whereas word pushdown automata are capable of testing for emptiness of the storage attached, this is not necessarily the case for multiset storage automata because of the commutativity of the basic operation $\oplus$ for multisets. Therefore we introduce two different models, one with emptiness testing, the other without.

In section 2 we introduce the necessary definitions, notations and facts of multisets, as well as of Petri nets. Sections 3 and 4 briefly present multiset grammars and non-deterministic multiset finite automata. In section 5 we introduce the two kinds of multiset storage automata, show normal forms, investigate accepting power, and present the relation to other classes of multiset languages as well as to Petri net languages.

One result is that the class of Parikh images of one class of Petri net languages is identical to one of the classes defined by multiset storage automata.

2 Multisets and Petri Nets

In this section we provide the basic definitions of multisets and Petri nets.

Definition 1. (Multisets) *If M is a set then $\alpha : M \to \mathbb{N}$ is a* multiset *over M, where $\mathbb{N}$ denotes the set of natural numbers including* 0*. The* norm *of α is defined by*

$$|\alpha| = \sum_{x \in M} \alpha(x) \ .$$

The set of all multisets over M is denoted by $M^{\oplus}$. If $|M| < \infty$ then $M^{\oplus}$ can be identified with $\mathbb{N}^{|M|}$. $M^{\oplus}$ is a commutative monoid with operation $\oplus$ and neutral element $\mathbf{0}_M$, defined by $\forall x \in M \ : \ (\alpha \oplus \beta)(x) = \alpha(x) + \beta(x)$ and $\forall x \in M \ : \ \mathbf{0}_M(x) = 0$.

The operation $\ominus$ is defined by $\forall x \in M \ : \ (\alpha \ominus \beta)(x) = max(0, \alpha(x) - \beta(x))$.

A partial order $\sqsubseteq$ is given by $\alpha \sqsubseteq \beta \Leftrightarrow \forall x \in M \ : \ \alpha(x) \leq \beta(x)$.
Futhermore, $\alpha \sqsubset \beta \Leftrightarrow \alpha \sqsubseteq \beta \wedge \alpha \neq \beta$. For $A \subseteq M^{\oplus}$, $B \subseteq M^{\oplus}$ let

$$A \oplus B = \bigcup_{\alpha \in A, \beta \in B} \alpha \oplus \beta \ .$$

If Σ is an alphabet, and $u \in \Sigma^*$ then $\Psi(u) \in \Sigma^{\oplus}$ denotes the Parikh *vector of u given by $|u|_x$ for $x \in \Sigma$. If $L \subseteq \Sigma^*$ then*

$$\Psi(L) = \{\Psi(u) \mid u \in L\} \subseteq \Sigma^{\oplus}$$

denotes the set of all Parikh vectors of L.

We also use the notation $\langle y \rangle$ for singleton multisets i.e. $\langle y \rangle(x) = 0$ for $y \neq x$ and $\langle y \rangle(y) = 1$. For any set B let $\bar{B} = \{\langle b \rangle \mid b \in B\}$.

The next definition presents the Petri net languages we will later relate to multiset storage automata. For more details see [2, 3].

Definition 2. (Labelled Petri net)

A labelled Petri net is a structure $N = (P, T, \tau, \Sigma, \rho, \mu_0, R_f)$ where P is a finite set of places*, T a finite set of* transitions*, $\tau : T \to P^{\oplus} \times P^{\oplus}$ a function defining the* pre- *and* post-conditions *of transitions, Σ a finite set of* labels*, $\rho : T \to \Sigma \cup \{\lambda\}$ the labelling function, $\mu_0 \in P^{\oplus}$ the initial marking, and $R_f \subseteq P^{\oplus}$ a finite set of final markings. A finite sequence of transitions (firing sequence) is expressed as an element $\sigma \in T^*$.*

For a subset $S \subseteq P^{\oplus}$, we define

$$L(N, S) = \{u \in \Sigma^* \mid \exists \sigma \in T^* \ \exists \mu \in S : \mu_0 \to_\sigma \mu, \ \rho(\sigma) = u\}$$

where $\mu_0 \to_\sigma \mu$ means that firing the sequence σ leads from marking μ_0 to μ. The languages *generated by N are defined as $L_f(N) := L(N, R_f)$ and $L(N) := L(N, P^{\oplus})$, respectively.*

The classes of languages of the form $L_f(N)$ and $L(N)$ are denoted by $\mathcal{L}_0^{\lambda}$ and $\mathcal{L}^{\lambda}$, respectively.

3 Multiset Grammars

We now recapitulate the notion of multiset grammars. In analogy to the Chomsky theory in the word case, a hierarchy of multiset grammars for the commutative case has been introduced [4].

Definition 3. (Multiset grammar)
A multiset grammar *is a structure* $G = (\Sigma_N, \Sigma_T, S, P)$ *where* Σ_N, Σ_T *are finite alphabets of* nonterminal *and* terminal *symbols, with* $\Sigma_N \cap \Sigma_T = \emptyset$ *and total alphabet* $\Sigma = \Sigma_N \cup \Sigma_T$*, a starting nonterminal* $S \in \Sigma_N$ *and a finite set of productions (rules)* $P \subseteq (\Sigma^{\oplus} \oplus \Sigma) \times \Sigma^{\oplus}$*. A production* $r = (\alpha, \beta) \in P$ *will also be referred to as* $\alpha \to \beta$*.*

A production $r = (\alpha, \beta) \in P$ *is applied on a multiset* μ *with result* ν *if* $\alpha \sqsubseteq \mu$ *and* $\nu = (\mu \ominus \alpha) \oplus \beta$*, also written as* $\mu \Rightarrow \nu$*. The reflexive transitive closure of* $\Rightarrow$ *is denoted by* $\overset{*}{\Rightarrow}$*.*

Definition 4. (Generated multiset language)
The sentential form multiset language $M_S(G)$ *generated by a multiset grammar* G *is given by* $M_S(G) = \{\mu \in \Sigma^{\oplus} \mid \langle S \rangle \overset{*}{\Rightarrow} \mu\}$*. The* multiset language $M(G)$ *generated by* G *is defined as* $M(G) = M_S(G) \cap \Sigma_T^{\oplus}$*. We call grammars* G *and* G' equivalent*, iff* $M(G) = M(G')$*.*

