Mutually Accepting Capacitated Automata

Ravid Alon and Orna Kupferman
The Hebrew University, Israel

Abstract

We study *capacitated automata* (CAs) [10], where transitions correspond to resources and have capacities, bounding the number of times they may be traversed. We follow the *utilization semantics* of CAs and view them as recognizers of *multi-languages* – sets of multisets of words, where a multiset S of words is in the multi-language of a CA A if all the words in S can be mutually accepted by A: the multiset of runs on all the words in S together respects the bounds induced by the capacities. Thus, capacitated automata model possible utilizations of systems with bounded resources. We study the basic properties of CAs: their expressive power in the nondeterministic and deterministic models, closure under classical operations, and the complexity of basic decision problems.

1 Introduction

Finite state automata are used in the modelling and design of finite-state systems and their behaviors, with applications in engineering, databases, linguistics, biology, and many more. The traditional definition of an automaton does not refer to its transitions as consumable resources. Indeed, a run of an automaton is a sequence of successive transitions, and there is no bound whatsoever on the number of times that a transition may be traversed. In some settings, the use of a transition may correspond to the use of some resource. For example, it may be associated with the usage of some energy-consuming machine, application of some material, or consumption of bandwidth.

In [6], the authors introduced *Parikh automata*, which do impose restrictions related to consumption. Essentially, a Parikh automaton is a pair $\langle A, C\rangle$, where A is a nondeterministic finite automaton (NFA) over alphabet Σ, and $C \subseteq \mathbb{N}^{\Sigma}$ is a set of "allowed occurrences". A word w is accepted by $\langle A, C\rangle$ if A accepts w and the Parikh's commutative image of w, which maps each letter in Σ to its number of occurrences in w, is in C. Thus, the semantics views occurrences of letters as consumable resources. Several variants of Parikh automata have seen studied. In particular, [3] studied *constrained automata*, a variant that counts traversals of transitions and requires the vector of counters to belong to C, now a semi-linear set of allowed vectors. Additional models include *multiple counters automata* [4], where transitions can be taken only if guards referring to traversals so far are satisfied, and *queue-content decision diagrams*, which are used to represent queue content of FIFO-channel systems [1, 2].

In [10], the authors introduced *capacitated automata* (CAs).[1] In this model, transitions correspond to resources and may have bounded capacities. Formally, each transition is associated with a (possibly infinite) integral bound on the number of times it may be traversed. A word w is accepted by a CA A if A has an accepting run on w; one that reaches an accepting state and respects the bounds on the transitions. The study of CAs considers two possible semantics to them. The first, which is more related to the models described above, views CAs as recognizers of formal languages (see also [9]). The second, referred to in [10] as the *utilization semantics*, is related to traditional resource-allocation theory, and views CAs as labeled flow networks.

Our work here focuses on the second view. In order to understand and motivate it, let us consider a simple example. Consider the CA A appearing in Figure 1. In the first semantics, we view A as a recognizer of a language of words. Then, for example, the word ab is accepted by A, as the run q_0, q_0, q_2 on it "consumes" the selfloop in q_0 and the transition from q_0 to q_2. Likewise, the word ac is accepted by A, by its run q_0, q_1, q_3, and so does the word aac, by the run q_0, q_0, q_1, q_3. On the other hand, the word aab is not accepted by A, as an accepting run on it has to traverse the selfloop in q_0 twice, yet the capacity of this selfloop in only 1.

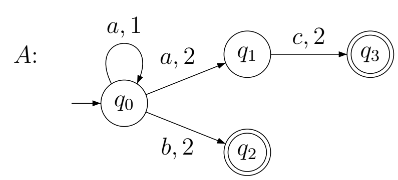

Figure 1: A CA that mutually accepts $\{b, b, ac, ac\}$, $\{ab, b, ac, ac\}$, $\{b, b, aac, ac\}$, and their sub-multisets.

We now proceed to the utilization semantics. Recall that both ab and ac are accepted by A. In fact, since the transitions consumed by both runs together respect the capacity bounds, the A *mutually accepts* the multiset $\{ab, ac\}$. We use the term *multiset*, namely a set with possible re-occurrences of elements, as words may be accepted by A several times, and we care about the number of times that each word is accepted. For example, since the capacity of the transitions $\langle q_0, a, q_1\rangle$ and $\langle q_1, c, q_2\rangle$ is 2, then A mutually accepts also the multiset $\{ac, ac\}$, in which the word ac appears twice. On the other hand, A cannot mutually accept $\{ab, aac\}$, as the accepting runs on both words traverse the selfloop q_0, whose capacity is 1.

As the example above demonstrates, the utilization semantics enables reasoning about the utilization of systems with consumable resources. Its applications depend on the setting modeled by the CA. If, for example, the CA models a communication network, with transitions corresponding to channels and ca-

[1]Not to be confused with *finite capacity automata* [12], which model the control of an automated manufacturing system, and are more related to Petri nets.

pacities corresponding to bounds on the number of times a channel may be used, then the CA accepts multisets of communication routes that can be transmitted simultanously in the network. Likewise, if the CA models a production system, then it accepts multisets of chains of services that can be processed mutually in the system.

The study of the utilization semantics in [10] focuses on the *maximal utilization* problem for CA: Given a CA A, return a multiset S of words, such that A mutually accepts all the words in S, and $|S|$ is maximal. The max-utilization problem can be viewed as a generalization of the max-flow problem in networks [5]. In the max-flow problem, the network is utilized by units of flow, each routed from the source to the target. The CA model enables a rich description of the feasible routes. The labels along a path correspond to a sequence of applications of resources. In particular, paths from an initial state to a final state correspond to feasible such sequences, and the goal is to mutually process as many of them as possible. It is shown in [10] that the problem can be solved in polynomial time, yet if we restrict the set of possible routes by a regular language, it becomes APX-complete, thus hard to approximate in polynomial time.

Here, we study theoretical properties of CAs as recognizers of *multi-languages*. A multi-language over an alphabet Σ is a set of multisets of words in Σ^*. The multi-language recognized by a CA A is the set $\mathcal{M}(A)$ of multisets S such that A mutually accepts S. For example the CA A from Figure 1 has in $\mathcal{M}(A)$ the multisets $\{b, b, ac, ac\}$, $\{ab, b, ac, ac\}$, and $\{b, b, aac, ac\}$, as well as all multisets contained in one of them.

We say that a multi-language $\mathcal{M}$ is *regular* if there is a CA A such that $\mathcal{M}(A) = \mathcal{M}$. We first study the expressive power of CAs, show that not all *finite* multi-languages are regular, and that nondeterministic CAs are strictly more expressive than deterministic ones (DCAs, for short). For example, there is no DCA that recongnizes the multi-language of the CA from Figure 1. We then study closure properties for CAs. In addition to the usual union and intersection operators, we consider *pairwise* union and intersection, where the operations are applied to the multisets in the multi-language. We study closure in both the nondeterministic and deterministic setting. We show that while regular multi-languages are closed under pairwise union, they are not closed under the other operators. Moreover, the deterministic fragment is not closed even under pairwise union.

Finally, we study the basic *decision problems* for CAs. We start with the membership problem, of deciding whether a given multiset is in the multi-language of a given CA. In practice, this problem is relevant for checking, for example, whether a certain list of tasks can be accomplished by a manufacturing system with bounded resources. We show that when the input is given explicitly (that is, the multiset is given by a list of its elements, and the capacities in the CA are given in unary), the problem can be solved in linear time for DCAs and is NP-complete for CAs. We continue with the containment problem, namely deciding, given CAs A and B, whether $\mathcal{M}(A) \subseteq \mathcal{M}(B)$. In practice, this problem is relevant for checking, for example, whether every multisets of routes that can be transmitted simultanously in a communication network A can also be transmitted in B. We show that the problem is EXPSPACE-complete in the general setting, going down to co-NP-complete when B is deterministic. The upper bounds in these latter results are the most technically challenging results in the paper, as they involve a careful analysis of the length of words in accepted and rejected multisets in CAs with transitions with infinite capacities.

