

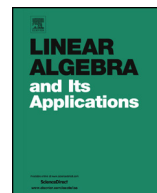


ELSEVIER

Contents lists available at ScienceDirect

## Linear Algebra and its Applications

www.elsevier.com/locate/laa



# The Markov Chain Tree Theorem in commutative semirings and the State Reduction Algorithm in commutative semifields



Buket Benek Gursoy<sup>a</sup>, Steve Kirkland<sup>c,1</sup>, Oliver Mason<sup>b,2</sup>,  
Sergei Sergeev<sup>d,\*,3</sup>

<sup>a</sup> Irish Centre for High-End Computing (ICHEC), Grand Canal Quay, Dublin 2, Ireland

<sup>b</sup> Hamilton Institute, National University of Ireland Maynooth, Maynooth, Co. Kildare, Ireland

<sup>c</sup> Department of Mathematics, University of Manitoba, Winnipeg, R3T 2N2, Canada

<sup>d</sup> University of Birmingham, School of Mathematics, Edgbaston, B15 2TT, UK

## ARTICLE INFO

*Article history:*

Received 22 December 2013

Accepted 14 June 2014

Available online 2 July 2014

Submitted by P. Psarrakos

*MSC:*

15A80

15A18

60J10

05C05

05C85

*Keywords:*

Markov chain

Universal algorithm

## ABSTRACT

We extend the Markov Chain Tree Theorem to general commutative semirings, and we generalize the State Reduction Algorithm to general commutative semifields. This leads to a new universal algorithm, whose prototype is the State Reduction Algorithm which computes the Markov chain tree vector of a stochastic matrix.

© 2014 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/3.0/>).

\* Corresponding author.

*E-mail addresses:* [buket.gursoy@ichec.ie](mailto:buket.gursoy@ichec.ie) (B. Benek Gursoy), [stephen.kirkland@umanitoba.ca](mailto:stephen.kirkland@umanitoba.ca) (S. Kirkland), [oliver.mason@nuim.ie](mailto:oliver.mason@nuim.ie) (O. Mason), [sergiej@gmail.com](mailto:sergiej@gmail.com) (S. Sergeev).

<sup>1</sup> Research supported in part by the University of Manitoba under grant 315729-352500-2000.

<sup>2</sup> Supported by the Irish Higher Education Authority (HEA) PRTL Network Mathematics Grant.

<sup>3</sup> Supported by EPSRC grant EP/J00829X/1 and RFBR grant 12-01-00886.

Commutative semiring  
State reduction

---

## 1. Introduction

The Markov Chain Tree Theorem states that each (row) stochastic matrix  $A$  has a left eigenvector  $x$ , such that each entry  $x_i$  is the sum of the weights of all spanning trees rooted at  $i$  and with edges directed towards  $i$ . This vector has all components positive if  $A$  is irreducible, and it can be 0 in the general case. It can be computed by means of the State Reduction Algorithm formulated independently by Sheskin [26] and Grassmann, Taksar and Heyman [11]; see also Sonin [27] for more information on this.

In the present paper, our main goal is to generalize this algorithm to matrices over commutative semifields, inspired by the ideas of Litvinov et al. [15,17,19]. To this end, let us mention first the tropical mathematics [1,5,12], which is a relatively new branch of mathematics developed over idempotent semirings, of which the tropical semifield, also known as the max algebra, is the most useful example. In one of its equivalent realizations (see Bapat [3]), the max algebra is just the set of nonnegative real numbers equipped with the two operations  $a \oplus b = \max(a, b)$  and  $a \cdot b = ab$ ; these operations extend to matrices and vectors in the usual way. Much of the initial development of max algebra was motivated by applications in scheduling and discrete event systems [1,12]. While this original motivation remains, the area is also a fertile source of problems for specialists in combinatorics and other areas of pure mathematics. See, in particular, [16,20].

According to Litvinov and Maslov [15], tropical mathematics (also called idempotent mathematics due to the idempotency law  $a \oplus a = a$ ) can be developed in parallel with traditional mathematics, so that many useful constructions and results can be translated from traditional mathematics to a tropical/idempotent “shadow” and back. Applying this principle to algorithms gives rise to the programme of making some algorithms universal, so that they work in traditional mathematics, tropical mathematics, and over a wider class of semirings.

There is a well-known universal algorithm, which derives from Gaussian elimination without pivoting. This universal version of Gaussian elimination was developed by Backhouse and Carré [2], see also Gondran [10] and Rote [25]. Based on it, Litvinov et al. [15,17,19] formulated a wider concept of a universal algorithm, and discovered some new universal versions of Gaussian elimination for Toeplitz matrices and other special kinds of matrices. The semifield version of the State Reduction Algorithm found in the present paper can be seen as a new development in the framework of those ideas.

The present paper is also a sequel of our earlier work [4], where the Markov Chain Tree Theorem was proved over the max algebra. To this end, we remark that the max-algebraic analogue of probability is known and has been studied, e.g., by Puhalskii [24] as idempotent probability. Our work is also related to the papers of Minoux [22,23]. However, the Markov Chain Tree Theorem established in the present paper is different from the theorem of [22] which establishes a relation between the spanning tree vector

and bi-determinants of associated matrices of higher dimension. Also, no algorithms for computing the spanning tree vector are offered in [22,23].

