Stochastic Graph Transformation For Social Network Modeling

Nicolas Behr
Univ. de Paris, CNRS, IRIF
Paris, France

Bello Shehu Bello
Dept. of Computer Science
Bayero Univ. Kano, Nigeria

Sebastian Ehmes
Real-Time Systems Lab
TU Darmstadt, Germany

Reiko Heckel*
School of Informatics
Univ. of Leicester, UK

Adaptive networks model social, physical, technical, or biological systems as attributed graphs evolving at the level of both their topology and data. They are naturally described by graph transformation, but the majority of authors take an approach inspired by the physical sciences, combining an informal description of the operations with programmed simulations, and systems of ODEs as the only abstract mathematical description. We show that we can capture a range of social network models, the so-called voter models, as stochastic attributed graph transformation systems, demonstrate the benefits of this representation and establish its relation to the non-standard probabilistic view adopted in the literature. We use the theory and tools of graph transformation to analyze and simulate the models and propose a new variant of a standard stochastic simulation algorithm to recreate the results observed.

1 Introduction

Modeling and analyzing the dynamics of social networks allows scientists to understand the impact of social interactions on areas as diverse as politics (opinion formation, spread of (dis)information), economic development, and health (spread of diseases, update of vaccines) [24, 1, 9]. Much of the more foundational literature approach social network analysis from a perspective informed by statistical physics [23] using a combination of mathematical models (differential equations) and programmed simulation, both derived from an intuitive understanding of the operation of the network. This works well for static networks, where the structure is fixed and changes are to node or link attributes only, but in complex adaptive networks [15] the interconnectedness of structure and data evolution poses additional challenges.

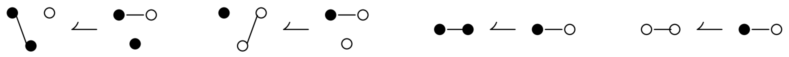
Stochastic typed attributed graph transformation [16] is an obvious choice to formalize and analyze complex adaptive networks. The formalism provides both tool support for simulation and analysis and an established theory to derive mathematical models from the same rule-based descriptions, thus replacing the informal, and sometimes vague, descriptions in natural language. A case in point are the various voter models [11, 28, 18] which describe opinion formation in a network of agents. The operations, as described in [11] seem clear enough.

> ... we consider two opinions (called 0 and 1) ... ; and on each step, we pick a discordant edge (x,y) at random .... With probability $1-\alpha$, the voter at x adopts the opinion of the voter at y. Otherwise (i.e., with probability α), x breaks its connection to y and makes a new connection to a voter chosen at random from those that share its opinion. The process continues until there are no edges connecting voters that disagree.

[*] Corresponding author email address: rh122@leicester.ac.uk

B. Hoffmann and M. Minas (Eds.): Twelfth International
Workshop on Graph Computation Models (GCM 2021)
EPTCS 350, 2021, pp. 35–50, doi:10.4204/EPTCS.350.3

Intuitively, this is a graph transformation system of four rules over undirected graphs whose nodes are attributed by 0 or 1, shown below using ○ and •, respectively. Referring to [8, 2, 6, 7] for further details, we utilise a "right-to-left" convention for rewriting rules in order to accommodate a natural order of composition of the rule algebra representations.



The first pair of rules models *rewiring* where an agents disconnects from another one with a different opinion to form a connection with a third agent of the same opinion. In the second pair of rules, an agent connected to one with a different opinion *adopts* the opinion of the other. We could now associate rates with these rules, resulting in a stochastic graph transformation system that allows simulation using the SimSG tool [12] and the derivation of differential equations using the rule algebra approach [8, 2, 6, 7].

However, on closer inspection we discover a number of discrepancies. First, the model in [11] and related papers is probabilistic, but without time. This is an abstraction of real-world behavior in social networks, where time is not discrete and actions not round-based. Arguably, a continuous-time model is a better representation of this behavior. Semantically, a stochastic graph transformation system induces a continuous-time Markov Chain (CTMC) — an established model with clear links to logics, model checking and simulation techniques, while the operational model behind [11] and others is left informal, but could be formalized as a discrete-time Markov chain (DTMC) or decision process (MDP) [4, 5, 8, 7].

More specifically, looking at the description of the operation, this is a two- or three-step process, where first a conflict edge is selected at random, then a decision made based on the fixed probability α between adopting another opinion, and rewiring which requires another random selection of a node with the same opinion. That means, in the rewiring case, the combined behavior of the operation is not reflected directly by the rewiring rules above, which choose all three nodes first. Also, the race between the rewiring and adopt rules in a stochastic graph transformation system depends on the number of available matches (the candidates for the 3rd agent to link) while the probability is fixed in the original formulation. Stochastic graph transformation realizes a mass action semantics which reflects the behavior of physical, chemical and biological processes where the frequency of actions (formally the jump rate of the CTMC) depends on both a rate constant and the concentration or amount available of the input materials required for the action.