Due to the lack of space, some proofs are omitted and can be found in the full version, at the authors' URLs.

2 Preliminaries

A *capacitated automaton* (CA, for short) [10] is a tuple $A = \langle \Sigma, Q, Q_0, \Delta, F, c \rangle$, where Σ is a finite alphabet, Q is a finite set of states, $Q_0 \subseteq Q$ is a set of initial states, $\Delta \subseteq Q \times \Sigma \times Q$ is a transition relation, $F \subseteq Q$ is a set of final states, and $c : \Delta \to \mathbb{N} \cup \{\infty\}$ is a capacity function on transitions. If $|Q_0| = 1$, and for all $q \in Q$ and $\sigma \in \Sigma$, there is at most one $q' \in Q$ such that $\langle q, \sigma, q' \rangle \in \Delta$, then we say that A is a deterministic CA (DCA, for short).

A *multiset* is a generalization of a set in which each element may appear more than once. The number of repetitions of an element is called its *multiplicity*. A multiset over a set X of elements can be represented by a list (with repetitions) of its elements or by a function $S : X \to \mathbb{N} \cup \{\infty\}$, where $S(x)$ is the multiplicity of the element $x \in X$. We focus here on multisets of words over some finite alphabet Σ, and use CAs to define and recognize such multisets. Essentially, a CA A recognizes a multiset S of words if A can accept all the words in S simultaneously without exceeding the allowed capacities. Formally, we have the following.

Let $S = \{w_1, \ldots, w_n\}$ be a (possibly infinite) multiset of finite words, with $w_i = \sigma_1^i \cdots \sigma_{k_i}^i$. An *operation* of a CA A on S is a multiset of runs $O = \{r_1, \ldots, r_n\}$, such that the following hold.

1. The operation consists of legal runs: For all $1 \leq i \leq n$, we have that $r_i = q_0^i, \ldots, q_{k_i}^i$ is a run of A on w_i: it starts in an initial state, thus $q_0^i \in Q_0$, and it obeys the transition function, thus for all $0 \leq j \leq k_i - 1$, we have that $\langle q_j^i, \sigma_{j+1}^i, q_{j+1}^i \rangle \in \Delta$.
2. The operation respects the capacities: For each run r_i, let $t_i : \Delta \to \mathbb{N}$ map each transition $e \in \Delta$ to the number of times it is traversed in r_i, thus $t_i(e) = |\{j : e = \langle q_j^i, \sigma_{j+1}^i, q_{j+1}^i \rangle\}|$. Then, the number of times each transition is traversed in all the runs in O is bounded by its capacity. Formally, for all $e \in \Delta$, it holds that $\sum_{i=1}^{n} t_i(e) \leq c(e)$.

We say that the operation O is *accepting* if the final states of all its runs are accepting, thus, $q_{k_i}^i \in F$, for all $1 \leq i \leq n$. When this happens, we say that A *mutually accepts S with the operation O*.

Note that if A is nondeterministic, it may have several operations on S. In contrast, a DCA has a single operation on each multiset. We say that A

mutually accepts the multiset S if there exists an operation O such that A mutually accepts S with O.

A *multi-language* over an alphabet Σ is a set of multisets of words from Σ^*. The multi-language recognized by a CA A is the set $\mathcal{M}(A) = \{S : A \text{ mutually accepts } S\}$ of all multisets of words in Σ^* that can be mutually accepted by A.

Example 2.1 Consider the DCA A described in Figure 2.

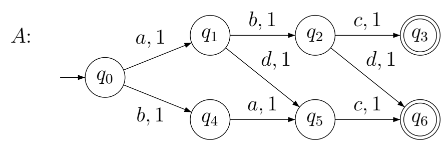

Figure 2: A DCA A with $\mathcal{M}(A) = \{\{abc, bac\}, \{abd, bac\}, \{abc\}, \{abd\}, \{adc\}, \{bac\}, \emptyset\}$.

Consider the runs $r_0 = q_0, q_4, q_5, q_6$ and $r_1 = q_0, q_1, q_2, q_3$ of A on the words bac and abc, respectively. Both runs respect the capacity function and end in an accepting state. Since the runs are disjoint, the multiset of runs $\{r_0, r_1\}$ is a legal operation, it accepts the multiset $\{abc, bac\}$, and so $\{abc, bac\} \in \mathcal{M}(A)$

Consider now the run $r_2 = q_0, q_1, q_5, q_6$ of A on adc. It respects the capacity function and ends in an accepting state, and so $\{adc\} \in \mathcal{M}(A)$. The run r_2 shares the transition $e = \langle q_5, c, q_6 \rangle$ with r_0. This transition has capacity 1. Thus, the multiset $\{r_0, r_2\}$ does not respect the capacity function and is not a legal operation, and so $\{bac, adc\} \notin \mathcal{M}(A)$.

Using similar considerations, it is easy to see that $\mathcal{M}(A) = \{\{abc, bac\}, \{abd, bac\}, \{abc\}, \{abd\}, \{adc\}, \{bac\}, \emptyset\}$. □

3 Expressive Power

A multi-language $\mathcal{M}$ is *regular* if there is a CA that recognizes $\mathcal{M}$. We denote by MREG the classes of regular multi-languages. In this section we study the expressive power of CAs. We show that not all finite multi-languages are regular, that nondeterministic CAs are strictly more expressive than deterministic one, and that the picture of closure properties in MREG is involved.

We first need some definitions and observations. Given two multisets S_1 and S_2 over a set X of elements, we say that S_1 is a *submultiset* of S_2, denoted by $S_1 \subseteq S_2$, if every element in X appears in S_1 (weakly) fewer times than in S_2. That is, for every element $x \in X$, it holds that $S_1(x) \leq S_2(x)$. Since a submultiset of an accepting operation is an accepting operation, we have the following.

Theorem 3.1 *Regular multi-languages are closed downwards: Consider a regular multi-language $\mathcal{M}$ and a multiset $S \in \mathcal{M}$. For all $S' \subseteq S$, it holds that $S' \in \mathcal{M}$.*

Proof: Consider a regular multi-language $\mathcal{M}$, a multiset $S = \{w_1, \ldots, w_n\} \in \mathcal{M}$, and a submultiset $S' \subseteq S$. Both S and S' may be infinite. Let $n' = |S'|$. Let A be a CA that recognizes $\mathcal{M}$, and $O = \{r_1, \ldots, r_n\}$ be an operation that mutually accepts S. We know that $S' \subseteq S$, so there are some indices $j_1, \ldots, j_{n'} \in \{1, \ldots, n\}$ such that $S' = \{w_{j_1}, \ldots, w_{j_{n'}}\}$. The multiset $O' = \{r_{j_1}, \ldots, r_{j_{n'}}\}$ is an accepting operation on the multiset S'. Thus, $S' \in \mathcal{M}(A) = \mathcal{M}$. □

Given a CA A, we say that a multiset S *saturates* A if S is mutually accepted by A and it is maximal with respect to containment in $\mathcal{M}(A)$. That is, $S \in \mathcal{M}(A)$, and for every $S' \in \mathcal{M}(A)$, it holds that $S \not\subset S'$. Theorem 3.1 implies that regular multi-languages are characterized uniquely by maximal multisets, which are the saturating multisets of the CA. Accordingly, we define the *saturating language* of a CA A, denoted $\mathcal{SM}(A)$, as the set of all saturating multisets of the CA.

Given a set $\mathcal{M}$ of multisets, we denote by $sub(\mathcal{M})$ the set of all submultisets of the multisets in $\mathcal{M}$. Formally, $sub(\mathcal{M}) = \{S' : \text{there exists } S \in \mathcal{M} \text{ such that } S' \subseteq S\}$. Note that for every CA A, we have that $\mathcal{M}(A) = sub(\mathcal{SM}(A))$.