Let us mention that the proof of universal Markov Chain Tree Theorem given in the present paper generalizes a proof that can be found in a technical report of Fenner and Westerdale [8]. In our development of the universal State Reduction Algorithm we build upon the above mentioned State Reduction Algorithm of [26,11,27]. The work of Sonin [27] appears to be particularly useful here, since it provides most of the necessary elements of the proof. We recommend both the works of Fenner–Westerdale [8] and Sonin [27] to the reader as well-written explanations of the Markov Chain Tree Theorem and the State Reduction Algorithm in the setting of classical probability. The proofs we give here are predominantly based on combining the arguments of these earlier works and verifying that they generalize to the abstract setting of commutative semirings and semifields.

When specialized to the max algebra, the universal State Reduction Algorithm provides a method for computing the maximal weight of a spanning tree in a directed network. Of course, the problems of minimal and maximal spanning trees in graphs, particularly undirected graphs, have attracted much attention [13]. Recall that in the case of directed graphs, the best known algorithm is the one suggested by Edmonds [7] and, independently, Chu and Liu [6]. This algorithm has some similarities with the universal State Reduction Algorithm (when the latter is specialized to the max algebra), but we will not give any further details on this.

Let us also mention that the State Reduction Algorithm can be seen as a special case of the stochastic complements technique, see Meyer [21].

The rest of the paper is organized as follows. In Section 2 we obtain the universal version of the Markov Chain Tree Theorem. In Section 3 we formulate the universal State Reduction Algorithm and provide a part of its proof. Section 4 is devoted to the proof of a particularly technical lemma (basically following Sonin [27]).

## 2. Markov Chain Tree Theorem in semirings

A semiring  $(S, +, \cdot)$  consists of a set  $S$  equipped with two (abstract) binary operations  $+$ ,  $\cdot$ . The generalized addition,  $+$ , is commutative and associative and has an identity element  $0$ . The generalized multiplication  $\cdot$  is associative and distributes over  $+$  on both the left and the right. There also exists a multiplicative identity element  $1$  and the additive identity is absorbing in the sense that  $a \cdot 0 = 0$  for all  $a \in S$ . We shall only be concerned with commutative semirings, in which  $\cdot$  is also commutative. Next we list some well-known examples of semirings where Theorem 2.6 is valid.

**Example 2.1.** Classical nonnegative algebra which consists of the set of all nonnegative real numbers together with the usual addition and multiplication is a commutative (but not idempotent) semiring.

**Example 2.2.** What we are referring to as the max algebra is often called the max-times algebra to distinguish it from other isomorphic realizations. The max-plus algebra (isomorphic to max algebra via the mapping  $x \rightarrow \exp(x)$ ) consists of  $S = \mathbb{R} \cup \{-\infty\}$  with the operations  $a + b = \max(a, b)$  and  $a \cdot b = a + b$ . The min-plus algebra (isomorphic to max-plus algebra by the mapping  $x \rightarrow -x$ ) consists of  $S = \mathbb{R} \cup \{+\infty\}$  with the operations  $a + b = \min(a, b)$  and  $a \cdot b = a + b$ . All of these realisations are commutative idempotent semirings.

**Example 2.3.** Let  $U$  be a set, and consider a Boolean algebra of subsets of  $U$ . This is an idempotent semiring where  $a + b = a \cup b$  and  $a \cdot b = a \cap b$  for any two subsets  $a, b \subseteq U$ . In the case of finite  $U$ , matrix algebra over  $U$  was considered, e.g., by Kirkland and Pullman [14].

**Example 2.4.** The max–min algebra consisting of  $S = \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$  equipped with  $a + b = \max(a, b)$  and  $a \cdot b = \min(a, b)$  for all  $a, b \in S$  is another commutative idempotent semiring.

**Example 2.5.** Given a semiring  $S$  with idempotent addition ( $a + a = a$ ), equipped with the canonical partial order  $a \preceq b$  iff  $a + b = b$ , an Interval Semiring  $I(S)$  (see [18]) can be constructed as follows.  $I(S)$  consists of order-intervals  $[a_1, a_2]$  (where  $a_1 \preceq a_2$ ) and is equipped with the operations  $+$  and  $\cdot$  defined by  $[a_1, a_2] + [b_1, b_2] = [a_1 + b_1, a_2 + b_2]$ ,  $[a_1, a_2] \cdot [b_1, b_2] = [a_1 \cdot b_1, a_2 \cdot b_2]$ .

We define addition  $A + B$  and multiplication  $AB$  of matrices over  $S$  in the standard fashion. Given a matrix  $A \in S^{n \times n}$ , the weighted directed graph  $D(A)$  is defined in exactly the same way as for matrices with real entries.

Let us proceed with some graph-theoretic definitions. By a (*spanning*) *i*-tree we mean a (directed) spanning tree rooted at  $i$  and directed towards  $i$ . A *functional graph*  $(V, E)$  is a directed graph in which each vertex has exactly one outgoing edge. Such graphs are referred to as “sunflower graphs” in [12]. It is easy to see that a functional graph in general contains several cycles, which do not intersect each other. A functional graph having only one cycle that goes through  $i$  and is not a loop (that is, not an edge of the form  $(i, i)$ ) will be called *i*-unicyclic.