In our view, the informal description of operations, lack of continuous time, and non-standard selection and execution procedure all represent weaknesses in the formulation which inhibit a natural mathematical interpretation of the operational behavior of the model and consequently a formal and systematic connection of this operational description with the simulation and equational model.

Other authors have studied algorithmic problems in social networks, such as what subset of actors to target with interventions (e.g., in marketing or public health campaigns) so as to maximise the impact of the campaign on the network as a whole [17, 22]. Such research is based on models of opinion formation and the spread of information in networks, typically divided into *cascading* and *threshold models*. In a cascading model, an actor is activated by a single message or activation from a peer. The behavior outlined above, where a discordant edge represents an activation, falls into this category. Instead, in a threshold model, an actor becomes active once a certain number of its peers are activated.

Different models give rise to different centrality measure for actors [25] and hence to different algorithmic solutions for discovering the most influential set of actors to target. These models assume static networks, but one could ask the question of centrality of actors for a dynamic network model given by a stochastic graph transformation system incorporating both the spread of information and the change of topology.

In this paper, we present an approach addressing weaknesses in dynamic social network modelling by starting from a formal graph transformation model as operational description. We investigate the links to the existing voter models both analytically, using the rule algebra framework to establish how to translate operations and parameters to stochastic graph transformation, and experimentally trying to recreate the observed emergent behavior in the literature. Apart from making a convincing claim for the superiority of our methodological approach, this will allow us in the future to both compare new analysis results to existing ones in the literature.

We start by introducing the model as a stochastic graph transformation system in the format accepted by our simulation tool, then study the theory of relating the different CTMC and DTMC semantics to support different possible conversions between models before reporting on the experimental validation of the resulting models through simulation.

2 Voter Models as Stochastic Typed Attributed Graph Rewrite Systems

We model voter networks as instances of the type graph below (Figure 1), where *Group* nodes represent undirected links connecting *Agent* nodes. This means that two agents are linked if they are members of the same group. For now, the cardinality of each group is exactly 2, so these groups are indeed a model of undirected edges between agents (and we are planning to generalize to groups with several members later). We see the voter network as an undirected multi graph, i.e., parallel links between two voters $v1$ and $v2$ are permitted. That means, in our representation, $v1, v2$ can jointly make up one group $g1$ as well as another group $g2$. It is worth noting that the multi-graph interpretation is never explicitly stated in [11] nor the literature building on it. However it is the only interpretation consistent with the usual assumptions made, in particular that rewiring is always possible after the selection of the 3rd node, which may or may not be connected to the first node $v1$ already, and that the total number of edges in the graph is constant, so we have to create a new edge on rewiring even if $v1$ and $v2$ are already connected.

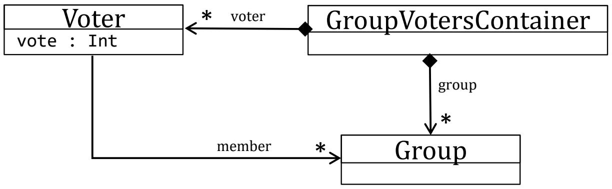


Figure 1: Type graph of the basic voter model

In this paper we use SimSG [12] to simulate rule-based models, such as the basic voter model, for a given start graph and rates according to the semantics of stochastic graph transformations [16]. SimSG implements its own version of Gillespie's well-known algorithm [14] which describes the behavior of stochastic systems over time using a continuous-time Markov Chain with exponentially distributed transition delays. SimSG is build upon the graph transformation rule interpreter eMoflon [21], which provides the means to define and execute rules. For example, there are four rules in the basic voter model, two dual variants each of the *adopt* and *rewire* operations. The graphs in Figure 2 show a visual representation of these four rules as defined in the eMoflon::IBeX-GT[1] syntax. In these rules, black and red elements (--) specify context elements that need to be present in an instance graph, while green elements (++) will be created. A pattern matcher will search for matches in an instance graph that fit these require-