Example 3.2 Recall the CA A appearing in Figure 2. It is easy to see that $\mathcal{SM}(A) = \{\{abc, bac\}, \{abd, bac\}, \{adc\}\}$. □

3.1 Regularity

Given a CA A, the *language* of A, denoted $L(A)$, is the set of all words that can be accepted by A while respecting the capacity function. It is easy to see that for every word $w \in \Sigma^*$, we have that $w \in L(A)$ iff $\{w\} \in \mathcal{M}(A)$. As we shall see in Section 3.3 (specifically, Corollary 3.12), this implies that for every finite multiset S of words, there is a CA A with $\mathcal{M}(A) = sub(\{S\})$. Essentially, the proof is similar to the proof showing that all finite languages are regular: a nondeterministic automaton for a finite language $L \subseteq \Sigma^*$ may consist of $|L|$ components, each for a word in L. Likewise, a CA for $sub(\{S\})$ may consist of $|S|$ components, each for a word in S. On the other hand, as we show below, there are finite multi-languages that are not regular.

Theorem 3.3 *Not all finite multi-languages are regular.*

Proof: We prove that the finite multi-language $sub(\{\{ab, ac\}, \{ad\}\}) = \{\{ab, ac\}, \{ad\}, \{ab\}, \{ac\}, \emptyset\}$ is not regular.

Assume by way of contradiction that there is a CA $A = \langle \Sigma, Q, Q_0, \Delta, F, c\rangle$ such that $\mathcal{M}(A) = sub(\{\{ab, ac\}, \{ad\}\})$. Thus, there is an accepting operation $O_1 = \{r_1, r_2\}$ of A on $\{ab, ac\}$, and an accepting operation $O_2 = \{r_3\}$ of A on $\{ad\}$. By definition, we know that r_1, r_2, r_3 are all accepting runs.

For all $i \in \{1, 2, 3\}$, let $r_i = q_0^i, q_1^i, q_2^i$. We distinguish between two cases. In the first case, we assume that the first transitions in the three runs coincide. That is, $q_0^1 = q_0^2 = q_0^3$ and $q_1^1 = q_1^2 = q_1^3$. Then, as O_1 respects the capacity function, it must be that $c(\langle q_0^1, a, q_1^1\rangle) \geq 2$. Since the letters read during their traversal are different, the second transitions traversed in the runs r_1 and r_3 are different from each other, and they also differ from the first transition. Thus, the fact the runs r_1 and r_3 are both legal and accepting implies that the multiset $\{r_1, r_3\}$ is a legal and accepting operation of A. However, it mutually accepts the multiset $\{ab, ad\}$. Yet, $\{ab, ad\} \notin \mathcal{M}(A)$, and we have reached a contradiction.

In the second case, the first transitions in r_1, r_2 and r_3 do not coincide. Hence, the first transition in r_3 is different from at least one of the first transitions in r_1 and r_2. Assume w.l.o.g that it is different from the first transition in r_2. Since the letters read during their traversal are different, the second transitions in the runs r_2 and r_3 are different from each other, and they also differ from the first transition. Hence, all transitions in r_2 and r_3 are different. It follows that r_2 and r_3 mutually respect the capacity function. In addition, as they are both legal and accepting runs, it follows that the multiset $\{r_2, r_3\}$ is a legal and accepting operation of A. However, it mutually accepts the multiset $\{ac, ad\}$. Yet, $\{ac, ad\} \notin \mathcal{M}(A)$, and we have reached a contradiction.

Thus, there is no A such that $\mathcal{M}(A) = sub(\{\{ab, ac\}, \{ad\}\})$, and so $sub(\{\{ab, ac\}, \{ad\}\}) \notin$ MREG. □

3.2 Determinism

A *deterministic regular multi-language* is a multi-language that can be recognized by a DCA. We denote by DMREG the class of deterministic regular multi-languages.

Theorem 3.4 *CAs are strictly more expressive than DCAs.*

Proof: Clearly, every DCA is a CA.

In order to prove strictness, consider the CA A appearing in Figure 3. Let

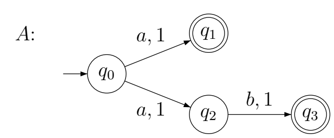

Figure 3: $\mathcal{M}(A) = sub(\{\{a, ab\}\}) \notin$ DMREG.

$r_1 = q_0, q_1$ and $r_2 = q_0, q_2, q_3$. It is easy to verify that the operation $\{r_1, r_2\}$ is legal, and that it accepts $\{a, ab\}$. Since A is saturated by this operation, we know that $\{a, ab\} \in \mathcal{SM}(A)$. In addition, there are no legal runs of A that are not in this operation, and so it must be the only saturating multiset, thus $\mathcal{SM}(A) = \{\{a, ab\}\}$. Hence, $sub(\{\{a, ab\}\}) \in$ MREG

We now claim that $sub(\{\{a, ab\}\}) \notin$ DMREG. Assume by way of contradiction that B is a DCA with $\mathcal{SM}(B) = \{\{a, ab\}\}$. Let q_0 be the initial state of B, and q_1 the only state such that $\langle q_0, a, q_1\rangle \in \Delta$. Such a state q_1 exists, as there is a legal run of B on a. It is unique, as B is deterministic. Since

$\{a, ab\} \in \mathcal{M}(B)$, the run on a must accept, and $q_1 \in F$. We also know that the runs of B on ab and on a mutually respect c, and so $c(\langle q_0, a, q_1\rangle) \geq 2$. However, this means that we can mutually accept the word a twice with the operation $\{r_1, r_1\}$. Hence, $\{a, a\} \in \mathcal{M}(B)$. However, $\{a, a\} \notin sub(\{\{a, ab\}\})$ and we have reached a contradiction. Thus, $sub(\{\{a, ab\}\}) \in$ MREG $\setminus$ DMREG, and we are done. □

Essentially, the reason no DCA can recognize $sub(\{\{a, ab\}\})$ is that the DCA should traverse twice a prefix of two different words, yet may accept only one occurrence of this prefix. Trying to generalize this to a characterization of languages in MREG$\setminus$DMREG, we say that a multi-language $\mathcal{M}$ is *prefix-replaceable* if whenever a word and its prefix are in a multiset in $\mathcal{M}$, then the multiset obtained by replacing the word by the prefix is also in $\mathcal{M}$. Formally, for every multiset of words $S \in \mathcal{M}$ and every word $w \in S$ that is a prefix of another word $w \cdot x \in S$, with $x \neq \epsilon$, it holds that $(S \setminus \{w \cdot x\}) \cup \{w\} \in \mathcal{M}$.

For example, the multi-language $sub(\{\{a, ab\}\})$ is not prefix-replaceable. Indeed, for $S = \{a, ab\}$, the word a is a prefix of the word ab, it is in S, and yet $(S \setminus \{ab\}) \cup \{a\} = \{a, a\}$ is not in $sub(\{\{a, ab\}\})$.

Theorem 3.5 *Every multi-language in* DMREG *is prefix-replaceable.*

Proof: Consider a multi-language $\mathcal{M}$. Let $S \in \mathcal{M}$ be a multiset and let $w, x \in \Sigma^*$ be words such that $x \neq \epsilon$ and $w, w \cdot x \in S$. Since $S \in \mathcal{M}$ and $\mathcal{M} \in$ DMREG, then there is a DCA A and an accepting operation O of A on S. Let $r = q_0, \ldots, q_{|w \cdot x|}$ be the run of A on $w \cdot x$. Since $w \cdot x \in S$, we know that $r \in O$. The run $r' = q_0, \ldots, q_{|w|}$ is a legal run of A on w. Moreover, since $w \in S$ and A is deterministic, the run r' is the only run of A on w, and so $q_{|w|}$ is an accepting state. The operation $O' = (O \setminus \{r\}) \cup \{r'\}$ trivially respects the capacity function, because we just remove transitions. The multiset that O' mutually accepts is $(S \setminus \{w \cdot x\}) \cup \{w\}$. Hence, $(S \setminus \{w \cdot x\}) \cup \{w\} \in \mathcal{M}$, and $\mathcal{M}$ is prefix-replaceable. □

We leave open the problem of characterizing DMREG, in particular the problem of deciding whether a given CA has an equivalent DCA. Below we refute two conjectures about a such a characterization. The first considers prefix-replaceability. The second considers determinization based on the classical subset construction.