Let  $T$  be a subgraph of  $D(A)$ . Define its *weight*  $\pi(T)$  as the product of the weights of the edges in  $T$ . We will use this definition only in the cases when  $T$  is a directed spanning tree or a unicyclic functional graph. By the total weight of a set of graphs (for example, the set of all *i*-trees or all *i*-unicyclic functional graphs) we mean the sum of the weights of all graphs in the set.

We now present a semiring version of the Markov Chain Tree Theorem. This proof is a semiring extension of the proof in Fenner–Westerdale [8]. See also Freidlin–Wentzell [9, Lemma 3.2] and Sonin [27, Lemma 6].

We denote the set of all  $i$ -trees in  $D(A)$  by  $\mathcal{T}_i$ . The Rooted Spanning Tree (RST) vector  $w \in S^n$  is defined by

$$w_i = \sum_{T \in \mathcal{T}_i} \pi(T), \quad i = 1, \dots, n. \tag{1}$$

In general, the set  $\mathcal{T}_i$  may be empty and then  $w_i = 0$ . In the usual algebra and in the max algebra,  $w$  is positive when  $A$  is irreducible.

A matrix  $A \in S^{n \times n}$  is said to be *stochastic* if  $a_{i1} + a_{i2} + \dots + a_{in} = 1$  for  $1 \leq i \leq n$ .

**Theorem 2.6** (*Markov Chain Tree Theorem in semirings*). *Let  $A \in S^{n \times n}$  and let  $w$  be defined by (1). Then for each  $i = 1, \dots, n$ , we have*

$$w_i \cdot \sum_{j \neq i} a_{ij} = \sum_{j \neq i} w_j a_{ji}. \tag{2}$$

If  $A$  is stochastic then

$$A^T \cdot w = w. \tag{3}$$

**Proof.** To prove (2) we will argue that both parts are equal to the total weight of all  $i$ -unicyclic functional digraphs, which we further denote by  $\pi[i]$ .

On the one hand, every combination of an  $i$ -tree and an edge  $(i, j)$  with  $j \neq i$  results in an  $i$ -unicyclic functional digraph. Indeed, the resulting digraph is clearly functional; moreover, every cycle in it has to contain the edge  $(i, j)$ , so there is only one cycle. Hence, using the distributivity, the left hand side of (2) can be represented as sum of weights of some  $i$ -unicyclic functional digraphs. As each  $i$ -unicyclic functional digraph is uniquely determined by an  $i$ -tree and an edge  $(i, j)$  where  $j \neq i$ , the above mentioned sum contains all weights of such digraphs, with no repetitions. Thus the left hand side of (2) is equal to  $\pi[i]$ .

On the other hand, every combination of a  $j$ -tree and an edge  $(j, i)$  with  $j \neq i$  also results in an  $i$ -unicyclic functional digraph (since every cycle in the resulting functional graph has to contain the edge  $(j, i)$ ). Hence, using the distributivity, the right hand side of (2) can be also represented as sum of weights of some  $i$ -unicyclic functional digraphs. If we take an  $i$ -unicyclic functional graph then  $i$  may have several incoming edges, but only one of them belongs to the (unique) cycle. Hence there is only one  $j$  such that there is an edge  $(j, i)$  and a path from  $i$  to  $j$  so that a  $j$ -tree exists. Thus an  $i$ -unicyclic functional digraph is uniquely determined by a  $j$ -tree and an edge  $(j, i)$  where  $j \neq i$ , and the right hand side of (2) is also equal to  $\pi[i]$ .

Eq. (3) results from adding  $w_i a_{ii}$  to both sides of (2) for each  $i$ , and using the stochasticity of  $A$ .  $\square$

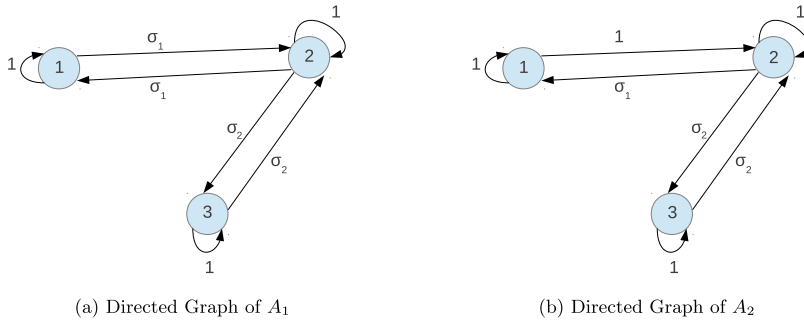


Fig. 1. Directed graphs for Example 2.7.

**Example 2.7.** Consider the Boolean algebra over the two-element set  $U = \{\sigma_1, \sigma_2\}$ . Observe that the  $3 \times 3$  matrix  $A_1 = \begin{bmatrix} 1 & \sigma_1 & 0 \\ \sigma_1 & 1 & \sigma_2 \\ 0 & \sigma_2 & 1 \end{bmatrix}$  is stochastic; the associated directed graph is shown in Fig. 1.

From the figure, it is clear that  $D(A_1)$  has exactly one spanning  $i$ -tree for  $i = 1, 2, 3$ . For example, the only spanning 1-tree is given by the edges  $(3, 2), (2, 1)$  and has weight  $\sigma_2\sigma_1 = 0$  by the properties of the underlying Boolean algebra. Similarly, the unique spanning 2-tree corresponds to the edges  $(1, 2), (3, 2)$  and has weight  $\sigma_1\sigma_2 = 0$ . Referring to (1), it is readily determined that the rooted spanning tree vector for  $A_1$  is the zero vector. The graph defined by the edges  $(1, 2), (2, 3), (3, 2)$  is a functional graph that is 2 unicyclic.