[1] eMoflon::IBeX-GT: `https://emoflon.org/#emoflonIbex`

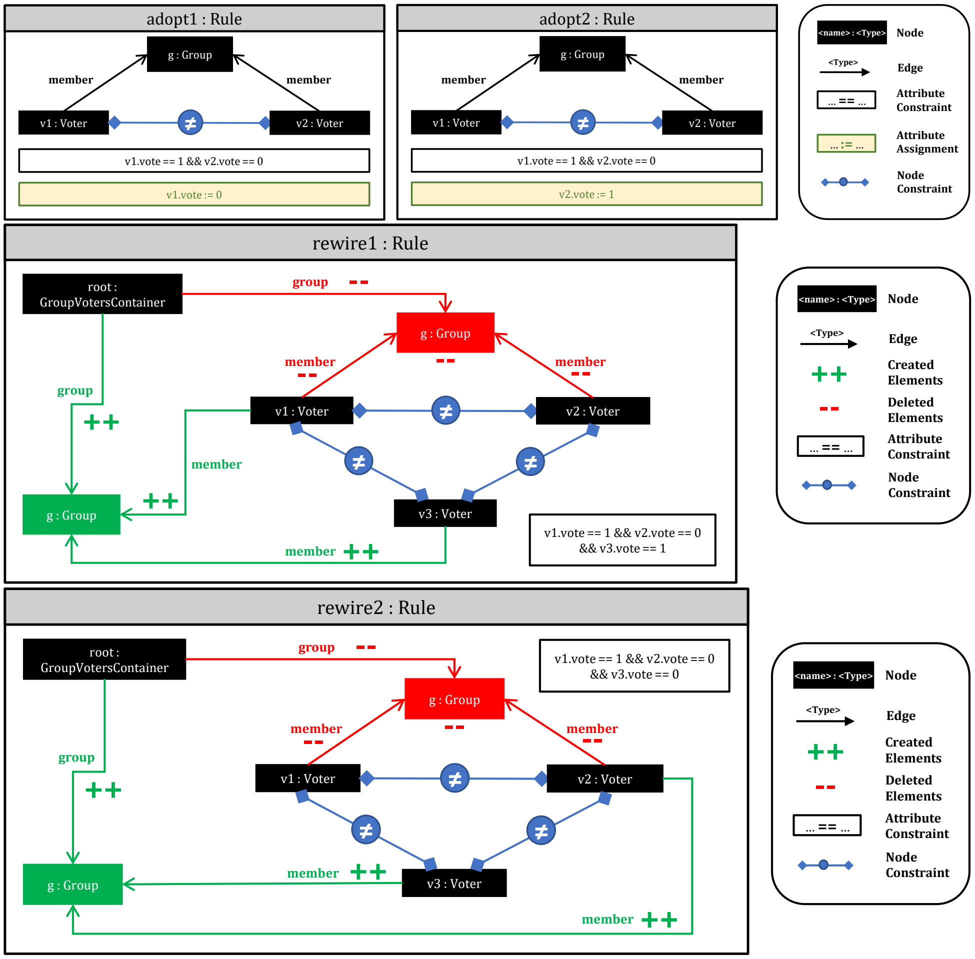


Figure 2: Rules of the basic voter model

ments. As shown in Figure 2, in addition to structural constraints, eMoflon allows the specification of attribute conditions. If a match is found, the rule can be applied by deleting all graph elements matching red elements in the rule, and creating new instances of all green elements. In addition to the structural constraints and attribute conditions shown in the example, eMoflon-GT also allows the definition of more complex conditions, such as negative application conditions that filter out matches connected to prohibited graph structures. Beyond the specification of rules, SimSG allows for the annotation of rules with rates, either through value literals or statically evaluated arithmetic expressions. Most importantly, eMoflon provides SimSG with an interface to its underlying incremental graph pattern matching engines, such as Viatra[26], Democles[27] or the recently developed HiPE[2], all of which can be used to find matches for rules and track observable patterns during simulations. During each simulation, SimSG tracks occurrence counts of these patterns and provide the user with the option to plot these counts over the simulation time or save them to a file.

[2] HiPE: `https://github.com/HiPE-DevOps/HiPE-Updatesite`

3 Theory: CTMCs and DTMCs via Rule-algebraic Methods

Approaches to social network modeling can be broadly classified by their underlying semantics. In this paper, we consider the two major classes consisting of *discrete-time Markov chains (DTMCs)* and *continuous-time Markov Chains (CTMCs)* (leaving for future work the class of Markov decision processes). We will utilize the *rule algebra formalism* [8, 2, 6, 7] as our central technical tool, with CTMCs in so-called mass-action semantics implemented via the *stochastic mechanics framework* [4, 5, 8, 7]. Here, the firing rate of a given rule for a system state at some time $t \geq 0$ is proportional to a *base rate* (i.e., a positive real parameter) times the number of admissible matches of the rule into the system state. Note however that even in the CTMC setting, this semantics is only one of several conceivable variants, and indeed we will study in this paper also a variation of rule-based CTMC semantics wherein rule activities in infinitesimal jumps are not weighted by their numbers of matches. In order to faithfully implement rule-based DTMCs, we will adopt the approach of [3].

3.1 Preliminaries: "compositional" rewriting theory

We will focus throughout this paper on the case[3] of Sesqui-Pushout (SqPO) semantics [10], which under certain conditions on the underlying base categories furnishes a "compositional" rewriting semantics [2], i.e., supports the requisite construction of rule algebras. Since the main theme of this paper is to advocate rewriting-based methods for implementing models, we discuss here the recent extension of the SqPO-formalism to include rules with conditions and under the influence of constraints [6, 7]. We refer the interested readers to [7] for the analogous formalism in the Double Pushout (DPO) setting.