Theorem 3.6 *Prefix-replaceability in* MREG *is not a sufficient condition for membership in* DMREG.

Proof: We describe a multi-language in MREG that is prefix-replaceable and is not in DMREG. Consider the CA A described in Figure 4.

It is easy to verify that $\mathcal{SM}(A) = \{\{a, a\}, \{a, ab\}, \{a, ac\}\}$. It is not hard to see that $\mathcal{SM}(A)$ is prefix-replaceable. Indeed, the the only multisets in $\mathcal{SM}(A)$ that contain a word and its prefix are $\{a, ab\}$ and $\{a, ac\}$, and for both the replacement of the word by the prefix result in the multiset $\{a, a\}$, which is

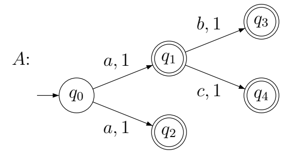

Figure 4: $\mathcal{M}(A)$ is prefix-replacable, yet $\mathcal{M}(A) \notin$ DMREG.

in $\mathcal{SM}(A)$. We argue that $\mathcal{SM}(A)$ cannot be recognized by a DCA. Indeed, a DCA for $\mathcal{SM}(A)$ has legal and accepting runs on the words ab and ac. In addition, since $\{a, a\}$ is mututally accepted, the transition from the initial state on a has capacity at least 2. Then, however, the multiset $\{ab, ac\}$ is mutually accepted, yet $\{ab, ac\} \notin \mathcal{SM}(A)$. □

The second refutation considers *powerset-typeness*. Researchers have studied typeness for automata in various setting, essentially restricting the search space for equivalent automata [7, 8]. In particular, a class γ of automata is powerset type if whenever a nondeterministic automaton A in the class γ has an equivalent deterministic automaton, then an equivalent deterministic automaton can be defined on top of the subset construction of A. It is well known, for example, that finite automata are powerset type. So are nondeterministic weak automata on infinite words [8]. On the other hand, Büchi automata are not powerset type [11].

Theorem 3.7 *CAs are not powerset type.*

Proof: Consider the CA A described in Figure 5. It is easy to see that $\mathcal{M}(A) \in$ DMREG. Indeed, the DCA D recognizes $\mathcal{M}(A)$. On the other hand, applying the subset construction on A results on the the structure A', and there is no way to define initial and final states and capacities on top of it and obtain a DCA for $\mathcal{M}(A)$. □

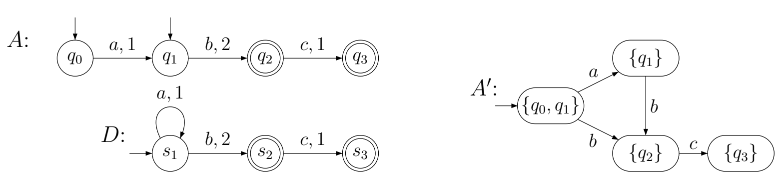

Figure 5: The CA A has an equivalent DCA D, yet no equivalent DCA can be defined on top of its subset construction A'.

3.3 Closure Properties

In this section we study closure properties for MREG and DMREG. Since all regular multi-languages are closed downwards, complementation is not interesting in the context of MREG. On the other hand, in addition to the usual union and intersection operators, we consider *pairwise* union and intersection, to be defined below.

Consider a set X of elements. The union of multisets over X is naturally defined by summing the repetitions of each element. That is, for two multistes S_1 and S_2, we define their union $S_1 \cup S_2$ such that for every element $x \in X$, we have that $(S_1 \cup S_2)(x) = S_1(x) + S_2(x)$. Then, the intersection of multisets is defined by taking the minimal number of repetitions of each element. That is, for two multistes S_1 and S_2, we define their intersection $S_1 \cap S_2$ such that for every element $x \in X$, we have that $(S_1 \cap S_2)(x) = \min\{S_1(x), S_2(x)\}$.

We continue to the pairwise operators, where we apply union and intersection between the multisets in the two sets of multisets. Formally, for two multi-languages $\mathcal{M}_1$ and $\mathcal{M}_2$, we define their pairwise union by $\mathcal{M}_1 \uplus \mathcal{M}_2 = \{S_1 \cup S_2 : S_1 \in \mathcal{M}_1 \text{ and } S_2 \in \mathcal{M}_2\}$, and their pairwise intersection by $\mathcal{M}_1 \Cap \mathcal{M}_2 = \{S_1 \cap S_2 : S_1 \in \mathcal{M}_1 \text{ and } S_2 \in \mathcal{M}_2\}$.

Example 3.8 Let $\mathcal{M}_1 = \{\{a\}, \{ab, ac\}\}$ and $\mathcal{M}_2 = \{\{ac\}, \{a, ab, ac\}$. Then,

- $\mathcal{M}_1 \cup \mathcal{M}_2 = \{\{a\}, \{ab, ac\}, \{ac\}, \{a, ab, ac\}\}$.
- $\mathcal{M}_1 \cap \mathcal{M}_2 = \emptyset$.
- $\mathcal{M}_1 \uplus \mathcal{M}_2 = \{\{a, ac\}, \{a, a, ab, ac\}, \{ab, ac, ac\}, \{ab, ac, a, ab, ac\}\}$.
- $\mathcal{M}_1 \Cap \mathcal{M}_2 = \{\emptyset, \{a\}, \{ac\}, \{ab, ac\}\}$.

When we focus on regular multi-languages, it is useful to observe that for all multisets $\mathcal{M}_1$ and $\mathcal{M}_2$, it follows directly from the definition that $sub(\mathcal{M}_1) \cup sub(\mathcal{M}_2) = sub(\mathcal{M}_1 \cup \mathcal{M}_2)$ and $sub(\mathcal{M}_1) \uplus sub(\mathcal{M}_2) = sub(S_1 \cup S_2)_{S_1 \in \mathcal{M}_1, S_2 \in \mathcal{M}_2}$. In addition, we have the following.

Lemma 3.9 *If $\mathcal{M}_1$ and $\mathcal{M}_2$ are closed downwards, then $\mathcal{M}_1 \Cap \mathcal{M}_2 = \mathcal{M}_1 \cap \mathcal{M}_2$.*

Proof: First, by definition, $\mathcal{M}_1 \cap \mathcal{M}_2 \subseteq \mathcal{M}_1 \Cap \mathcal{M}_2$ regardless of $\mathcal{M}_1$ and $\mathcal{M}_2$ being closed downwards. We prove that $\mathcal{M}_1 \Cap \mathcal{M}_2 \subseteq \mathcal{M}_1 \cap \mathcal{M}_2$. Consider a multiset $S = S_1 \cap S_2$ for $S_1 \in \mathcal{M}_1$ and $S_2 \in \mathcal{M}_2$. Clearly, $S_1 \cap S_2 \subseteq S_1$. Therefore, as $\mathcal{M}_1$ is closed downwards, we have that $S_1 \cap S_2 \in \mathcal{M}_1$. Similarly, $S_1 \cap S_2 \in \mathcal{M}_2$, and so $S_1 \cap S_2 \in \mathcal{M}_1 \cap \mathcal{M}_2$, and we are done. ☐

As MREG and DMREG are closed downwards, Lemma 3.1 implies that in the context of MREG and DMREG, pairwise intersection coincides with intersection. For union, this is not true.

We can now state our results about closure properties.

Theorem 3.10 DMREG *and* MREG *are not closed under union.*

Proof: Consider the DCAs A_1 and A_2 appearing in Figure 6.

It is easy to see that $\mathcal{SM}(A_1) = \{\{ab, ac\}\}$ and $\mathcal{SM}(A_2) = \{\{ad\}\}$. Therefore, both multi-languages $sub(\{\{ab, ac\}\})$ and $sub(\{\{ad\}\})$ are in DMREG. However, their union $sub(\{\{ad\}\}) \cup sub(\{\{ab, ac\}\}) = sub(\{\{ad\}, \{ab, ac\}\})$ is not regular, as we have seen in the proof of Theorem 3.3. ☐

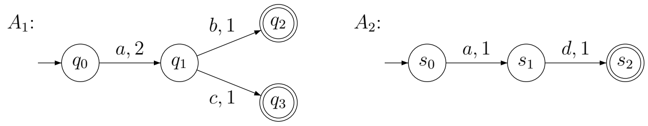

Figure 6: The DCAs A_1 and A_2, with $\mathcal{M}(A_1) \cup \mathcal{M}(A_2)$ not in MREG.