On the other hand, consider the stochastic matrix  $A_2 = \begin{bmatrix} 1 & 1 & 0 \\ \sigma_1 & 1 & \sigma_2 \\ 0 & \sigma_2 & 1 \end{bmatrix}$ , with the associated directed graph shown in Fig. 1.

On this occasion, the unique spanning 1-tree  $(3, 2), (2, 1)$  again has weight  $\sigma_2\sigma_1 = 0$ ; however it is easily checked that the spanning trees rooted at 2 and 3 both have weight  $\sigma_2$ . Thus we find that the rooted spanning tree vector is  $[0 \ \sigma_2 \ \sigma_2]$ .

We note in passing that for the matrix  $A_2$ , the techniques of [14] can be used to show that the vectors  $[1 \ 1 \ \sigma_2]$  and  $[0 \ \sigma_2 \ 1]$  form a basis for the left eigenspace of  $A_2$  corresponding to the eigenvalue 1.

### 3. State Reduction Algorithm in semifields

In this section, we describe an algorithm for computing the spanning tree vector  $w$  in anti-negative semifields. We first recall some necessary definitions. A semiring  $(S, +, \cdot)$  is called a semifield if every nonzero element of  $S$  has a multiplicative inverse. The semifields in Examples 2.1 and 2.2 are commutative semifields.

A semifield  $S$  is antinegative if  $a + b = 0$  implies that  $a = b = 0$  for  $a, b \in S$ . Algorithm 3.1 below provides a universal version of the State Reduction Algorithm for calculating the rooted spanning tree vector of a matrix  $A$  in  $S^{n \times n}$  containing at least 1 off-diagonal entry in each row. Before describing the algorithm itself, it is appropriate

to make some remarks. In the context of semi-fields the problem we are solving with the State Reduction Algorithm is closely related to the classical eigenvector problem, which is fully resolved for the case of max-plus semifield and (in part) for more general idempotent semifields [28]. For the case of a stochastic matrix, this is immediate from (3); for a matrix  $A$  with entries in a semifield and containing at least one non-zero entry in each row, it follows from Theorem 2.6 (2) that the rooted spanning tree vector is an eigenvector of a matrix  $\tilde{A}$  obtained from  $A$  by setting diagonal entries equal to 0 and suitably normalizing each row. It is important to note however that the eigenspace of this matrix may contain several eigenvectors that are linearly independent (over the semifield). Thus methods for solving the eigenvector problem in semirings and semifields will not automatically yield the spanning tree vector. The algorithm we present below selects the particular eigenvector corresponding to the spanning tree vector.

Following [19] we describe this in a language derived from MATLAB. The basic arithmetic operations here are  $a + b$ ,  $ab$  and  $inv(a) := a^{-1}$ . For simplicity, we avoid making too much use of MATLAB vectorization here. However, we exploit the functions “sum” and, respectively, “prod”, which sum up and, respectively, take product of all the entries of a given vector.

**Algorithm 3.1.** State Reduction Algorithm for anti-negative semifields.

**Input:** An  $n \times n$  matrix  $A$  with entries  $a(i, j)$  and at least one non-zero off-diagonal entry in each row,  $A$  is also used to store intermediate results of the computation process.

*Phase 1: State Reduction*

```

for  $i = 1 : n - 1$ 
 $s(i) = \text{sum}(a(i, i + 1 : n))$ 
for  $k = i + 1 : n$ 
for  $l = i + 1 : n$ 
 $a(k, l) = a(k, l) + a(k, i) \cdot a(i, l) \cdot inv(s(i))$ 
end
end
end

```

*Phase 2: Backward Substitution*

```

 $w(n) = \text{prod}(s(1 : n - 1))$ 
 $w(1 : n - 1) = 0$ 
for  $i = n - 1 : -1 : 1$ 
for  $k = i + 1 : n$ 
 $w(i) = w(i) + w(k) \cdot a(k, i) \cdot inv(s(i))$ 
end
end

```

In order for the algorithm to work, it is necessary to ensure that the elements  $s_i$  are non-zero at each step. To this end, we assume that the matrix  $A$  has at least 1 non-zero off-diagonal element in each row. Formally, for  $1 \leq i \leq n$ , there exists some  $j \neq i$  such that  $a_{ij} \neq 0$ . A simple induction using the next lemma then shows that  $s_i$  will be non-zero at each stage of Algorithm 3.1, Phase 1.

**Lemma 3.2.** *Let  $A \in S^{n \times n}$  have at least one non-zero off-diagonal element in each row. Let  $s = \sum_{j=2}^n a_{1j}$  and define  $\hat{A} \in S^{n \times n}$  as follows:*

- (i)  $\hat{a}_{ij} = a_{ij} + s^{-1}a_{i1}a_{1j}$  for  $i, j \geq 2$ ;
- (ii)  $\hat{a}_{ij} = a_{ij}$  otherwise.

Then for  $2 \leq i \leq n$ , there is some  $j \geq 2, j \neq i$  with  $\hat{a}_{ij} \neq 0$ .