Consider thus a *base category* $\mathbf{C}$ that is *$\mathscr{M}$-adhesive* (for some class $\mathscr{M}$ of monomorphisms), *finitary*, possesses an $\mathscr{M}$-initial object, $\mathscr{M}$-effective unions and an epi-$\mathscr{M}$-factorization, and such that all final pullback complements (FPCs) along composable pairs of $\mathscr{M}$-morphisms exist (compare [7]). Throughout this paper, we will consider the particular example of the category $\mathbf{uGraph}/T$ (for some $T \in \mathsf{obj}(\mathbf{uGraph})$) of *typed undirected multigraphs*, which according to [7] satisfies all of the aforementioned requirements.

Let us briefly recall the salient points of SqPO-type "compositional" rewriting theory:

Definition 1 (Rules)**.** Denote by $\overline{\mathsf{Lin}}(\mathbf{C})$ the set of equivalence classes of linear rules with conditions,

$$\overline{\mathsf{Lin}}(\mathbf{C}) := \{R = (O \xleftarrow{o} K \xrightarrow{i} I, \mathsf{c}_I) | i, o \in \mathscr{M}, \mathsf{c}_I \in \mathsf{cond}(\mathbf{C})\} \diagup_{\sim}, \tag{1}$$

where $R \sim R'$ if and only if $\mathsf{c}_I \dot{\equiv} \mathsf{c}'_I$ (i.e., if the conditions are equivalent for all matches of the rules) and the rules are isomorphic. The latter entails the existence of isomorphisms $\omega : O \xrightarrow{\cong} O'$, $\kappa : K \xrightarrow{\cong} K'$ and $\iota : I \xrightarrow{\cong} I'$ such that the obvious diagram commutes. We will adopt the convention to speak of "a" rule $R \in \overline{\mathsf{Lin}}(\mathbf{C})$ to mean an equivalence class of rules.

Definition 2. Given a rule $R \in \overline{\mathsf{Lin}}(\mathbf{C})$ and an object $X \in \mathsf{obj}(\mathbf{C})$, an *SqPO-admissible match* $m \in \mathsf{M}_R(X)$ is an $\mathscr{M}$-morphism $m : I \to X$ that satisfies the application condition c_I. Then an *SqPO-type direct derivation* of X along R with a match $m \in \mathsf{M}_R(X)$ is defined via a commutative diagram of the form

$$\begin{tikzcd}
O \arrow[d, "m^*"'] \arrow[dr, phantom, "PO"] & K \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\mathsf{FPC}"] & I \arrow[d, "m"] \\
R_m(X) & \overline{X} \arrow[l] \arrow[r] & X
\end{tikzcd} \tag{2}$$

[3] Since the rules in our concrete models are *linear* and will not involve vertex deletions nor creations, we could have equivalently opted for DPO- or SPO-semantics, which are well-known to coincide with SqPO-semantics in this special situation.

where the object $R_m(X)$ is defined up to the universal isomorphisms of FPCs and pushouts (POs).

Note that in all of our applications, we will only consider objects up to isomorphisms, so we will in a slight abuse of notations speak of taking direct derivations of iso-classes $X \in \mathsf{obj}(\mathbf{C})_{\cong}$ along (equivalence classes of) rules, understood as a mapping from iso-classes to iso-classes. With the focus of the present paper upon implementation strategies, we forgo here a complete review of the concepts of compositional rewriting theory (see e.g. [6, 7]), and limit ourselves to the following salient points. From hereon, $\mathbf{C}$ will always denote a category suitable for SqPO-type transformation.

Definition 3. Let $\mathscr{R}_{\mathbf{C}}$ denote the $\mathbb{R}$-*vector space "over rules"* spanned by basis vectors $\delta(R)$, where $\delta : \overline{\mathsf{Lin}}(\mathbf{C}) \xrightarrow{\cong} \mathsf{basis}(\mathscr{R}_{\mathbf{C}})$ is an isomorphism. Let $\hat{\mathbf{C}}$ denote the $\mathbb{R}$-*vector space "over states"*, which is defined via the isomorphism $|.\rangle : \mathsf{obj}(\mathbf{C})_{\cong} \to \mathsf{basis}(\hat{\mathbf{C}})$ from isomorphism-classes of objects to basis vectors of $\hat{\mathbf{C}}$. We introduce the notation $End_{\mathbb{R}}(\hat{\mathbf{C}})$ to denote the space of *endomorphisms* on $\hat{\mathbf{C}}$ (i.e., of linear operators from $\hat{\mathbf{C}}$ to itself). Then we denote by $\rho_{\mathbf{C}} : \mathscr{R}_{\mathbf{C}} \to End_{\mathbb{R}}(\hat{\mathbf{C}})$ the morphism defined via

$$\forall R \in \overline{\mathsf{Lin}}(\mathbf{C}), X \in \mathsf{obj}(\mathbf{C})_{\cong} : \quad \rho_{\mathbf{C}}(\delta(R))\,|X\rangle := \sum_{m \in \mathsf{M}_R(X)} |R_m(X)\rangle\,. \tag{3}$$