Definition 5. (Multiset grammar types)
Analogous to word grammars multiset grammars can be classified as follows :

a. arbitrary *if there are no restrictions,*
b. monotone *if* $|\alpha| \leq |\beta|$ *for all* $(\alpha, \beta) \in P$*,*
c. context-free *if* $|\alpha| = 1$ *for all* $(\alpha, \beta) \in P$*,*
d. regular *if* $\alpha = \langle A \rangle$*,* $\beta = \langle B \rangle \oplus \gamma$ *with* $A, B \in \Sigma_N$ *and* $\gamma \in \Sigma_T^{\oplus}$*, or* $\beta \in \Sigma_T^{\oplus}$ *for all* $(\alpha, \beta) \in P$*.*

The corresponding multiset language families are denoted by **mARB***,* **mMON***,* **mCF***, and* **mREG***, respectively. The classes of Parikh images of common word families are denoted by a prefix* **Ps***, e.g.* **PsCS** *or* **PsRE***.*

A somewhat counterintuitive result is that, since $\oplus$ is a commutative operation the classes **mREG** and **mCF** coincide. Finally it should be mentioned that **mARB** $\subset$ **PsRE** (Parikh images of **RE**) is a strict inclusion [4]. Therefore the following inclusions hold (**sLIN** denoting the family of semilinear sets):

Theorem 1. $\mathbf{mREG} = \mathbf{mCF} = \mathbf{sLIN} \subset \mathbf{mMON} \subseteq \mathbf{mARB} \subset \mathbf{PsRE}$ □

As for word grammars normal form theorems exist for multiset grammars [4]:

Theorem 2. (Normal forms)
To any multiset grammar $G = (\Sigma_N, \Sigma_T, S, P)$ *there exists an equivalent multiset grammar* $G' = (\Sigma'_N, \Sigma_T, S', P')$ *with productions only of the following forms, where* $A, B, C \in \Sigma'_N$*,* $a \in \Sigma_T$ *:*

a. arbitrary: $\langle A\rangle \oplus \langle B\rangle \to \langle A\rangle \oplus \langle C\rangle$, $\langle A\rangle \to \langle B\rangle \oplus \langle C\rangle$, $\langle A\rangle \to \langle a\rangle$, *or* $\langle A\rangle \to \mathbf{0}_\Sigma$,
b. monotone: $\langle A\rangle \oplus \langle B\rangle \to \langle A\rangle \oplus \langle C\rangle$, $\langle A\rangle \to \langle B\rangle \oplus \langle C\rangle$, $\langle A\rangle \to \langle a\rangle$, *or* $\langle S'\rangle \to \mathbf{0}_\Sigma$ *and* S' *does not appear on any right hand side.*
c. context-free: $\langle A\rangle \to \langle B\rangle \oplus \langle C\rangle$, $\langle A\rangle \to \langle a\rangle$, *or* $\langle A\rangle \to \mathbf{0}_\Sigma$,
d. regular: $\langle A\rangle \to \langle B\rangle \oplus \langle a\rangle$, $\langle A\rangle \to \langle a\rangle$, *or* $\langle A\rangle \to \mathbf{0}_\Sigma$

□

4 Multiset Finite Automata

In this section we briefly recall the definition and some results of multiset finite automata from [1] and [5].

Definition 6. (Multiset Finite Automaton)
A Multiset Finite Automaton (MFA *is a quintuple* $A = (Q, \Sigma, Q_F, q_0, \delta)$ *with a finite set of states* Q*, a finite alphabet* Σ*, a set of terminal states* $Q_F \subseteq Q$*, an initial state* q_0*, and a finite set of instructions* $\delta \subseteq Q \times \Sigma^\oplus \times Q$*.*

A configuration *is a pair* $(q, \mu) \in Q \times \Sigma^\oplus$*. A step* $(q, \mu) \vdash (q', \mu')$ *leads from one configuration to another if there is an* $\alpha \in \Sigma^\oplus$ *with* $\alpha \sqsubseteq \mu$*,* $(q, \alpha, q') \in \delta$*, and* $\mu' = \mu \ominus \alpha$*.* $\vdash^*$ *denotes the reflexive and transitive closure of* $\vdash$*.*

The automaton works in such a way that if $\mu \in \Sigma^\oplus$*,* $\alpha \in \Sigma^\oplus$ *with* $\alpha \sqsubseteq \mu$*,* $(q, \alpha, q') \in \delta$*, then* $\mu \in \Sigma^\oplus$ *is reduced to* $\mu' = \mu \ominus \alpha$*.*

A accepts a multiset μ *if* μ *is reduced in finitely many steps to* $\mathbf{0}_\Sigma$*, after which* A *is in a terminal state. The multiset language* $M(A)$*, accepted by a MFA* A *consits of all accepted multisets. Thus*

$$M(A) = \{\mu \in \Sigma^\oplus \mid \exists q \in Q_F \ : \ (q_0, \mu) \vdash^* (q, \mathbf{0}_\Sigma)\}.$$

Let **mFA** *denote the class of multiset languages accepted by MFA. Two Multiset finite automata* A *and* A' *are called* equivalent*, iff* $M(A) = M(A')$*.*

Note that due to the commutativity of the basic operation $\oplus$ *for multisets the choice between applicable instructions* (q, α, q') *and* (q, β, q'')*, possibly reading different parts of the input is non-deterministic.*

As for classical finite automata with respect to catenation a number of theorems can be shown to bring a MFA into a simple form.

Lemma 1. *To any MFA* $A = (Q, \Sigma, Q_F, q_0, \delta)$ *there exists an equivalent MFA* $A' = (Q', \Sigma, Q'_F, q_0, \delta')$ *with* $\delta' \subseteq Q' \times (\Sigma \cup \{\mathbf{0}_\Sigma\}) \times Q'$*.* □

Lemma 2. (Removal of $\mathbf{0}_\Sigma$)
To any MFA $A = (Q, \Sigma, Q_F, q_0, \delta)$ *an equivalent MFA* $A' = (Q', \Sigma, Q'_F, q'_0, \delta')$ *can be constructed, with* $\delta' \subseteq Q' \times \Sigma \times Q'$*.* □

Lemma 3. *The family of multiset languages accepted by MFA is identical to the class of regular multiset languages:* **mFA** = **mREG** = **mCF***.* □

5 Multiset Storage Automata

In order to extend the expressiveness of the MFA one might think of adding a word storage equivalent to the one used with push down automata (PDA). However, for such a multiset PDA A one can construct an ordinary PDA B such that $L(A) = \Psi(L(B))$ by replacing arcs that read singleton multisets by ones that read only that symbol. Now $L(A)$ is obviously regular (i.e. $L(A) \in \mathbf{mREG}$) since the Parikh images of contextfree languages ($\mathbf{PsCF}$) are known to coincide with $\mathbf{mCF} = \mathbf{mREG}$.