**Proof.** Let  $i \geq 2$  be given. By assumption, there is some  $j \neq i$  with  $a_{ij} \neq 0$ . If  $j \geq 2$ , (i) combined with the antinegativity of  $S$  implies that  $\hat{a}_{ij} \neq 0$ . If not, then it follows that  $a_{i1} \neq 0$  and again by assumption there is some  $j$  with  $a_{1j} \neq 0$ . As  $S$  is antinegative, it is immediate from (i) that  $\hat{a}_{ij} \neq 0$ .  $\square$

**Remark 3.3.** Phase 1 is, in fact, similar to the universal LDM decomposition described in [19], with algebraic inversion operations instead of algebraic closure (Kleene star).

Algorithm 3.1 requires  $\frac{n^3}{3} + O(n^2)$  operations of addition,  $\frac{2n^3}{3} + O(n^2)$  operations of multiplication and  $n - 1$  operations of taking inverse. The operation performed in Phase 1 can be seen as a **state reduction**, where a selected state of the network is suppressed, while the weights of the edges not using that state are modified. Recall that in the usual arithmetic and if  $A$  is stochastic, the weights of edges are transition probabilities.

For instance, on the first step of Phase 1 we suppress state 1 and obtain a network with weights

$$a_{kl}^{(1)} = a_{kl} + \frac{a_{k1}a_{1l}}{s_1}, \quad k, l > 1.$$

We inductively define

$$a_{kl}^{(i)} = a_{kl}^{(i-1)} + \frac{a_{ki}^{(i-1)} a_{il}^{(i-1)}}{s_i}, \quad k, l > i,$$

for  $i = 1, \dots, n - 1$ . So  $A^{(i)} = a_{kl}^{(i)}$  is the matrix of the reduced network obtained on the  $i$ th step of Phase 1, by forgetting the states  $1, \dots, i$ .

Denote by  $w^{(i)}$  the spanning tree vector of the  $i$ th reduced Markov model (with  $n - i$  states). This vector has components  $w_{i+1}^{(i)}, \dots, w_n^{(i)}$ . We will further use the following nontrivial statement, whose proof (following Sonin [27]) will be recalled below in Section 4.



**Lemma 3.4.** For all  $i < k$  we have  $s_i \cdot w_k^{(i)} = w_k^{(i-1)}$ .

Let us show (modulo this lemma) that Algorithm 3.1 actually works.

**Theorem 3.5.** Let  $S$  be a commutative anti-negative semifield and  $A \in S^{n \times n}$  be such that every row contains at least one nonzero off-diagonal element. Then Algorithm 3.1 computes the spanning tree vector of  $A$ . If  $A$  is stochastic then this vector is a left eigenvector of  $A$ .

**Proof.** We will prove this theorem by induction, analyzing Phase 2 of Algorithm 3.1.

To begin, we show that initializing  $w(n) = s_{n-1}$  and performing 1 step of Phase 2,  $(w(n-1), w(n))$  is the spanning tree vector of the reduced matrix  $A^{(n-2)}$  on the 2 states  $n-1, n$ . It is easy to check that in this case, we obtain  $w(n-1) = a_{n,n-1}^{(n-2)}$ . We also have  $w(n) = s_{n-1} = a_{n-1,n}^{(n-2)}$  so that in this case,  $(w(n-1), w(n))$  is indeed the spanning tree vector of  $A^{(n-2)}$  as claimed.

For the inductive step, let us make the following assertion: If we initialize  $w(n) = s_{i+1} \cdot \dots \cdot s_{n-1}$  instead of  $w(n) = s_1 \cdot \dots \cdot s_{n-1}$  in the beginning of Phase 2, then the vector  $w(i+1), \dots, w(n)$  obtained on the  $n-i-1$  step of Phase 2 is the spanning tree vector  $w_{i+1}^{(i)}, \dots, w_n^{(i)}$  of the  $i$ th reduced network, with the states  $1, \dots, i$  suppressed.

We have to show that with the above assertion, if we initialize  $w(n) = s_i \cdot \dots \cdot s_{n-1}$  then the vector  $w(i), \dots, w(n)$  obtained on the  $n-i$  step of Phase 2 is the spanning tree vector of the  $i-1$  reduced network.

Indeed, we have

$$w_{i+1}^{(i-1)} = s_i w_{i+1}^{(i)}, \quad \dots, \quad w_n^{(i-1)} = s_i w_n^{(i)},$$

by Lemma 3.4. Combining this with the induction hypothesis and our choice of  $w(n)$ , we see that the components  $w(i+1), \dots, w(n)$  are indeed equal to the entries  $w_{i+1}^{(i-1)}, \dots, w_n^{(i-1)}$  of the spanning tree vector. Next, observe that Algorithm 3.1 computes  $w(i)$  using  $w(i+1), \dots, w(n)$  via the balance equation:

$$s_i w(i) = \sum_{k>i} w_k^{(i-1)} a_{ki}^{(i-1)}.$$

As  $s_i$  is invertible, it now follows from Theorem 2.6 that  $w(i) = w_i^{(i-1)}$ .  $\square$

#### 4. Proof of Lemma 3.4

This proof follows closely that given in Sonin [27, Section 5]. Our main reason for including it in full is to verify that it generalizes to an arbitrary antinegative semifield and to give, in our view, a different and more transparent explanation of the initial proof.