Theorem 3.11 MREG *is closed under pairwise union.*

Proof: Given two CAs A_1 and A_2, let A be the CA obtained by putting A_1 and A_2 "side-by-side", as in the union construction for NFAs. We prove that $\mathcal{M}(A_1) \uplus \mathcal{M}(A_2) = \mathcal{M}(A)$.

We first prove that $\mathcal{M}(A_1) \uplus \mathcal{M}(A_2) \subseteq \mathcal{M}(A)$. Consider a multiset $S \in \mathcal{M}(A_1) \uplus \mathcal{M}(A_2)$. By definition, there exist $S_1 \in \mathcal{M}(A_1)$ and $S_2 \in \mathcal{M}(A_2)$ such that $S = S_1 \cup S_2$. Let O_1 and O_2 be accepting operations of A_1 on S_1 and of A_2 on S_2. It is easy to see that $O_1 \cup O_2$ is a legal and accepting operation of A on S. Thus, $S \in \mathcal{M}(A)$ and we are done.

For the other direction, consider a multiset $S \in \mathcal{M}(A)$. Let O be the accepting opeartion of A on S. Since A is obtained by putting A_1 and A_2 side-by-side, every run of A on a word must be a run of either A_1 or A_2, and cannot contain transitions from both automata. Let O_1 and O_2 be the parition of O into runs of A_1 and A_2, respectively. Since O is a legal and accepting operation, O_1 must respect the capacity function of A_1. Thus, it is a legal and accepting operation of A_1. Similarly, O_2 is a legal and accepting operation of A_2. Let $S_1 \in \mathcal{M}(A_1)$ and $S_2 \in \mathcal{M}(A_2)$ be the multisets accepted by O_1 and O_2, respectively. Since $O = O_1 \cup O_2$ it holds that $S = S_1 \cup S_2$. Hence, $S \in \mathcal{M}(A_1) \uplus \mathcal{M}(A_2)$ and $\mathcal{M}(A) \subseteq \mathcal{M}(A_1) \uplus \mathcal{M}(A_2)$. □

Consider a finite multiset S. As discussed above, for a single word $w \in S$, it is easy to define a DCA that mutually accepts $\{\{w\}, \emptyset\}$. By Theorem 3.11, we can put these DCAs side-by-side and obtain a CA that accepts their pairwise union, namely $\biguplus_{w \in S} \{\{w\}, \emptyset\} = sub(\{S\})$. Thus, $sub(\{S\}) \in$ MREG. Hence, we have following.

Corollary 3.12 *For every finite multiset S, we have that $sub(\{S\}) \in$ MREG.*

The "side-by-side" construction introduces nondeterminism. As we show now, this is inevitable.

Theorem 3.13 DMREG *is not closed under pairwise union.*

Proof: Consider the DCAs A_1 and A_2 appearing in Figure 7.

It is easy to see that $\mathcal{SM}(A_1) = \{\{a\}\}$ and $\mathcal{SM}(A_2) = \{\{ab\}\}$. Therefore, $sub(\{\{a\}\})$ and $sub(\{\{ab\}\})$ are both in DMREG. However, their pairwise union $sub(\{\{a\}\}) \uplus sub(\{\{ab\}\}) = sub(\{\{a, ab\}\})$ is not in DMREG, as we have seen in the proof of Theorem 3.4. □

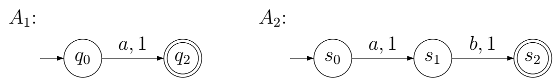

Figure 7: The DCAs A_1 and A_2, with $\mathcal{M}(A_1) \uplus \mathcal{M}(A_2)$ not in DMREG.

Theorem 3.14 MREG *is not closed under intersection.*

Proof: Consider the CAs A_1 and A_2 appearing in Figure 8.

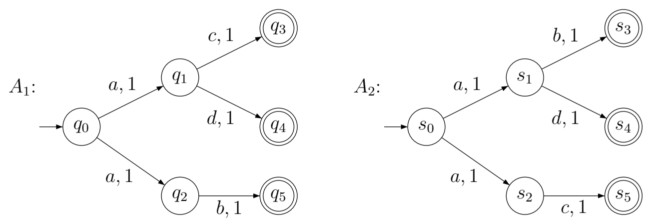

Figure 8: The CAs A_1 and A_2, with $\mathcal{M}(A_1) \cap \mathcal{M}(A_2)$ not in MREG.

It is easy to verify that $\mathcal{SM}(A_1) = \{\{ab, ac\}, \{ab, ad\}\}$ and $\mathcal{SM}(A_2) = \{\{ab, ac\}, \{ac, ad\}\}$. Hence, $sub(\{\{ab, ac\}, \{ab, ad\}\})$ and $sub(\{\{ab, ac\}, \{ac, ad\}\})$ are in MREG. However, their intersection $\mathcal{M}(A_1) \cap \mathcal{M}(A_2) = sub(\{\{ab, ac\}, \{ad\}\})$ is not regular, as we have seen in the proof of Theorem 3.3. □

Theorem 3.15 DMREG *is not closed under intersection.*

Proof: Consider the DCAs A_1 and A_2 described in Figure 9.

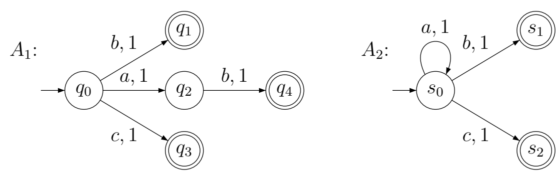

Figure 9: The DCAs A_1 and A_2, with $\mathcal{M}(A_1) \cap \mathcal{M}(A_2)$ not in DMREG.

It is easy to verify that $\mathcal{SM}(A_1) = \{\{ab, b, c\}\}$. For A_2, we can accept one word for each accepting state by reading b and c, and we can choose to traverse the self-loop in s_0 before either of these letters, or not traverse it at all. Overall, we get that we can accept $\{b, c\}$, $\{ab, c\}$, or $\{b, ac\}$, and their submultisets. Thus, $\mathcal{SM}(A_2) = \{\{b, c\}, \{ab, c\}, \{b, ac\}\}$. It follows that $sub(\{\{ab, b, c\}\})$ and $sub(\{\{b, c\}, \{ab, c\}, \{b, ac\}\})$ are both in DMREG. We prove that their intersection $sub(\{\{b, c\}, \{ab, c\}\})$ is not in DMREG.

Assume by way of contradiction that there is a DCA A with $\mathcal{SM}(A) = \{\{b, c\}, \{ab, c\}\}$. Let q_0 be its initial state, and q_1 the unique state such that

$\langle q_0, a, q_1 \rangle \in \Delta$. Such a state q_1 exists, as there is a legal run of A on ab. It is unique, as A is determinsitic. We distinguish between two cases. In the first case, the transition $\langle q_0, a, q_1 \rangle$ is a self-loop, that is, $q_1 = q_0$. Since $\{c\} \in \mathcal{M}(A)$, the latter implies that after reading a we can also read c. This implies that $\{ac\} \in \mathcal{M}(A)$, and we have reached a contradiction. In the second case, we have that $q_1 \neq q_0$. Then, we know that the second transition in the run of A on ab is different from the single transition in the run on b. This implies that these runs are disjoint. Since both are accepting runs that respect the capacity function, we get that $\{b, ab\} \in \mathcal{M}(A)$, which is again a contradiction.