This gives reason to extend the MFA with a multiset storage rather than a pushdown stack. We continue by introducing these so called multiset storage automata in two variants: The first one has the ability to test the storage for emptiness, similar to word pushdown automata, whereas the other does not.

Definition 7. (Multiset storage automaton with 0-test)

A (nondeterministic) multiset storage automaton with 0-test (msaO) *is a structure of the form* $A = (Q, \Sigma, \Gamma, K, Q_F, Q_0, \perp)$ *where*

Q *is a finite set of states,*
Σ *a finite set of input symbols,*
Γ *a finite set of stack symbols,*
$K_1 \subseteq (Q \times \Sigma^{\oplus} \times \Gamma^{\oplus} \times \Gamma^{\oplus} \times Q)$,
$K_2 \subseteq (Q \times \Sigma^{\oplus} \times \{\langle\perp\rangle\} \times (\Gamma^{\oplus} \oplus \{\langle\perp\rangle\} \cup \{\mathbf{0}_{\Gamma\cup\{\perp\}}\}) \times Q)$,
$K = K_1 \cup K_2$,
$Q_F \subseteq Q$ *the set of final states,*
$Q_0 \subseteq Q$ *the set of initial states,*
$\perp \notin \Gamma$ *the* bottom *symbol*

A transition *is a quintuple* $t = (q, \alpha, \gamma, \delta, q') \in K$.
The division of the transition relation K *into* K_1 *and* K_2 *is just done to syntactically ensure that* $\perp$ *can always be read during a computation and if so, only as a singleton.*

A configuration *is a triple* $c = (q, \mu, \sigma)$ *where* $q \in Q$, $\mu \in \Sigma^{\oplus}$, *and* $\sigma \in (\Gamma^{\oplus} \oplus \{\langle\perp\rangle\}) \cup \{\mathbf{0}_{\Gamma\cup\{\perp\}}\}$. *Here* σ *represents the current content of the memory in which the symbol* $\perp$ *occurs exactly once.*
An initial configuration *has the form* $c_0 = (q_0, \mu_0, \langle\perp\rangle)$ *with* $q_0 \in Q_0$ *and* $\mu_0 \in \Sigma^{\oplus}$. *A* terminal configuration with final state *has the form* $c_f = (q_f, \mathbf{0}_{\Sigma}, \sigma)$ *with* $q_f \in Q_F$ *and* $\sigma \in (\Gamma^{\oplus} \oplus \{\langle\perp\rangle\}) \cup \{\mathbf{0}_{\Gamma\cup\{\perp\}}\}$. *A configuration of the form* $c_f = (q, \mathbf{0}_{\Sigma}, \mathbf{0}_{\Gamma\cup\{\perp\}})$ *with* $q \in Q$ *is called a* terminal configuration with empty stack.

A step $\vdash_0$ *is a relation defined by* $c \vdash_0 c'$ *iff* $c = (q, \mu, \sigma)$, $c' = (q', \mu', \sigma')$, *where* $(q, \alpha, \gamma, \delta, q') \in K$, $\alpha \sqsubseteq \mu$, $\sigma' = \sigma \ominus \gamma \oplus \delta$ *and* $\mu' = \mu \ominus \alpha$.
Additionally, if $t \in K_1$ *we require* $\gamma \sqsubseteq \sigma$ *and if* $t \in K_2$, $\gamma = \langle\perp\rangle = \sigma$ *must hold. Note that the* bottom *symbol* $\perp$ *can only be read if there are no other symbols in the storage thus being a test for the empty storage. Let* $\vdash_0^*$ *denote the reflexive and transitive closure of* $\vdash_0$.

The multiset language *accepted by a msaO* A *with final state is defined by*

$$M_f^0(A) = \{\mu \in \Sigma^{\oplus} \mid (q_0, \mu, \langle\bot\rangle) \vdash_0^* (q_f, \mathbf{0}_\Sigma, \sigma),\ \textit{for some}\ q_0 \in Q_0,\ q_f \in Q_F,\ \sigma \in (\Gamma^{\oplus} \oplus \{\langle\bot\rangle\}) \cup \{\mathbf{0}_{\Gamma\cup\{\bot\}}\}\}$$

The multiset language *accepted by a msaO* A *with empty stack is defined by*

$$M_0^0(A) = \{\mu \in \Sigma^{\oplus} \mid (q_0, \mu, \langle\bot\rangle) \vdash_0^* (q_f, \mathbf{0}_\Sigma, \mathbf{0}_{\Gamma\cup\{\bot\}}),\ q_0 \in Q_0\}$$

Let $\mathbf{mSA_f^0}$ *(*$\mathbf{mSA_0^0}$*) denote the class of multiset languages accepted by msaO with final state (empty stack).*

A quasi Σ lettering *MFA*^{0} *is a msaO with transitions solely of the form* $t = (q, \alpha, \gamma, \delta, q')$ *where* $\alpha \in (\bar{\Sigma} \cup \{\mathbf{0}_\Sigma\})$. *A msaO with transitions of the form* $t = (q, \alpha, \gamma, \delta, q')$ *where* $\gamma \in (\bar{\Gamma} \cup \{\bot\} \cup \{\mathbf{0}_{\Gamma\cup\{\bot\}}\})$ *is called* quasi Γ lettering.
In an analogous way a quasi $\Sigma\Gamma$ lettering *msaO is defined.*
A Σ lettering *msaO is a msaO with transitions* $t = (q, \alpha, \gamma, \delta, q')$ *where* $\alpha \in \bar{\Sigma}$.
A Γ lettering *msaO* A *is a msaO with transitions* $t = (q, \alpha, \gamma, \delta, q')$ *where* $\gamma \in (\bar{\Gamma} \cup \{\bot\})$. *Analogously* $\Sigma\Gamma$ *lettering msaO are defined.*

Two msaO A *and* A' *are called* equivalent*, iff* $M_0^0(A) = M_0^0(A')$ *and* $M_f^0(A) = M_f^0(A')$.