We have to show that  $s_i \cdot w_k^{(i)} = w_k^{(i-1)}$  for all  $k > i$ . It is enough to consider the case when  $i = 1$  and  $k > 1$ . For convenience, let us assume  $k = n$ , so we are to prove that  $s_1 w_n^{(1)} = w_n$ . Recall that here  $w_n$  is the total weight of all  $n$ -trees,  $s_1 = \sum_{j>1} a_{1j}$ , and  $w_n^{(1)}$  is the total weight of all  $n$ -trees in the reduced Markov model where the weight of any edge  $(k, l)$  for  $k, l > 1$  equals

$$a_{kl}^{(1)} = a_{kl} + \frac{a_{k1}a_{1l}}{s_1}. \tag{4}$$

In every tree  $T = (V(T), E(T))$  that contributes to  $w_n$  we can identify the set  $D$  of nodes  $i$  such that  $(i, 1) \in E(T)$  (the edge originating at  $i$  terminates at 1). Further, each tree contributing to  $w_n$  is uniquely determined by (1) the set  $D$ , (2) the forest  $F$  whose (directed) trees are rooted at the nodes of  $D \cup \{n\}$ , and (3) the edge starting at node 1 and ending at a node of the tree rooted at  $n$ .

In contrast to the case of  $w_n, w_n^{(1)}$  (using the distributivity property of  $S$ ) can be written as a sum of terms, where each term is determined not only by an  $n$ -tree on the set  $\{2, \dots, n\}$ , but also by the choice of the first or the second term in (4), made for each edge of the tree. For every such term we can identify the set of nodes  $\tilde{D}$  such that for each edge starting at one of these nodes the second term in (4) is chosen. Further, each term contributing to  $w_n^{(1)}$  is uniquely determined by (1) the set  $\tilde{D}$ , (2) the forest  $\tilde{F}$  whose trees are rooted at the nodes of  $\tilde{D} \cup \{n\}$  and (3) by the mapping  $\tau$  from  $\tilde{D}$  to  $\{2, \dots, n\}$  (which is, in general, neither surjective nor injective).

Given a forest  $F$  on the set  $D \cup \{n\}$  and  $k \in D \cup \{n\}$ , we denote by  $T_k(F)$  the tree rooted at  $k$ .

In view of the above and making use of the distributivity property of  $S$ , the equation  $w_n = s_1 w_n^{(1)}$  is equivalent to the following:

$$\begin{aligned} & \sum_{D,F} \left( \prod_{l \in D} a_{l1} \cdot \prod_{(i,j) \in F} a_{ij} \cdot \sum_{k \in T_n(F)} a_{1k} \right) \\ &= s_1 \cdot \sum_{\tilde{D}, \tilde{F}} s_1^{-|\tilde{D}|} \cdot \left( \prod_{l \in \tilde{D}} a_{l1} \cdot \prod_{(i,j) \in \tilde{F}} a_{ij} \cdot \sum_{\tau: \tilde{D} \rightarrow \{2, \dots, n\}} \prod_{k \in \tilde{D}} a_{1\tau(k)} \right). \end{aligned}$$

As the set of all pairs  $(D, F)$  and the set of all pairs  $(\tilde{D}, \tilde{F})$  are identical, we are left to prove the following identity

$$s_1^{|D|-1} \cdot \sum_{k \in T_n(F)} a_{1k} = \sum_{\tau: D \rightarrow \{2, \dots, n\}} \prod_{k \in D} a_{1\tau(k)}, \quad \forall D, F. \tag{5}$$

The proof of (5) makes use of the following well-known combinatorial identity, whose derivation we will briefly explain, for the reader’s convenience. Let  $T$  be an  $n$ -tree on  $\{1, \dots, n\}$ , and let  $\mathcal{T}$  be the set of all  $n$ -trees. For each node  $k \in \{1, \dots, n\}$ , its *indegree*

$\text{indeg}(k, T)$  in  $T$  is defined as the number of ingoing edges. Let  $x_1, \dots, x_n$  be arbitrary scalars from  $S$ . We will use the following version of Cayley’s tree enumerator formula:

$$(x_1 + \dots + x_n)^{n-2} \cdot x_n = \sum_{T \in \mathcal{T}} x_1^{\text{indeg}(1,T)} \cdot \dots \cdot x_n^{\text{indeg}(n,T)}. \tag{6}$$

Recall that this formula admits a classical proof which works in any commutative semiring. Indeed, observe that for each term on the right hand side of (6), there is at least one variable among  $x_1, \dots, x_{n-1}$  which does not appear, since each tree has at least one leaf. The same is true about the left hand side of (6), since any monomial in the expansion of  $(x_1 + \dots + x_n)^{n-2}$  has total degree  $n - 2$ , which is one less than  $n - 1$ . Due to this observation, it suffices to prove

$$(x_2 + \dots + x_n)^{n-2} \cdot x_n = \sum_{T \in \mathcal{T}: 1 \text{ is a leaf}} x_2^{\text{indeg}(2,T)} \cdot \dots \cdot x_n^{\text{indeg}(n,T)}. \tag{7}$$