Hence, $sub(\{\{b, c\}, \{ab, c\}\}) \notin$ DMREG, and we are done. □

4 Decision Problems

In this section we study the following decision problems for CAs in the utilization semantics:

1. Membership: given a CA A and a finite multiset S, decide whether $S \in \mathcal{M}(A)$.
2. Containment: given two CAs A and B, decide whether $\mathcal{M}(A) \subseteq \mathcal{M}(B)$.

Remark 4.1 A classical decision problem for automata is *nonemptiness*, namely deciding whether their language is not empty. In the setting of CA, we need to decide, given a CA A, whether $\mathcal{M}(A) \neq \emptyset$. Since all regular multi-languages are nonempty, as they contain $\emptyset$, this question is not of much interest. Alternatively, one may ask, in the nonemptiness problem for CA, whether $\mathcal{M}(A) \neq \{\emptyset\}$. It is easy to see that the latter holds iff the language of A is not empty, which is NLOGSPACE-complete [10]. □

Studying the complexities of decision problems on CAs, it is important to specify how the input to the problems is given. For a multiset S of words, we define the *length* of S, denoted $\|S\|$, as the sum of lengths of words in S. We also refer to the *size* of S, denoted $|S|$, which is the number of words in S. Alternative definitions represent a multiset by a list of its words along with their multiplicity, in unary or binary. Note that in either case, the "list with multiplicity" representation is more succinct than our "list with repetition" representation. For a CA $A = \langle \Sigma, Q, Q_0, \Delta, F, c \rangle$, we define the size of A by $\sum_{e \in \Delta} c'(e)$, where $c'(e)$ is $c(e) + 1$ if $c(e) \neq \infty$, and is 1 if $c(e) = \infty$. Note that our definition corresponds to a representation of A with capacities given in unary.

For traditional automata, the membership problem is to decide, given a word w and an NFA or DFA A, whether $w \in L(A)$. In both cases, the problem can be solved in linear time and is NLOGSPACE-complete. For the traditional semantics of CAs, namely when we care about $L(A)$, the membership problem can be solved in linear time for DCAs and is NP-complete for CAs [10]. For the containment problem, the complexity depends on whether the containing automaton is deterministic. For a CA A, the complexity of deciding whether $L(A) \subseteq L(B)$ is co-NP-complete for a DCA B and is EXPSPACE-complete for a CA B [9].

We now study the complexity of the problems for CAs in the utilization semantics. We start with membership.

Theorem 4.1 *The membership problem in the utilization semantics can be solved in linear time for DCAs and is NP-complete for CAs.*

Proof: Given a DCA A and a finite multiset S, we trace the single run of A on each word in S and maintain for each transition a counter of the number of times it is traversed. Clearly, $S \in \mathcal{M}(A)$ iff all runs end in an accepting state, and the counters are bounded by the corresponding capacities.

For a CA A and a finite multiset S, a witness to the membership of S in $\mathcal{M}(A)$ is an operation that accepts S. The length of the operation agrees with that of S, and as in the case of DCAs, it can be checked in linear time.

For the lower bound, recall that given a CA A and a word w, we have that $w \in L(A)$ iff $\{w\} \in \mathcal{M}(A)$. Thus, the lower bound follows from the NP-hardness of the membership problem in the traditional semantics for CAs [10]. $\square$

We continue to the containment problem. Recall that $\mathcal{M}(A)$ and $\mathcal{M}(B)$ may be infinite and may contain infinite multisets, which makes the setting challenging. Indeed, if all capacities are finite, then all multisets accepted by a CA A are of length linear in A. As we now show, we are able to bound the length of a multiset in $\mathcal{M}(A) \setminus \mathcal{M}(B)$ even when both may contain transitions with an infinite capacity. Intuitively, long words must traverse cycles all whose transitions have infinite capacity, and can be shortened.

In order to formalize this intuition, we first need some notations.

Consider a CA $A = \langle \Sigma, Q, Q_0, \Delta, F, c \rangle$. For a capacity function $c : \Delta \to \mathbb{N} \cup \{\infty\}$, let $c_\downarrow$ be the set of capacity functions obtained by closing c downwards. Formally, a function $c' : \Delta \to \mathbb{N} \cup \{\infty\}$ is in $c_\downarrow$ if for all transitions $e \in \Delta$ with $c(e) = \infty$, we have $c'(e) = \infty$, and for all transitions $e \in \Delta$ with $c(e) \in \mathbb{N}$, we have $0 \leq c'(e) \leq c(e)$. It is easy to see that the size of $c_\downarrow$, denoted $|c_\downarrow|$, is $\Pi_{e:c(e)\in\mathbb{N}^+}(c(e)+1)$, and is thus exponential in the size of the CA.

It is shown in [10] that, when following the traditional semantics, the CA A can be translated to an equivalent NFA A'; that is, $L(A) = L(A')$, and all transitions in A' have capacity ∞. The state space of A' is $Q \times c_\downarrow$, where a pair $\langle q, c' \rangle$ indicates it is possible to reach the state q by "consuming" capacities that reduce the initial capacity function to c'. We refer to a function $c' \in c_\downarrow$ as a *residual capacity function*, as it specifies the capacities that are "leftovers" of these consumed so far. Accordingly, the NFA A' contains a transition $\langle \langle q_1, c_1 \rangle, \sigma, \langle q_2, c_2 \rangle \rangle$ iff $\Delta(q_1, \sigma, q_2)$, $c_1(\langle q, \sigma, q' \rangle) > 0$, and c_2 is obtained from c_1 by taking the traversal of $\langle q_1, \sigma, q_2 \rangle$ into an account by reducing its capacity by 1. Formally, $c_2(e) = c_1(e)$ for all $e \neq \langle q_1, \sigma, q_2 \rangle$, and $c_2(\langle q_1, \sigma, q_2 \rangle) = c_1(\langle q_1, \sigma, q_2 \rangle) - 1$. Note that cycles in A' are possible only along cycles in A all whose transitions have capacity ∞. By determinizing A', we get a DFA D equivalent to A.[2] The size of D is exponential in $|Q|$ and $|c_\downarrow|$, and so it is doubly-exponential in $|A|$.

Lemma 4.2 *Consider CAs A and B. If $\mathcal{M}(A) \not\subseteq \mathcal{M}(B)$, then there is a finite multiset in $\mathcal{M}(A) \setminus \mathcal{M}(B)$ whose size is linear in the size of B and whose length is polynomial in A and doubly exponential in B in the general case, and polynomial in both A and B when B is a DCA.*

Proof: Let $A = \langle \Sigma, Q, Q_0, \Delta, F, c\rangle$ and $B = \langle \Sigma, Q', Q'_0, \Delta', F', c'\rangle$. As $\mathcal{M}(A) \not\subseteq \mathcal{M}(B)$, there is a multiset $S \in \mathcal{M}(A) \setminus \mathcal{M}(B)$. Since $\mathcal{M}(B)$ is closed downwards and $\emptyset \in \mathcal{M}(B)$, there is a submultiset $S' \subseteq S$ such that $S' \notin \mathcal{M}(B)$ and for every $S'' \subset S'$, it holds that $S'' \in \mathcal{M}(B)$. That is, S' is a minimal submultiset of S that is not in $\mathcal{M}(B)$. Note that since $S' \subseteq S$ and $S \in \mathcal{M}(A)$, we have that $S' \in \mathcal{M}(A)$. Hence, $S' \in \mathcal{M}(A) \setminus \mathcal{M}(B)$.

We claim that S' is of a finite size, linear in B. Since S' is a minimal submultiset of S that is not in $\mathcal{M}(B)$, it does not contain words that can be accepted via a run that only uses transitions with infinite capacities. To see this, assume by way of contradiction that S' contains a word w that is read in B along a path π from Q'_0 to F' all whose transitions have capacity ∞. By the minimality of S', we have that $S' \setminus \{w\} \in \mathcal{M}(B)$. But then, by adding to the operation that mutually accepts $S' \setminus \{w\}$ an accepting run on w along the path π, we obtain an operation that respects the capacities and mutually accepts S', contradicting the fact that $S' \notin \mathcal{M}(B)$. Now, consider a word $w \in S'$ and consider the operation O of B that mutually accepts $S' \setminus \{w\}$. Since S' does not contain words that can be accepted via a run that only uses transitions with infinite capacities, we know that every run in O consumes at least 1 from the capacity of some transition. Hence, $|S' \setminus \{w\}| = |S'| - 1 \leq |B|$, and we are done.