The above definition extends to arbitrary elements of $\mathscr{R}_{\mathbf{C}}$ and $\hat{\mathbf{C}}$ by *linearity*, i.e., for $A = \sum_{j=1}^{M} \alpha_j \delta(R_j)$ and $|\Psi\rangle := \sum_X \psi_X\,|X\rangle$, we define $\rho_{\mathbf{C}}(A)\,|X\rangle := \sum_{j=1}^{M} \alpha_j \sum_X \psi_X\, \rho_{\mathbf{C}}(\delta(R_j))\,|X\rangle$.

Intuitively, $\rho_{\mathbf{C}}$ takes a rule vector in $\mathscr{R}_{\mathbf{C}}$ and delivers a transformation between state vectors, applying the rules to the input states, and "weighing" the resulting states according to the coefficients in the rule vector. This concept is useful in particular in order to formalize the jumps of a CTMC (see Section 3.2).

Theorem 1 (Cf. [7])**.** *(i) The* trivial rule $R_{\varnothing} := [(\varnothing \leftarrow \varnothing \to \varnothing, \mathsf{true})]_{\sim} \in \overline{\mathsf{Lin}}(\mathbf{C})$ *satisfies*

$$\rho_{\mathbf{C}}(\delta(R_{\varnothing})) = Id_{End_{\mathbb{R}}(\hat{\mathbf{C}})}\,. \tag{4}$$

(ii) Denote by $\langle| : \hat{\mathbf{C}} \to \mathbb{R}$ *the so-called* (dual) projection vector*, defined via* $\langle|X\rangle := 1_{\mathbb{R}}$ *for all* $X \in \mathsf{obj}(\mathbf{C})_{\cong}$*. Extending this definition by linearity,* $\langle|$ *thus implements the operation of* summing coefficients*, i.e., for* $|\Psi\rangle = \sum_X \psi_X\,|X\rangle$*, we define* $\langle|\Psi\rangle := \sum_X \psi_X \langle|X\rangle = \sum_X \psi_X$*. Then* $\rho_{\mathbf{C}}$ *satisfies the so-called* ***SqPO-type jump-closure property****:*

$$\forall R \in \overline{\mathsf{Lin}}(\mathbf{C}) : \quad \langle|\,\rho_{\mathbf{C}}(\delta(R)) = \langle|\,\hat{\mathbb{O}}(\delta(R))\,, \quad \hat{\mathbb{O}}(\delta(R)) := \rho_{\mathbf{C}}(\delta([I \leftarrow I \to I, \mathsf{c}_I]_{\sim}))\,. \tag{5}$$

The last point of the above theorem alludes to the important special case of rule-based linear operators that are *diagonal* in the basis of $\hat{\mathbf{C}}$, since these implement operations of *counting patterns*. In the particular case at hand, we may combine the definition of $\rho_{\mathbf{C}}$ in (3) with (5) in order to obtain

$$\langle|\,\hat{\mathbb{O}}(\delta(R))\,|X\rangle = \langle|\,\rho_{\mathbf{C}}(\delta(R))\,|X\rangle = \sum_{m \in \mathsf{M}_R(X)} \langle|R_m(X)\rangle = \sum_{m \in \mathsf{M}_R(X)} 1_{\mathbb{R}} = |\mathsf{M}_R(X)|\,. \tag{6}$$

In this sense, $\hat{\mathbb{O}}(\delta(R))$ as defined in (5) "counts" the numbers of admissible matches of R into objects.

3.2 Rule-based CTMCs

Traditionally, CTMCs based upon transformation rules have been defined in so-called *mass-action semantics*. In the rule algebra formalism, such types of CTMCs are compactly expressed as follows (with $\rho \equiv \rho_{\mathbf{C}}$):

Theorem 2 (Mass-action semantics for CTMCs)**.** *Let* $\mathscr{T} := \{(\kappa_j, R_j)\}_{j=1}^{N}$ *denote a (finite) set of pairs of* base rates $\kappa_j \in \mathbb{R}_{>0}$ *and rules* $R_j \in \overline{\mathsf{Lin}}(\mathbf{C})$ *(for* $j = 1, \dots, N$*) over some category* $\mathbf{C}$*. Denote by* $\mathsf{Prob}(\hat{\mathbf{C}})$ *the space of (sub-)probability distributions over the state-space* $\hat{\mathbf{C}}$*, and choose an* initial state $|\Psi_0\rangle := \sum_X \psi_X^{(0)} |X\rangle \in \mathsf{Prob}(\hat{\mathbf{C}})$*. Then the data* $(\mathscr{T}, |\Psi_0\rangle)$ *defines a* ***SqPO-type rule-based CTMC*** *via the following* ***evolution equation*** *for the* ***system state*** $|\Psi(t)\rangle := \sum_X \psi_X(t) |X\rangle$ *at time* $t \geq 0$*:*

$$\tfrac{d}{dt} |\Psi(t)\rangle = H |\Psi(t)\rangle \,, \quad |\Psi(0)\rangle = |\Psi_0\rangle \,, \quad H := \rho(h) - \hat{\mathbb{O}}(h) \,, \quad h := \sum_{j=1}^{N} \kappa_j \delta(R_j) \tag{7}$$