Observe that by induction (whose basis for  $n = 2$  is trivial) we have

$$(x_2 + \dots + x_n)^{n-3} \cdot x_n = \sum_{T \in \mathcal{T}'} x_2^{\text{indeg}(2,T)} \cdot \dots \cdot x_n^{\text{indeg}(n,T)}, \tag{8}$$

where  $\mathcal{T}'$  is the set of all (directed)  $n$ -trees on nodes  $2, \dots, n$ . Multiplying both parts of (8) by  $(x_2 + \dots + x_n)$  and using the identity

$$\begin{aligned} & \sum_{T \in \mathcal{T}: 1 \text{ is a leaf}} x_2^{\text{indeg}(2,T)} \cdot \dots \cdot x_n^{\text{indeg}(n,T)} \\ &= (x_2 + \dots + x_n) \cdot \sum_{T \in \mathcal{T}'} x_2^{\text{indeg}(2,T)} \cdot \dots \cdot x_n^{\text{indeg}(n,T)}, \end{aligned}$$

which is due to the bijective correspondence between the trees in  $\mathcal{T}$  having node 1 as a leaf and the combinations of trees in  $\mathcal{T}'$  and edges issuing from node 1, we obtain (7) and hence (6).

To apply (6), observe first that each mapping  $\tau$  in (5) defines a mapping on  $D \cup \{n\}$ : we put an edge  $(u, v)$  for  $u, v \in D \cup \{n\}$  if  $\tau(u)$  belongs to the tree rooted at  $v$ . Further, this mapping defines a directed tree on  $D \cup \{n\}$ , rooted at  $n$ . In particular, observe that any cycle induced by  $\tau$  would yield a cycle in the original graph (which is a spanning tree on the nodes  $2, \dots, n$  rooted at  $n$ ). Also, none of the nodes except for  $n$  can be a root since  $\tau$  is defined for all nodes of  $D$ . We will refer to such a tree on  $D \cup \{n\}$  as a  $\tau$ -induced tree, or just induced tree if the mapping is not specified.

For any pair  $(D, F)$  and for any  $n$ -tree  $T$  on  $D \cup \{n\}$  we can find a mapping  $\tau: D \rightarrow \{2, \dots, n\}$  which yields  $T$  as a  $\tau$ -induced tree. Thus for any given pair  $(D, F)$ , the set of all possible induced trees (with all possible  $\tau$ ), coincides with the set of all  $n$ -trees on  $D \cup \{n\}$ . This set will be further denoted by  $\mathcal{T}_{induced}$ .

Let us set  $x_l = \sum_{k \in T_l(F)} a_{1k}$  for all  $l \in D \cup \{n\}$ . Applying (6) to the set of all reduced trees, with these  $x_l$ , a fixed pair  $D, F$ , and  $|D| + 1$  instead of  $n$ , we have

$$s_1^{|D|-1} \sum_{k \in T_n(F)} a_{1k} = \sum_{T \in \mathcal{T}_{induced}} \left( \prod_{l \in D \cup \{n\}} \left( \sum_{k \in T_l(F)} a_{1k} \right)^{\text{indeg}(l,T)} \right), \quad \forall D, F. \tag{9}$$

We are left to show that the right-hand sides of (5) and (9) coincide.

For an induced tree  $T$ , let  $\tau: D \xrightarrow{T} \{2, \dots, n\}$  denote the fact that  $T$  is  $\tau$ -induced. For each  $l \in D \cup \{n\}$  let  $\text{in}(l, T)$  denote the set of in-neighbors of  $l$ . Consider the following chain of equalities, with  $D$  and  $F$  fixed

$$\begin{aligned} \sum_{\tau: D \rightarrow \{2, \dots, n\}} \left( \prod_{k \in D} a_{1\tau(k)} \right) &= \sum_{T \in \mathcal{T}_{induced}} \left( \sum_{\tau: D \xrightarrow{T} \{2, \dots, n\}} \left( \prod_{k \in D} a_{1\tau(k)} \right) \right) \\ &= \sum_{T \in \mathcal{T}_{induced}} \prod_{l \in D \cup \{n\}} \left( \sum_{\sigma: \text{in}(l, T) \rightarrow T_l(F)} \prod_{s \in \text{in}(l, T)} a_{1, \sigma(s)} \right) \\ &= \sum_{T \in \mathcal{T}_{induced}} \prod_{l \in D \cup \{n\}} \left( \sum_{k \in T_l(F)} a_{1k} \right)^{\text{indeg}(l, T)}. \end{aligned} \tag{10}$$

These equalities can be explained as follows. On the first step, we classify mappings  $\tau$  according to the induced trees that they yield. On the next step,  $D$  is represented as a union over all sets  $\text{in}(l, T)$  where  $l \in D \cup \{n\}$ , and we use the fact that each  $\tau: D \xrightarrow{T} \{2, \dots, n\}$  can be decomposed into a set of some “partial” mappings  $\sigma: \text{in}(l, T) \rightarrow T_l(F)$ , and vice versa; every combination of such “partial” mappings gives rise to a mapping  $\tau$  that yields  $T$  (as a  $\tau$ -induced tree). On the last step we use the multinomial semiring identity

$$\left( \sum_{k \in T_l(F)} a_{1k} \right)^{\text{indeg}(l, T)} = \sum_{\sigma: \text{in}(l, T) \rightarrow T_l(F)} \left( \prod_{s \in \text{in}(l, T)} a_{1, \sigma(s)} \right). \tag{11}$$

To understand this identity observe that the left hand side of (11) is a product of  $\text{indeg}(l, T) = |\text{in}(l, T)|$  identical sums of  $|T_l(F)|$  terms. By distributivity, this product can be written as a sum of monomials, where each monomial corresponds to a combination of choices made in each bracket, and hence to a mapping  $\sigma: \text{in}(l, T) \rightarrow T_l(F)$ .