We continue to the length argument and show that we can replace every word in S' by a word whose length is polynomial in A and doubly exponential in B in the general case, and is polynomial in both A and B when B is a DCA, while maintaining that $S' \in \mathcal{M}(A) \setminus \mathcal{M}(B)$. Recall that $S' = \{w_1, \ldots, w_n\}$. For $i = 1, 2, \ldots, n$, we proceed iteratively and replace w_i by w'_i, defined so that $S'' = S' \cup \{w'_i\} \setminus \{w'_i\}$ still satisfies $S'' \in \mathcal{M}(A) \setminus \mathcal{M}(B)$. Also, if S'' is no longer minimal, we take a minimal submultiset of S'', namely one for which $S'' \setminus \{w\} \in \mathcal{M}(B)$ for all $w \in S''$. Once this is done, we update S' to S'', and continue to the next word in S'.

It is left to point to w'_i. First, if w_i is of the required length, then $w'_i = w_i$. Otherwise, we distinguish between the case B is nondeterministic and the case it is deterministic.

We start with the case B is nondeterministic. Let B' be a DFA with $L(B') = L(B)$. As detailed above, we can define B' with state space $2^{Q' \times c'_\downarrow}$. Consider the product $P = A \times B'$. Note that P is polynomial in A and doubly exponential in B. Let O be an operation with which A mutually accepts S', and let r_i be the accepting run of A on w_i in O. We obtain w'_i from w_i by removing subwords

[2]Not to be confused with determinization in the utilization semantics, which is impossible (Theorem 3.4 here).

that traverse cycles in the path induced by r_i in the product P. Note that since B' is deterministic, there is a single such path. Since words that are longer than the number of states in P must traverse a cycle in it, we end up with a word w'_i of the desired length.

Let $S'' = S' \cup \{w'_i\} \setminus \{w_i\}$. We claim that $S'' \in \mathcal{M}(A) \setminus \mathcal{M}(B)$. First, $S'' \in \mathcal{M}(A)$, as we can replace r_i in O by the run r'_i obtained from r_i by removing the sub-runs along the sub-words of w_i that we removed in the transition to w'_i. Since these sub-runs are cycles in A, and we only reduce the consumption of capacities, the run r'_i accepts w'_i, and the new operation mutually accepts S''. Second, $S'' \notin \mathcal{M}(B)$. To see this, assume by way of contradiction that $S'' \in \mathcal{M}(B)$. Let O' be an operation of B that mutually accepts S'', and let r'_i be that accepting run on w'_i in O'. We claim we can obtain from r'_i an accepting run r_i of B on w_i such that the consumption of transitions in r_i is bounded by that in r'_i. Thus, the operation of B obtained from O' by replacing r'_i by r_i mutually accepts S', contradicting the fact $S' \notin \mathcal{M}(B)$. Recall that the cycles we remove obtaining w'_i from w_i traverse a cycle in B'. Consider a cycle from a state $s = \{\langle q_1, c_1\rangle, \ldots, \langle q_k, c_k\rangle\}$ in B'. Recall that for all $1 \leq j \leq k$, we have that $q_j \in Q'$ and c_j is a residual capacity function in $c'_{\downarrow}$, specifying the capacities that are left after q_j has been reached.

Let $y \in \Sigma^*$ be the word read along the cycle, and let $h \in \Sigma^*$ be a prefix of w'_i (and w_i) that is read before the cycle y is reached. Let $\langle q_j, c_j\rangle \in s$ be such that r'_i reaches the state q_j with a residual capacity function c_j after reading h. By the definition of B', there is some $1 \leq l \leq k$ such that from the state q_l with residual capacity function c_l, the CA B can read y and reach the state q_j with residual capacity function c_j. Since the traversal of y can only consume capacities, we know that for every transition $e \in \Delta'$, we have that $c_j(e) \leq c_l(e)$. Recall that in the process of defining w'_i, we replace the subword $h \cdot y$ by h. Now, in the process of defining the run r_i, we change the prefix of the run r'_i to reach the state q_l with residual capacity function c_l after reading h (note that since $\langle q_l, c_l\rangle \in s$, this is possible), and then reach the state q_j with residual capacity function c_j after reading $h \cdot y$. From there, r_i continues as r'_i. The run r_i is obtained by repeating the above procedure for all removed cycles.

We continue to the case B is deterministic. Here, reasoning about the product $A \times B$ is not sufficient, and we have to be more careful. As S' is minimal and B is deterministic, the operation $O_i = O \setminus \{r_i\}$ of B mutually accepts $S' \setminus \{w_i\}$. Let B_i denote the CA obtained from B by reducing the capacities of the transitions according to their traversals in O_i. That is, for each occurrence of a transition in O_i, we reduce by 1 its capacity in B_i. Since O_i respects the capacity function, the obtained capacities in B_i are all non-negative.

Let $O^A = \{r^A_1, \ldots, r^A_n\}$ be an accepting operation of A on S', with r^A_i being the accepting run on w_i. Let A_i be the CA obtained from A by reducing the capacities of the transitions used in $O^A \setminus \{r^A_i\}$. That is, for each occurrence of a transition in $r^A_1, r^A_2, \ldots, r^A_{i-1}, r^A_{i+1}, \ldots, r^A_n$, we reduce by 1 its capacity in A_i. We claim that $w_i \in L(A_i) \setminus L(B_i)$. First, as O^A mutually accepts S' in A and in the transition to A_i we reduced capacities only according to runs in $O^A \setminus \{r^A_i\}$, then r^A_i is a legal and accepting run of w_i in A_i. In addition, if we assume by way of contradiction that $w_i \in L(B_i)$, then by adding the accepting run of B_i on w_i to O_i, we get an operation of B that is legal and mutually accepts S', contradicting the fact that $S' \notin \mathcal{M}(B)$. Hence, $w_i \in L(A_i) \setminus L(B_i)$, and so $L(A_i) \not\subseteq L(B_i)$.

It is shown in [9] that if $L(A_i) \not\subseteq L(B_i)$, then there is a word of length polynomial in the size of A_i and B_i that is in $L(A_i) \setminus L(B_i)$. We define w'_i to be such a word. Since the capacities of A_i and B_i are bounded by the capacities of A and B respectively, the length of w'_i is at most polynomial also with respect to the sizes of A and B. Let $S'' = S' \cup \{w'_i\} \setminus \{w_i\}$. We prove that $S'' \in \mathcal{M}(A) \setminus \mathcal{M}(B)$. First, since $w'_i \in L(A_i)$, we can replace the accepting run on w_i in O^A with the run on w'_i and get that $S'' \in \mathcal{M}(A)$. In addition, we argue that since $w'_i \notin L(B_i)$, then $S'' \notin \mathcal{M}(B)$. Indeed, since B is deterministic, there is a unique run of B on w'_i. Since $w'_i \notin L(B_i)$, this run in B_i must be not accepting or not legal. If it is not accepting, then it is not accepting in B too. If it is not legal, then it must not respect the capacity function of B_i. Hence, adding this run to the operation O_i must not respect the capacity function of B. It follows that $S'' \notin \mathcal{M}(B)$. □

Lemma 4.3 *The containment problem for CAs in the utilization semantics is at least as hard as the containment problem for CAs in the traditional semantics.*

Proof: We describe a logspace reduction from the containment problem in the traditional semantics to the containment problem in the utilization semantics. Consider a CA $A = \langle \Sigma, Q, Q_0, \Delta, F, c\rangle$, and let \$ be a letter not in Σ. We define the \$\$-*padding* of A as the CA $A' = \langle \Sigma \cup \{\$\}, Q \cup \{q_0, q_1\}, \{q_0\}, \Delta', F, c'\rangle$, where Δ' and c' are defined as follows.