The following lemmata exhibit normal forms of MSA with 0-testing.

Lemma 4. *To each msaO* A *an equivalent msaO* $A' = (Q', \Sigma, \Gamma, K', Q'_F, Q'_0, \bot)$ *with* $|Q'_0| = 1$ *and* $|Q'_f| = 1$ *can be constructed.*

Proof. Let $A = (Q, \Sigma, \Gamma, K, Q_F, Q_0, \bot)$. Define $Q' = Q \cup \{q'_0, q'_F\}$, and $Q'_0 = \{q'_0\}$, $Q'_F = \{q'_F\}$, as well as

$$\begin{aligned} K' = K &\cup \{(q'_0, \mathbf{0}_\Sigma, \langle\bot\rangle, \langle\bot\rangle, q_0) \mid q_0 \in Q_0\} \\ &\cup \{(q_f, \mathbf{0}_\Sigma, \langle y\rangle, \langle y\rangle, q'_F) \mid y \in \Gamma, q_f \in Q_F\} \\ &\cup \{(q_f, \mathbf{0}_\Sigma, \langle\bot\rangle, \langle\bot\rangle, q'_F) \mid q_f \in Q_F\} \end{aligned}$$

Clearly, then $M_0^0(A') = M_0^0(A)$ and $M_f^0(A') = M_f^0(A)$. Note that the new transitions are Γ lettering. □

Lemma 5. *To each msaO* A *an equivalent quasi* Σ *lettering msaO* A' *can be constructed.*

Proof. Let $A = (Q, \Sigma, \Gamma, K, Q_F, Q_0, \bot)$.
For each transition $t = (q, \alpha, \gamma, \delta, q') \in K$ with $\alpha = \langle x_1\rangle \oplus \cdots \oplus \langle x_k\rangle$ and $k > 1$ define new states $Q_t = \{q_i(t) \mid 1 \le i < k\}$ and new transitions K'_t with $t(1) = (q, \langle x_1\rangle, \mathbf{0}_{\Gamma\cup\{\bot\}}, \mathbf{0}_{\Gamma\cup\{\bot\}}, q_1(t))$, $t(i) = (q_{i-1}(t), \langle x_i\rangle, \mathbf{0}_{\Gamma\cup\{\bot\}}, \mathbf{0}_{\Gamma\cup\{\bot\}}, q_i(t))$ for $1 < i < k$, $t(k) = (q_{k-1}(t), \langle x_k\rangle, \gamma, \delta, q')$.
Let the constructed quasi Σ lettering MSA be A'. Then it follows that $M_f^0(A') = M_f^0(A)$ and $M_0^0(A') = M_0^0(A)$. □

Lemma 6. *To each msaO A an equivalent quasi Γ lettering msaO A' can be constructed.*

Proof. Let $A = (Q, \Sigma, \Gamma, K, Q_F, Q_0, \perp)$.
Mor each transition $t = (q, \alpha, \gamma, \delta, q') \in K$ with $\gamma = \langle y_1 \rangle \oplus \cdots \oplus \langle y_k \rangle$ and $k > 1$ define new states $Q_t = \{q_i(t) \mid 1 \leq i < k\}$ as well as transitions K'_t with $t(1) = (q, \mathbf{0}_\Sigma, \langle y_1 \rangle, \mathbf{0}_{\Gamma \cup \{\perp\}}, q_1(t))$, $t(i) = (q_{i-1}(t), \mathbf{0}_\Sigma, \langle y_i \rangle, \mathbf{0}_{\Gamma \cup \{\perp\}}, q_i(t))$ $(1 < i < k)$, $t(k) = (q_{k-1}(t), \alpha, \langle y_k \rangle, \delta, q')$.
Let the constructed quasi Γ lettering MSA be A'. Then it follows that $M_f^0(A') = M_f^0(A)$ and $M_0^0(A') = M_0^0(A)$. □

Lemma 7. *To each msaO A an equivalent Γ lettering msaO A' can be constructed.*

Proof. Let $A = (Q, \Sigma, \Gamma, K, Q_F, Q_0, \perp)$. It can be assumed that A is quasi Γ lettering. Replace each transition $t = (q, \alpha, \mathbf{0}_\Gamma, \delta, q') \in K$ by new transitions $t(y) = (q, \alpha, \langle y \rangle, \langle y \rangle \oplus \delta, q')$ for $y \in \Gamma$ and $t(\perp) = (q, \alpha, \langle \perp \rangle, \langle \perp \rangle \oplus \delta, q')$. Let A' be the constructed Γ lettering MSA. Then it follows easily that $M_f^0(A') = M_f^0(A)$ and $M_0^0(A') = M_0^0(A)$. □

The next lemmata show the equivalence between acceptance with final state and empty stack for MSA with 0-testing.

Lemma 8. *To each msaO A accepting $M = M_f^0(A)$ with final state an msaO A' can be constructed accepting $M = M_0^0(A') = M_f^0(A')$ with empty stack and final state, i.e.* $\mathbf{mSA_f^0} \subseteq \mathbf{mSA_0^0}$.

Proof. Let $A = (Q, \Sigma, \Gamma, K, \{q_f\}, \{q_0\}, \perp)$. We introduce a new sink state q_s that can only be reached from the former final state and from which it is possible to blank the memory. Add the new transitions K':

$$\begin{aligned} K' = &\{(q_s, \mathbf{0}_\Sigma, \langle y \rangle, \mathbf{0}_{\Gamma \cup \{\perp\}}, q_s) \mid y \in \Gamma\} \\ &\cup \{t(\perp) = (q_s, \mathbf{0}_\Sigma, \langle \perp \rangle, \mathbf{0}_{\Gamma \cup \{\perp\}}, q_s)\} \\ &\cup \{(q_f, \mathbf{0}_\Sigma, \mathbf{0}_\Gamma, \mathbf{0}_{\Gamma \cup \{\perp\}}, q_s)\} \end{aligned}$$

Then $A' = (Q, \Sigma, \Gamma, K \cup K', \{q_s\}, \{q_0\}, \perp)$ is a newly constructed MSA, with $M_f^0(A') = M_0^0(A') = M_f^0(A) = M$. □

Lemma 9. *To each msaO A accepting $M = M_0^0(A)$ with empty stack an msaO A' can be constructed accepting $M = M_f^0(A') = M_0^0(A')$ with final state and empty stack, i.e.* $\mathbf{mSA_0^0} \subseteq \mathbf{mSA_f^0}$.