Here, we used linearity to extend the definition of the jump-closure operator $\hat{\mathbb{O}}(.)$ *of* (5) *to arbitrary elements of* $\mathscr{R}_{\mathbf{C}}$*, i.e.,* $\hat{\mathbb{O}}(h) := \sum_{j=1}^{N} \kappa_j \hat{\mathbb{O}}(\delta(R_j))$*.*

However, while mass-action semantics is of key importance in the modeling of chemical reaction systems, it is of course not the only conceivable semantics. In particular, as we shall demonstrate in the later part of this paper, for certain applications it will prove useful to utilize alternative adjustments of the "firing rates" of individual rules other than the one fixed in mass-action semantics.

Definition 4 (Generalized rule-based CTMC semantics)**.** For a suitable base category $\mathbf{C}$, let $\mathscr{T} := \{(\gamma_j, R_j, W_j)\}_{j=1}^{N}$ be a set of triples of *base rates* $\gamma_j \in \mathbb{R}_{>0}$, *rules* $R_j \in \overline{\mathsf{Lin}}(\mathbf{C})$ and *(inverse) weight functions* $W_j \in End(\hat{\mathbf{C}})_{diag}$ $(j = 1, \dots, N)$. Here, $End(\hat{\mathbf{C}})_{diag}$ denotes the space of *diagonal operators* (with respect to the basis of $\hat{\mathbf{C}}$). Together with a choice of *initial state* $|\Psi_0\rangle \in \mathsf{Prob}(\hat{\mathbf{C}})$, this data defines a *rule-based SqPO-type CTMC* via the *evolution equation*

$$\begin{aligned}
&\tfrac{d}{dt} |\Psi(t)\rangle = H_{\underline{W}} |\Psi(t)\rangle \,, \quad |\Psi(0)\rangle = |\Psi_0\rangle \,, \quad H_{\underline{W}} := \sum_{j=1}^{N} \gamma_j \left(\rho(\delta(R)_j) - \hat{\mathbb{O}}(\delta(R_j))\right) \frac{1}{W_j^*} \\
&\forall F \in End(\hat{\mathbf{C}})_{diag}, |X\rangle \in \hat{\mathbf{C}} : \quad \frac{1}{F^*} |X\rangle := \begin{cases} |X\rangle & \text{if } F|X\rangle = 0_{\mathbb{R}} \\ \frac{1}{\langle |F|X\rangle} |X\rangle & \text{otherwise.} \end{cases}
\end{aligned} \tag{8}$$

Example 1. Given a set of pairs of base rates and rules $\mathscr{T} = \{(\kappa_j, R_j)\}_{j=1}^{N}$ for a rule-based CTMC with mass-action semantics as in Definition 2, one may define from this transition set in particular a *uniformed* CTMC with infinitesimal generator H_U defined as follows (with $\rho \equiv \rho_{\mathbf{C}}$):

$$H_U := -Id_{End(\hat{\mathbf{C}})} + \sum_{j=1}^{N} p_j \cdot \rho(\delta(R_j)) \frac{1}{\hat{\mathbb{O}}(\delta(R_j))^*} \,, \quad p_j := \frac{\kappa_j}{\sum_{j=1}^{N} \kappa_j} \,. \tag{9}$$

Note in particular that since all base rates κ_j are positive real numbers, the parameters p_j are *probabilities*. Moreover, applying H_U to an arbitrary basis state $|X\rangle$, one finds that the overall jump rate (i.e., minus the coefficient of the diagonal part of H_U) is precisely equal to 1, so that the p_j in the non-diagonal part of H_U encodes in fact the probability for the "firing" of rule R_j, regardless of the numbers of admissible matches of R_j into the given state $|X\rangle$. Finally, under the assumption that both the mass-action and the uniformed CTMC admit a *steady state*, we find by construction that

$$\lim_{t\to\infty} \tfrac{d}{dt} |\Psi(t)\rangle = 0 \quad \Leftrightarrow \quad \lim_{t\to\infty} H |\Psi(t)\rangle = 0 \quad \Leftrightarrow \quad \lim_{t\to\infty} H_U \left(\hat{\mathbb{O}}(h) \cdot |\Psi(t)\rangle\right) = 0 \,. \tag{10}$$