- The transition relation Δ' is obtained from Δ by adding one \$-transition from q_0 to q_1 and $|Q_0|$ \$-transitions from q_1 to all initial states in A. That is, $\Delta' = \Delta \cup \{\langle q_0, \$, q_1\rangle\} \cup \{\langle q_1, \$, q'\rangle \,:\, q' \in Q_0\}$.
- The capacity of all new transitions is 1, and the capacity of the transitions in Δ stays as in A. That is, $c' : \Delta' \to \mathbb{N} \cup \{\infty\}$ is such that $c'(e) = c(e)$ if $e \in \Delta$, and $c'(e) = 1$ otherwise.

Note that the construction preserves determinism: the CA A' has a single initial state and there is nondeterminism in the transitions from q_1 only when $|Q_0| > 1$.

Now, given CAs A and B – an input to the containment problem in the traditional semantics, our reduction returns their \$\$-paddings CAs A' and B'. Clearly, the reduction can be done in logspace. We prove that the reduction is correct, thus $L(A) \subseteq L(B)$ iff $\mathcal{M}(A') \subseteq \mathcal{M}(B')$.

First note that $L(A') = \$\$ \cdot L(A)$. Indeed, every run of A' is of the form q_0, q_1, r, for a run r of A, and the first two transitions in it can only be traversed while reading \$\$. Similarly, $L(B') = \$\$ \cdot L(B)$, and thus we get that $L(A) \subseteq L(B)$ iff $L(A') \subseteq L(B')$.

Now, since the only transition from the initial state has capacity 1, all multisets in A' contain at most one word. Since, in addition, we know that for every $w \in (\Sigma \cup \{\$\})^*$, we have that $w \in L(A')$ iff $\{w\} \in \mathcal{M}(A')$, it follows that $\mathcal{M}(A') = \{\emptyset\} \cup \{\{w\} : w \in L(A')\}$. Likewise, $\mathcal{M}(B') = \{\emptyset\} \cup \{\{w\} : w \in L(B')\}$. Accordingly, $\mathcal{M}(A') \subseteq \mathcal{M}(B')$ iff $L(A') \subseteq L(B')$ iff $L(A) \subseteq L(B)$, and we are done. □

Theorem 4.4 *The containment problem $\mathcal{M}(A) \subseteq \mathcal{M}(B)$ is EXPSPACE-complete for a CA or a DCA A and a CA B, and is co-NP-complete for a CA or a DCA A and a DCA B.*

Proof: We start with the upper bounds. By Lemma 4.2, if $\mathcal{M}(A) \not\subseteq \mathcal{M}(B)$, then there is a finite multiset $S \in \mathcal{M}(A) \setminus \mathcal{M}(B)$ such that the size of S is linear in B and its length is polynomial in A and doubly-exponential in B in the general case, and is polynomial in both A and B in the case B is a DCA. In the case B is a DCA, the multiset S and the accepting operation of A on S, which is of the same length as S, are a polynomial witness for $\mathcal{M}(A) \not\subseteq \mathcal{M}(B)$. Verifying that the operation mutually accepts S takes linear time. In addition, by Theorem 4.1, checking that $S \notin \mathcal{M}(B)$ takes linear time as well. It follows that the containment problem for a CA in a DCA is in co-NP.

We continue to the case B is a CA. Let $A = \langle \Sigma, Q, Q_0, \Delta, F, c \rangle$ and $B = \langle \Sigma, Q', Q'_0, \Delta', F', c' \rangle$. Intuitively, the EXPSPACE algorithm (in fact, NEXPSPACE, yet NEXPSPACE = EXPSAPCE [13]), guesses a witness multiset $S \in \mathcal{M}(A) \not\subseteq \mathcal{M}(B)$. It maintains the residual capacity function of A while it operates to mutually accept S, namely some $c^A \in c_{\downarrow}$ that corresponds to capacities used for processing the guessed multiset via guessed runs. It also maintains the set of all possible residual capacity functions that an operation of B may induce when it operates to mutually accept S, namely a set $\{c_1^B, \ldots, c_m^B\}$, with $c_i^B \in c'_{\downarrow}$. Note that $m \leq |c'_{\downarrow}|$ and is thus exponential in B. Initially, $c^A = c$, $m = 1$, and $c_1^B = c'$. In each iteration, the algorithm guesses a word w and a run of A on w, it updates c^A according to the consumption of transitions in this run, and it goes over all the possible runs of B on w and duplicates each of the existing residual capacity functions c_i^B to reflect all these possible runs. Accordingly, at the end of each iteration, the algorithm has one residual capacity function for A and at most exponentially many residual capacity functions for B. If c^A is positive (that is $c^A(e) \geq 0$, for all $e \in \Delta$) and all c_i^B's have a negative capacity (that is $c_i^B(e) < 0$, for some $e \in \Delta'$), the algorithm stops and accepts. Indeed, the guessed multiset is in $\mathcal{M}(A) \setminus \mathcal{M}(B)$. Otherwise, the algorithm continues to the next word. After linearly many rounds, if no bad configuration as above has been encountered, it stops and rejects. Indeed, by Lemma 4.2, if $\mathcal{M}(A) \not\subseteq \mathcal{M}(B)$, then at least one run should accept.

The lower bounds follow from the known lower bounds in the traditional semantics, namely EXPSPEC-hard for a CA B and co-NP-hard for a DCA B, and Lemma 4.3. □

References

[1] Boigelot, B., Godefroid, P.: Symbolic verification of communication protocols with infinite state spaces using QDDs. In: Proc. 8th Int. Conf. on Computer Aided Verification. Lecture Notes in Computer Science, vol. 1102, pp. 1–12. Springer (1996)

[2] Bouajjani, A., Habermehl, P., Vojnar, T.: Verification of parametric concurrent systems with prioritised FIFO resource management. Formal Methods in System Design **32**(2), 129–172 (2008)

[3] Cadilhac, M., Finkel, A., McKenzie, P.: On the expressiveness of Parikh automata and related models. In: 3rd Workshop on Non-Classical Models for Automata and Applications - NCMA. pp. 103–119 (2011)

[4] Comon, H., Jurski, Y.: Multiple counters automata, safety analysis and presburger arithmetic. In: Proc. 10th Int. Conf. on Computer Aided Verification. Lecture Notes in Computer Science, vol. 1427, pp. 268–279. Springer (1998)

[5] Ford, L., Fulkerson, D.: Flows in networks. Princeton Univ. Press, Princeton (1962)

[6] Klaedtke, F., Rueß, H.: Monadic second-order logics with cardinalities. In: Proc. 30th Int. Colloq. on Automata, Languages, and Programming. Lecture Notes in Computer Science, vol. 2719, pp. 681–696. Springer (2003)

[7] Krishnan, S., Puri, A., Brayton, R.: Deterministic ω-automata vis-a-vis deterministic Büchi automata. In: Algorithms and Computations. Lecture Notes in Computer Science, vol. 834, pp. 378–386. Springer (1994)

[8] Kupferman, O., Morgenstern, G., Murano, A.: Typeness for ω-regular automata. International Journal on the Foundations of Computer Science **17**(4), 869–884 (2006)

[9] Kupferman, O., Sheinvald, S.: Capacitated automata and systems. Information and Computation **269** (2019)

[10] Kupferman, O., Tamir, T.: Properties and utilization of capacitated automata. In: Proc. 34th Conf. on Foundations of Software Technology and Theoretical Computer Science. LIPIcs, vol. 29, pp. 33–44. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany (2014)

[11] Löding, C.: Optimal bounds for the transformation of ω-automata. In: Proc. 19th Conf. on Foundations of Software Technology and Theoretical Computer Science. Lecture Notes in Computer Science, vol. 1738, pp. 97–109 (1999)

[12] Qiu, R., Joshi, S.: Deterministic finite capacity automata: a solution to reduce the complexity of modeling and control of automated manufacturing systems. In: Proc. Symp. on Computer-Aided Control System Design. pp. 218 –223 (1996)

[13] Savitch, W.: Relationship between nondeterministic and deterministic tape complexities. Journal of Computer and Systems Science **4**, 177–192 (1970)