Proof. Let $A = (Q, \Sigma, \Gamma, K, Q_F, \{q_0\}, \perp)$. Now A can accept only by using a transition $t = (q, \alpha, \langle \perp \rangle, \mathbf{0}_{\Gamma \cup \{\perp\}}, q')$ since $\perp$ can only be read as a singleton and must be cleared to accept by empty memory. Therefore replace each such transition by $t' = (q, \alpha, \langle \perp \rangle, \mathbf{0}_{\Gamma \cup \{\perp\}}, q_f)$ where q_f is a new final state. Then $M_f^0(A') = M_0^0(A') = M$ trivially holds. □

From the previous lemmata follows

Theorem 3. $\mathbf{mSA_f^0} = \mathbf{mSA_0^0}$. *Moreover, the automata can be in* normal form*, i.e. having one initial and one final state, and being quasi Σ lettering and Γ lettering.* $\square$

Now we introduce multiset storage automata without 0-testing.

Definition 8. (Multiset storage automaton)

A (nondeterministic) multiset storage automaton (MSA) *is a structure of the form* $A = (Q, \Sigma, \Gamma, K, Q_F, Q_0, \triangle)$ *where*

Q *is a finite set of states,*
Σ *a finite set of input symbols,*
Γ *a finite set of stack symbols,*
$K \subseteq (Q \times \Sigma^{\oplus} \times \Gamma^{\oplus} \times \Gamma^{\oplus} \times Q)$
$Q_F \subseteq Q$ *the set of final states,*
$Q_0 \subseteq Q$ *the set of initial states,*
$\triangle \in \Gamma$ *a special symbol*

A transition *is a quintuple* $t = (q, \alpha, \gamma, \delta, q') \in K$.

A configuration *is a triple* $c = (q, \mu, \sigma)$ *where* $q \in Q$, $\mu \in \Sigma^{\oplus}$, *and* $\sigma \in \Gamma^{\oplus}$.
An initial configuration *has the form* $c_0 = (q_0, \mu_0, \langle\triangle\rangle)$ *with* $q_0 \in Q_0$ *and* $\mu_0 \in \Sigma^{\oplus}$. *A* terminal configuration with final state *has the form* $c_f = (q_f, \mathbf{0}_\Sigma, \sigma)$ *with* $q_f \in Q_F$ *and* $\sigma \in \Gamma^{\oplus}$.
And a configuration of the form $c_f = (q, \mathbf{0}_\Sigma, \mathbf{0}_\Gamma)$ *with* $q \in Q$ *is called a* terminal configuration with empty stack.

The step-relation $\vdash$ *is defined by* $c \vdash c'$ *iff* $c = (q, \mu, \sigma)$, $c' = (q', \mu', \sigma')$, *where* $\alpha \sqsubseteq \mu$, $\sigma' = \sigma \ominus \gamma \oplus \delta$, $\mu' = \mu \ominus \alpha, \gamma \sqsubseteq \sigma$, *and* $(q, \alpha, \gamma, \delta, q') \in K$.
Let $\vdash^*$ *denote the reflexive and transitive closure of* $\vdash$.

The multiset language *accepted by a MSA A with final state is defined by*

$$M_f(A) = \{\mu \in \Sigma^{\oplus} \mid (q_0, \mu, \langle\triangle\rangle) \vdash^* (q_f, \mathbf{0}_\Sigma, \sigma),\ \textit{for some}\ q_0 \in Q_0,\ q_f \in Q_F,\ \sigma \in \Gamma^{\oplus}\}$$

The multiset language *accepted by a MSA A with empty stack is defined by*

$$M_0(A) = \{\mu \in \Sigma^{\oplus} \mid (q_0, \mu, \langle\triangle\rangle) \vdash^* (q_f, \mathbf{0}_\Sigma, \mathbf{0}_\Gamma),\ q_0 \in Q_0\}.$$

Two MSA A and A′ are called equivalent*, iff* $M_0(A) = M_0(A')$ *and* $M_f(A) = M_f(A')$. *Let* $\mathbf{mSA_f}$ *and* $\mathbf{mSA_0}$ *denote the classes of multiset languages accepted by MSA with final state or empty stack respectively.*

The definitions of (Quasi-) lettering (Σ and Γ) for msaO apply analogously to MSA.

As for MSA with 0-testing we shall now investigate the notion of normal forms for automata without the ability of 0-testing.

Lemma 10. *To each MSA A an equivalent MSA $A' = (Q', \Sigma, \Gamma, K', Q'_0, Q'_F, \triangle)$ with $|Q'_0| = 1$ and $|Q'_f| = 1$ can be constructed.*

Proof. Let $A = (Q, \Sigma, \Gamma, K, Q_F, Q_0, \triangle)$, $Q' = Q \cup \{q'_0, q'_F\}$, $Q'_0 = \{q'_0\}$, and $Q'_F = \{q'_F\}$, as well as

$$\begin{aligned} K' = K &\cup \{(q_0, \mathbf{0}_\Sigma, \mathbf{0}_\Gamma, \mathbf{0}_\Gamma, q'_0) \mid q_0 \in Q_0\} \\ &\cup \{(q_f, \mathbf{0}_\Sigma, \mathbf{0}_\Gamma, \mathbf{0}_\Gamma, q'_F) \mid q_f \in Q_F\} \end{aligned}$$

Clearly $M_0(A') = M_0(A)$ and $M_f(A') = M_f(A)$ hold for the constructed MSA $A' = (Q', \Sigma, \Gamma, K', Q'_F, Q'_0, \triangle)$. □

Lemma 11. *To each MSA A an equivalent quasi Σ lettering MSA A' can be constructed.*