Letting $|\Psi_U(t)\rangle$ denote the system state at time t of the "uniformed" CTMC with generator H_U, and assuming $|\Psi(0)\rangle = |\Psi_U(0)\rangle$, the above entails in particular that $|\Psi_U(t \to \infty)\rangle = (\hat{\mathbb{O}}(h)^*)^{-1} \cdot |\Psi(t \to \infty)\rangle$, i.e., the steady state of the CTMC generated by H_U and the one of the CTMC generated by H are related by an operator-valued rescaling (via the operator $(\hat{\mathbb{O}}(h)^*)^{-1}$, and thus in general in non-constant fashion). A similar construction will play a key role when studying rule-based discrete-time Markov chains.

Another interesting class of examples is motivated via the “Potsdam approach” to probabilistic graph transformation systems as advocated in [20].

Example 2 (LCM-construction)**.** Given a generalized CTMC with infinitesimal generator $H_{\underline{W}}$ specified as in Definition 4, construct the least common multiple (LCM) L of the inverse weight functions:

$$L := LCM(W_1, \dots, W_N)\,. \tag{11}$$

Then we define the *LCM-variant* of the CTMC as

$$H_{LCM} := H_{\underline{W}} \cdot L\,. \tag{12}$$

By construction, H_{LCM} does not contain any operator-valued inverse weights (in contrast to the original $H_{\underline{W}}$). Moreover, for certain choices of (inverse) weight functions $W_1, \dots, W_N$, the LCM-construction results in an infinitesimal generator H_{LCM} in which all contributing rules have the same input motif I, thus making contact with the methodology of [20].

3.3 Rule-based DTMCs

Despite their numerous applications in many different research fields, to date discrete-time Markov chains (DTMCs) that are based upon notions of probabilistic transformation systems have not been considered in quite a comparable detail as the CTMC constructions. Following [3], we present here a possible general construction of rule-based DTMC via a rule-algebraic approach:

Definition 5. For a suitable category $\mathbf{C}$, let $\mathscr{T} = \{(\gamma_j, R_j, W_j)\}_{j=1}^{N}$ be a (finite) set of triples of positive coefficients $\gamma_j \in \mathbb{R}_{>0}$, transformation rules $R_j \in \overline{\mathsf{Lin}}(\mathbf{C})$ and (inverse) weight functions $W_j \in End(\hat{\mathbf{C}})_{diag}$ ($j = 1, \dots, N$), with the additional constraint that

$$\sum_{j=1}^{N} \gamma_j \hat{\mathbb{O}}(\delta(R_j)) \cdot \frac{1}{W_j^*} = Id_{End(\hat{\mathbf{C}})}\,. \tag{13}$$

Then together with an initial state $|\Phi_0\rangle \in \mathsf{Prob}(\hat{\mathbf{C}})$, this data defines a *SqPO-type rule-based discrete-time Markov chain (DTMC)*, whose n-th state (for non-negative integer values of n) is given by

$$|\Phi_n\rangle := D^n\,|\Phi_0\rangle\,, \quad D := \sum_{j=1}^{N} \gamma_j \rho(\delta(R_j)) \cdot \frac{1}{W_j^*} \qquad (\rho \equiv \rho_{\mathbf{C}})\,. \tag{14}$$

Example 3 (Adaptive Voter Model)**.** As a typical example of a social network model, consider the specification of the *adaptive voter model (AVM)* in the variant according to Durrett et al. [ref], which has as its *input parameters* an *initial graph state* $|\Phi_0\rangle = |X_0\rangle$ (with $N_{\bullet}^{(0)}$ and $N_{\circ}^{(0)}$ vertices of types $\bullet$ and $\circ$, respectively, and with N_E undirected edges) and a *probability* $0 \leq \alpha \leq 1$, and whose transitions as depicted in Figure 3 are given via a form of a probabilistic decision procedure in several steps (black arrows): in each round of the AVM, an edge is chosen at random; if the edge is linking two nodes of different kind, with probability p, one of the rewiring rules is “fired”, or with probability $(1-p)$ one of the adopt rules, respectively; otherwise, i.e., if the chosen edge is linking two vertices of the same kind, no action is performed. As annotated in dark blue, each phase of this probabilistic decision procedure is dressed with a probability (and so that all probabilities for a given phase sum to 1). These probabilities in turn depend upon constant parameters (here, p and N_E), as well as on *pattern counts* $N_{\bullet}$,