Finally, by (10) the right-hand sides of (5) and (9) are equal, and this completes the proof.

**References**

[1] F. Baccelli, G. Cohen, G.J. Olsder, J.-P. Quadrat, Synchronization and linearity: an algebra for discrete event systems, free web edition, <http://www-roc.inria.fr/metalau/cohen/documents/BCOQ-book.pdf>, 2001.

- [2] R.C. Backhouse, B.A. Carré, Regular algebra applied to path-finding problems, *J. Inst. Math. Appl.* 15 (1975) 161–186.
- [3] R.B. Bapat, A max version of the Perron–Frobenius theorem, *Linear Algebra Appl.* 275–276 (1998) 3–18.
- [4] B. Benek Gursoy, S. Kirkland, O. Mason, S. Sergeev, On the Markov chain tree theorem in the max algebra, *Electron. J. Linear Algebra* 26 (2013) 15–27.
- [5] P. Butkovič, *Max-Linear Systems: Theory and Algorithms*, Springer, 2010.
- [6] Y.J. Chu, T.H. Liu, On the shortest arborescence of a directed graph, *Sci. Sin.* 14 (1965) 1396–1400.
- [7] J. Edmonds, Optimum branchings, *J. Res. Natl. Bur. Stand.* 69B (1967) 125–130.
- [8] T.I. Fenner, T.H. Westerdale, A random graph proof for the irreducible case of the Markov chain tree theorem, Technical Report BBKCS-00-05, Birkbeck, University of London, Department of Computer Science and Information Systems, 2000, available from <http://www.dcs.bbk.ac.uk/research/techreps/2000/bbkcs-00-05.pdf>.
- [9] M.I. Freĭdlin, A.D. Wentzell, *Perturbations of Stochastic Dynamic Systems*, Springer, 1984; translation of Russian edition: Nauka, Moscow, 1979.
- [10] M. Gondran, Path algebra and algorithms, in: B. Roy (Ed.), *Combinatorial Programming: Methods and Applications*, Reidel, Dordrecht, 1975, pp. 137–148.
- [11] W.K. Grassmann, M.I. Taksar, D.P. Heyman, Regenerative analysis and steady-state distributions for Markov chains, *Oper. Res.* 33 (1985) 1107–1116.
- [12] B. Heidergott, G.J. Olsder, J. van der Woude, *Max Plus at Work: Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications*, Princeton University Press, 2006.
- [13] D. Jungnickel, *Graphs, Networks and Algorithms*, Springer-Verlag, 2005.
- [14] S. Kirkland, N.J. Pullman, Boolean spectral theory, *Linear Algebra Appl.* 175 (1992) 177–190.
- [15] G.L. Litvinov, V.P. Maslov, The correspondence principle for idempotent calculus and some computer applications, in: J. Gunawardena (Ed.), *Idempotency*, Cambridge University Press, 1998, pp. 420–443.
- [16] G.L. Litvinov, V.P. Maslov (Eds.), *Idempotent Mathematics and Mathematical Physics*, *Contemp. Math.*, vol. 307, Amer. Math. Soc., Providence, 2005.
- [17] G.L. Litvinov, E.V. Maslova, Universal numerical algorithms and their software implementation, *Program. Comput. Software* 26 (5) (2000) 275–280.
- [18] G.L. Litvinov, A.N. Sobolevskii, Idempotent interval analysis and optimization problems, *Reliab. Comput.* 7 (5) (2001) 353–377.
- [19] G.L. Litvinov, A.Ya. Rodionov, S.N. Sergeev, A.N. Sobolevski, Universal algorithms for solving the matrix Bellman equations over semirings, *Soft Comput.* 17 (10) (2013) 1767–1785.
- [20] G.L. Litvinov, S.N. Sergeev (Eds.), *Tropical and Idempotent Mathematics*, *Contemp. Math.*, vol. 495, Amer. Math. Soc., Providence, 2009.
- [21] C.D. Meyer, Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems, *SIAM Rev.* 31 (1989) 240–272.
- [22] M. Minoux, Bideterminants, arborescences and extension of matrix-tree theorem to semirings, *Discrete Math.* 171 (1997) 191–200.
- [23] M. Minoux, Extension of Mac-Mahon’s master theorem to pre-semi-rings, *Linear Algebra Appl.* 338 (2001) 19–26.
- [24] A. Puhalskii, *Large Deviations and Idempotent Probability*, Chapman & Hall, 2001.
- [25] G. Rote, A systolic array algorithm for the algebraic path problem, *Computing* 34 (1985) 191–219.
- [26] T. Sheskin, A Markov partitioning algorithm for computing steady-state probabilities, *Oper. Res.* 33 (1985) 228–235.
- [27] I. Sonin, The state reduction and related algorithms and their applications to the study of Markov chains, graph theory, and the optimal stopping problem, *Adv. Math.* 145 (1999) 159–188.
- [28] U. Zimmermann, *Linear and Combinatorial Optimization in Ordered Algebraic Structures*, *Ann. Discrete Math.*, vol. 10, North-Holland, Amsterdam, 1981.