Proof. Let $A = (Q, \Sigma, \Gamma, K, Q_F, Q_0, \triangle)$.
For each transition $t = (q, \alpha, \gamma, \delta, q') \in K$ with $\alpha = \langle x_1 \rangle \oplus \cdots \oplus \langle x_k \rangle$ and $k > 1$ define new states $Q_t = \{q_i(t) \mid 1 \leq i < k\}$ and new transitions K'_t with $t(1) = (q, \langle x_1 \rangle, \langle \triangle \rangle, \langle \triangle \rangle, q_1(t))$, $t(i) = (q_{i-1}(t), \langle x_i \rangle, \langle \triangle \rangle, \langle \triangle \rangle, q_i(t))$ $(1 < i < k)$, $t(k) = (q_{k-1}(t), \langle x_k \rangle, \gamma, \delta, q')$. Let the constructed quasi Σ lettering MSA be A'. Then it follows that $M_f(A') = M_f(A)$ and $M_0(A') = M_0(A)$. □

Lemma 12. *To each MSA A an equivalent quasi Γ lettering MSA A' can be constructed.*

Proof. Let $A = (Q, \Sigma, \Gamma, K, Q_F, Q_0, \triangle)$.
For each transition $t = (q, \alpha, \gamma, \delta, q') \in K$ with $\gamma = \langle y_1 \rangle \oplus \cdots \oplus \langle y_k \rangle$ and $k > 1$ define new states $Q_t = \{q_i(t) \mid 1 \leq i < k\}$ and new transitions K'_t with $t(1) = (q, \mathbf{0}_\Sigma, \langle y_1 \rangle, \mathbf{0}_\Gamma, q_1(t))$, $t(i) = (q_{i-1}(t), \mathbf{0}_\Sigma, \langle y_i \rangle, \mathbf{0}_\Gamma, q_i(t))$ $(1 < i < k)$, $t(k) = (q_{k-1}(t), \alpha, \langle y_k \rangle, \delta, q')$. Let the constructed quasi Σ lettering MSA be A'. Then it follows that $M_f(A') = M_f(A)$ and $M_0(A') = M_0(A)$. □

Lemma 13. *To each MSA A an equivalent Γ lettering MSA A' can be constructed.*

Proof. Let $A = (Q, \Sigma, \Gamma, K, \{q_F\}, \{q_0\}, \triangle)$. A can be assumed to be quasi Γ lettering. The idea of the proof is to replace each transition that does not read from the memory by one that only reads and writes the fixed new symbol $\square \notin \Gamma$ to the memory. Thus define a new MSA A' with storage alphabet $\Gamma' := \Gamma \cup \{\square\}$ and replacing K with

$$\begin{aligned} K' = &\{(q_0, \mathbf{0}_\Sigma, \langle \triangle \rangle, \langle \triangle \rangle \oplus \langle \square \rangle, q_0)\} \\ &\cup \{(q, \alpha, \langle y \rangle, \delta, q') \mid (q, \alpha, \langle y \rangle, \delta, q') \in K, y \in \Gamma\} \\ &\cup \{(q, \alpha, \langle \square \rangle, \langle \square \rangle \oplus \delta, q') \mid (q, \alpha, \mathbf{0}_\Gamma, \delta, q') \in K\} \\ &\cup \{(q, \mathbf{0}_\Sigma, \langle \square \rangle, \mathbf{0}_\Gamma, q) \mid q \in Q\} \end{aligned}$$

Then it follows that $M_f(A') = M_f(A)$ and $M_0(A') = M_0(A)$. □

The following lemmata exhibit the relation between multiset language classes defined by MSA and other classes of multiset languages.

Lemma 14. *To each MSA A accepting $M = M_f(A)$ with final state an MSA A' can be constructed accepting $M = M_0(A') = M_f(A')$ with empty stack and final state, i.e.* $\mathbf{mSA_f} \subseteq \mathbf{mSA_0}$.

Proof. Let $A = (Q, \Sigma, \Gamma, K, \{q_f\}, \{q_0\}, \triangle)$. Add a new final state q'_F and to K new transitions $t(0) = (q_f, \mathbf{0}_\Sigma, \mathbf{0}_\Gamma, \mathbf{0}_\Gamma, q'_F)$, $t(y) = (q'_F, \mathbf{0}_\Sigma, \langle y\rangle, \mathbf{0}_\Gamma, q'_F)$ $(y \in \Gamma)$ and $t(\triangle) = (q'_F, \mathbf{0}_\Sigma, \langle\triangle\rangle, \mathbf{0}_\Gamma, q'_F)$.
The newly defined MSA A' is a copy of A that resigns in the new final state and cleans up its memory iff a final state has been reached in A.
Thus $M_f(A') = M_0(A') = M_f(A) = M$ holds. □

Lemma 15. $\mathbf{mSA_0} \subseteq \mathbf{mSA_0^0}$.

Proof. Let $A = (Q, \Sigma, \Gamma, K, \{q_f\}, \{q_0\}, \triangle)$. From this construct a msaO $A' = (Q', \Sigma, \Gamma, K', \{q'_F\}, \{q'_0\}, \perp)$ with $Q' = Q \cup \{q'_F, q'_2\}$ and $K' = K \cup \{(q, \mathbf{0}_\Sigma, \langle\perp\rangle, \mathbf{0}_\Gamma, q'_F) \mid q \in Q\}$. Then $M_0^0(A') = M_f^0(A') = M_0(A)$, since A' reaches the empty stack iff A reaches the empty stack. □

Lemma 16. $\mathbf{mARB} \subseteq \mathbf{mSA_0}$.

Proof. Let $G = (\Sigma_N, \Sigma_T, \{\langle S\rangle\}, R)$ be an arbitrary multiset grammar in *normal form*, i.e. $S \in \Sigma_N$ and the productions are of the following normal forms $(\langle B\rangle \oplus \langle C\rangle), \langle D\rangle \oplus \langle E\rangle)$, $(\langle B\rangle, \langle C\rangle \oplus \langle D\rangle)$, $(\langle B\rangle, \langle a\rangle)$, or $(\langle B\rangle, \mathbf{0}_\Sigma)$ where $\Sigma = \Sigma_N \cup \Sigma_T$, $B, C, D, E \in \Sigma_N$, $a \in \Sigma_T$. Construct a MSA_0

$A = (\{q_0, q_1, q_2\}, \Sigma_T, \Sigma_N \cup \{\triangle\}, K, \{q_2\}, \{q_0\}, \triangle)$ with
$(q_0, \mathbf{0}_{\Sigma_T}, \mathbf{0}_{\Sigma_N}, \langle S\rangle \oplus \langle\triangle\rangle, q_1) \in K$
$(q_1, \mathbf{0}_{\Sigma_T}, \langle B\rangle \oplus \langle C\rangle, \langle D\rangle \oplus \langle E\rangle, q_1) \in K$ if $(\langle B\rangle \oplus \langle C\rangle), \langle D\rangle \oplus \langle E\rangle) \in R$,
$(q_1, \mathbf{0}_{\Sigma_T}, \langle B\rangle, \langle C\rangle \oplus \langle D\rangle, q_1) \in K$ if $(\langle B\rangle, \langle C\rangle \oplus \langle D\rangle) \in R$,
$(q_1, \langle x\rangle, \langle B\rangle, \mathbf{0}_{\Sigma_N}, q_1) \in K$ if $(\langle B\rangle, \langle x\rangle) \in R$,
$(q_1, \mathbf{0}_{\Sigma_T}, \langle B\rangle, \mathbf{0}_{\Sigma_N}, q_1) \in K$ if $(\langle B\rangle, \mathbf{0}_\Sigma) \in R$,
$(q_1, \mathbf{0}_{\Sigma_T}, \triangle, \mathbf{0}_{\Sigma_N}, q_2) \in K$.

Then $M(G) \subseteq M_0(A)$, since any derivation of G can be simulated by a computation of A ending with empty stack. $M_0(A) \subseteq M(G)$, since A reaches the empty stack only if the entire input $\alpha \in \Sigma_T^\oplus$ has been read.
Therefore $M_0(A) = M(G)$. □

Lemma 17. $\mathbf{mSA_0} \subseteq \mathbf{mARB}$.

Proof. Let $A = (Q, \Sigma, \Gamma, K, \{q_f\}, \{q_0\}, \triangle)$. It can be assumed that A is quasi Σ lettering and Γ lettering. Now construct from this the multiset grammar $G = (\Sigma_N, \Sigma_T, \{S\}, R)$ with

$\Sigma_N = Q \cup \Gamma$, $\Sigma_T = \Sigma$, $S = q_0$, and
$(\langle q\rangle \oplus \langle y\rangle, \langle q'\rangle \oplus \delta) \in R$ if $(q, \mathbf{0}_\Sigma, \langle y\rangle, \delta, q') \in K$,
$(\langle q\rangle \oplus \langle y\rangle, \langle q'\rangle \oplus \langle x\rangle \oplus \delta) \in R$ if $(q, \langle x\rangle, \langle y\rangle, \delta, q') \in K$,
$(\langle q\rangle, \mathbf{0}_{\Sigma_N}) \in R$ $(q \in Q)$, corresponding to $(q, \mathbf{0}_\Sigma, \langle\triangle\rangle, \mathbf{0}_\Gamma, q') \in K$.

If A reaches the empty stack, reading only $\mu_0' \sqsubseteq \mu_0$, $\mu_0' \in M_0(A)$ would hold. But G produces $\mu_0' \in \Sigma^{\oplus}$ iff A reaches the empty stack.
Therefore $M(G) = M_0(A)$. □

Lemma 18. $\mathbf{mSA_0} \subseteq \Psi(\mathcal{L}_0^{\lambda})$.

Proof. For $M \in \mathbf{mSA_0}$ let $A = (Q, \Sigma, \Gamma, K, \{q_f\}, \{q_0\}, \triangle)$ be a quasi Σ lettering and Γ lettering MSA with $M_0(A) = M$.
Construct a labelled Petri net $N = (P, T, \tau, \Sigma, \rho, \mu_0', R_f)$ with
$P = Q \cup \Gamma$,
$\tau : T \to (P^{\oplus} \oplus P) \times P^{\oplus}$ such that $\tau(t) = ({}^{\bullet}t, t^{\bullet})$,
$\rho : T \to (\Sigma \cup \{\lambda\})$,
$\mu_0' \in P^{\oplus}$, $R_f \subseteq P^{\oplus}$, $|R_f| < \infty$.
Now let $T = \{t_k \mid k = (q, \alpha, \langle y \rangle, \delta, q') \in K\}$ with
$\tau(t_k) = (\langle q \rangle \oplus \langle y \rangle, \langle q' \rangle \oplus \delta)$ and
$\rho(t_k) = \alpha \in (\Sigma \cup \{\lambda\})$
$\mu_0' = \langle q_0 \rangle$, $R_f = Q$.
Then the application of $k = (q, \alpha, \langle y \rangle, \delta, q')$ to the configuration (q, μ, σ) giving $(q', \mu \ominus \alpha, \sigma \ominus \langle y \rangle \oplus \delta)$ corresponds to a firing of t_k in N transforming $\langle q \rangle \oplus \sigma$ into $\langle q' \rangle \oplus (\sigma \ominus \langle y \rangle \oplus \delta)$, and vice versa.
If $(q_0, \mu, \langle \triangle \rangle) \vdash^* (q, \mathbf{0}_{\Sigma}, \mathbf{0}_{\Gamma})$ for some $q \in Q$ then there exists a firing sequence $\varphi = t_{k(1)} \cdots t_{k(n)}$ with $\Psi(\rho(\varphi)) = \mu$ such that $\mu \in L_f(N)$.
On the other hand, if $\mu \in L_f(N)$ then there must exists a firing sequence $\varphi = t_{k(1)} \cdots t_{k(n)}$ of N such that $\langle q \rangle$ with $q \in Q$ is reached by N, and
$\mu = \Psi(\rho(\varphi))$. □

Lemma 19. $\Psi(\mathcal{L}_0^{\lambda}) \subseteq \mathbf{mSA_0}$.

Proof. Let $M \in \Psi(\mathcal{L}_0^{\lambda})$. Then there exists $L \in \mathcal{L}_0^{\lambda}$ such that $M = \Psi(L)$, and a labelled Petri net $N = (P, T, \tau, \Sigma, \rho, \sigma_0', R_f)$ with $\sigma_0' \in P^{\oplus}$ and $L_f(N) = L$. Let $\tau(t) = ({}^{\bullet}t, t^{\bullet})$.

Construct a MSA $A = (\{q_0, q_f\}, \Sigma, \Gamma, K, \{q_f\}, \{q_0\}, \triangle)$ with $\Gamma = P \cup \{\triangle\}$, $K = \{(q_0, \alpha, {}^{\bullet}t, t^{\bullet}, q_0) \mid t \in T, \rho(t) = \alpha\} \cup \{(q_0, \mathbf{0}_{\Sigma}, \sigma_0' \oplus \langle \triangle \rangle, \mathbf{0}_{\Gamma}, q_f)\}$.

If $\sigma \to \sigma'$ in N with t and $\rho(t) = \alpha$ then $(q_0, \mu, \sigma) \vdash (q_0, \mu \ominus \alpha, \sigma \ominus {}^{\bullet}t \oplus t^{\bullet})$ in A.
If $\sigma_0' \to^* \sigma_F'$ in N then there exists a firing sequence $\varphi = t_1, \cdots, t_n$ with $\rho(\varphi) = u$ and $\Psi(u) = \mu$. Then $(q_0, \mu, \langle \triangle \rangle) \vdash^* (q_f, \mathbf{0}_{\Sigma}, \mathbf{0}_{\Gamma})$ is a corresponding sequence in A. On the other hand, if $(q_0, \mu, \langle \triangle \rangle) \vdash^* (q_f, \mathbf{0}_{\Sigma}, \mathbf{0}_{\Gamma})$ then there exists a firing sequence $\varphi = t_1, \cdots, t_n$ with $\rho(\varphi) = u$ and $\Psi(u) = \mu$.
Therefore, $\Psi(L_f(N)) = M(A)$. □

From the previous lemmata follows

Corollary 1. $\Psi(\mathcal{L}_0^{\lambda}) = \mathbf{mSA_0} = \mathbf{mARB}$. □

Lemma 20. $\Psi(\mathcal{L}^\lambda) \subseteq \mathbf{mSA_f}$.

Proof. Tis can be proved as in Lemma 19, except that the MSA A accepts with final state, $(q_0, \mu, \langle\triangle\rangle) \vdash^* (q_f, \mathbf{0}_\Sigma, \sigma)$ for some $\sigma \in \Gamma^\oplus$, and the language accepted by the labelled Petri net N is defined by arbitrary firing sequences $\sigma_0' \to^* \sigma'$. □

Lemma 21. $\mathbf{mREG} = \mathbf{mCF} = \mathbf{mFA} \subseteq \mathbf{mSA_f}$.

Proof. This follows from the fact that a MFA is a special MSA (without effective use of its stack). □

Lemma 22. $\Psi(\mathcal{L}^\lambda) \subset \mathbf{mSA_f}$.

Proof. This follows from the fact that $M = \{\langle a\rangle \oplus \langle a\rangle\} \in \mathbf{mREG}$.
But $M \notin \Psi(\mathcal{L}^\lambda)$ since for any $L \in \mathcal{L}^\lambda$ holds: if $uv \in L$ then $u \in L$. □

Lemma 23. $\Psi(\mathcal{L}^\lambda) \not\subseteq \mathbf{mREG}$.

Proof. For $L \subseteq \Sigma^*$ let $pref(L) = \{u \in \Sigma^* \mid \exists v \in \Sigma^* : uv \in L\}$ be the set of all prefixes of words in L. Now $\mathcal{L}^\lambda$ is closed under intersection with $\mathbf{Pref}(\mathbf{REG}) = \{L = pref(R) \mid R \in \mathbf{REG}\}$. This follows from the fact that any DFA can be interpreted as a special Petri net where a DFA transition (q, a, q') is interpreted as a tranition s with $\tau(s) = (q, q')$ and $\rho(s) = a$.

If N is a Petri net having transitions t with $\tau(t) = ({}^\bullet t, t^\bullet)$ and $\rho(t) = a$ or $\rho(t) = \lambda$, construct a new Petri net N' having all λ transitions of N, and the others replaced by transitions (s, t) with $\tau((s, t)) = ({}^\bullet t \oplus \langle q\rangle, t^\bullet \oplus \langle q'\rangle)$ and $\rho((s, t)) = a$.

Consider the language $L_b = \{wc^k \mid w \in \{a, b\}^*, k \leq bin(w)\}$, $bin(w) \in \mathbb{N}$ denoting the integer represented by w. $L_b \in \mathcal{L}^\lambda$ ([3]).
Then $L' = L_b \cap pref(\{b\}\{a\}^*\{c\}^*) = \{ba^kc^n \mid 0 \leq k \leq 2^n\} \in \mathcal{L}^\lambda$ where a, b represent 0,1, respectively.
Now L' is not semilinear, and therefore $\Psi(L') \notin \mathbf{mREG}$ but $\Psi(L') \in \Psi(\mathcal{L}^\lambda)$. □

Corollary 2. $\mathbf{mREG} \subset \mathbf{mSA_f}$. □

From the previous lemmata follows

Theorem 4. $\Psi(\mathcal{L}^\lambda) \subset \mathbf{mSA_f} \subseteq \mathbf{mARB} = \Psi(\mathcal{L}_0^\lambda) = \mathbf{mSA_0}$
$\subseteq \mathbf{mSA_f^0} = \mathbf{mSA_0^0} \subseteq \mathbf{PsRE}$ □

6 Conclusion

So far we did not consider the deterministic versions of multiset storage automata. Several different variants of such are possible. The investigation of such deterministic multiset finite and storage automata will be given in another paper. Another topic is closure properties and algebraic characterizations of multiset languages accepted by multiset storage automata which will be dealt with in the forthcoming paper also.

References

1. E. Csuhaj-Varjú, C. Martín Vide, V. Mitrana (2001): *Multiset Automata.* In: Multiset Processing, eds. C. Calude, Gh. Păun, G. Rozenberg, A. Salomaa, *LNCS* **2235**, 69-83.
2. M. Hack (1976): *Petri Net Languages.* MIT Technical Report 159.
3. M. Jantzen (1979): *On the Hierarchy of Petri Net Languages. RAIRO* ITTI, **13**, nr. 1, 19-30.
4. M. Kudlek, C. Martín Vide, Gh. Păun (2001): *Toward FMT (Formal Macroset Theory).* In: Multiset Processing, eds. C. Calude, Gh. Păun, G. Rozenberg, A. Salomaa, *LNCS* **2235**, 123-134.
5. M. Kudlek, V. Mitrana (2001): *Closure Properties of Multiset Language Families. Fund. Inf.*, **49**, 191-203.