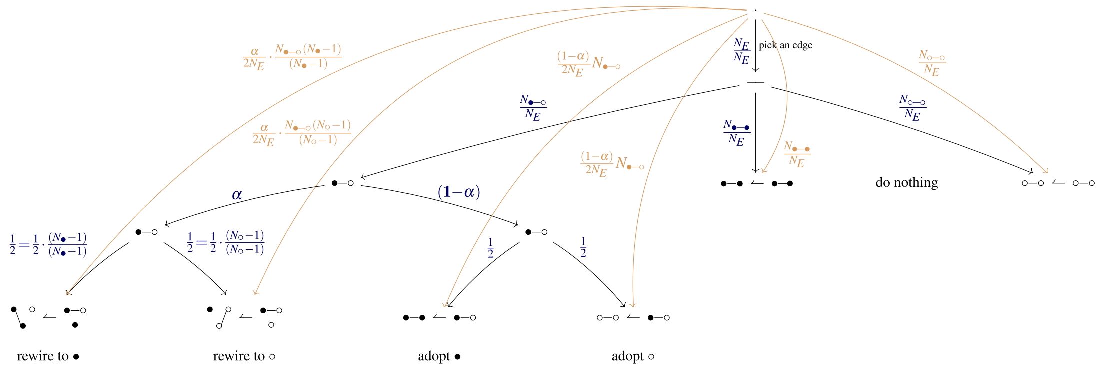

$\frac{\alpha}{2N_E}\cdot\frac{N_{\bullet-\circ}(N_\bullet-1)}{(N_\bullet-1)}$
$\frac{\alpha}{2N_E}\cdot\frac{N_{\bullet-\circ}(N_\circ-1)}{(N_\circ-1)}$
$\frac{(1-\alpha)}{2N_E}N_{\bullet-\circ}$
$\frac{(1-\alpha)}{2N_E}N_{\bullet-\circ}$
$\frac{N_E}{N_E}$
pick an edge
$\frac{N_{\circ-\circ}}{N_E}$
$\frac{N_{\bullet-\circ}}{N_E}$
$\frac{N_{\bullet-\bullet}}{N_E}$
$\frac{N_{\bullet-\bullet}}{N_E}$
$\frac{N_{\circ-\circ}}{N_E}$
do nothing
α
$(1-\alpha)$
$\frac{1}{2}=\frac{1}{2}\cdot\frac{(N_\bullet-1)}{(N_\bullet-1)}$
$\frac{1}{2}=\frac{1}{2}\cdot\frac{(N_\circ-1)}{(N_\circ-1)}$
$\frac{1}{2}$
$\frac{1}{2}$
rewire to •
rewire to ○
adopt •
adopt ○

Figure 3: Specification of the adaptive voter model (AVM) according to [11] via a probabilistic decision tree (black arrows), and as combined one-step probabilistic transitions (orange arrows).

$N_\circ$, $N_{\bullet-\bullet}$, $N_{\bullet-\circ}$ and $N_{\circ-\circ}$ that dynamically depend on the current system state. Note that $N_V := N_\bullet + N_\circ$ and $N_E = N_{\bullet-\bullet} + N_{\bullet-\circ} + N_{\circ-\circ}$ are *conserved quantities* in this model, since none of the transitions create or delete vertices, but at most exchange vertex types[4], and since the transitions manifestly preserve the overall number of edges.

Upon closer inspection, the transition probabilities may be written in a form that permits to compare the "firing" semantics to the mass-action and to the generalized semantics used in rule-based CTMCs. According to (6), we may implement the operation of counting patterns via linear operators based upon "identity rules", since for arbitrary graph patterns P and graph states $|X\rangle$,

$$N_P(X) = \langle|\,\hat{O}_P\,|X\rangle\,, \quad \hat{O}_P := \rho(\delta([P \leftarrow P \rightarrow P, \mathsf{true}]_\sim))\,. \tag{15}$$

Since rule algebra theory is not the main topic of the present paper, we provide below some operator relations without derivation[5], noting that they may be computed utilizing the fact that ρ is a so-called representation of the SqPO-type rule algebra (cf. [7, Thm. 3]):

$$\hat{O}_{\bullet-\circ}\hat{O}_x = \hat{O}_{\bullet-\circ\, x} + \hat{O}_{\bullet-\circ} \quad \Leftrightarrow \quad \hat{O}_{\bullet-\circ\, x} = \hat{O}_{\bullet-\circ}(\hat{O}_x - 1) \qquad (x \in \{\bullet, \circ\})\,, \tag{16}$$

which permit us to interpret the AVM model in the following way as a rule-based DTMC: starting by computing the overall one-step transitions and their respective probabilities (orange arrows in Figure 3),

[4] Strictly speaking, we implement the two vertex types • and ○ as attributes in our `SimSG` implementation, which in the theoretical setting can be emulated via using self-loops of two different types; evidently, this amounts merely to a slight modification of the type graph plus the enforcement of some structural constraints (compare [7]), thus for notational simplicity, we do not make this technical detail explicit in our diagrams.

[5] In fact, for the simple case at hand, one may derive the relations heuristically, noting that applying $\hat{O}_{\bullet-\circ}\hat{O}_x$ to some graph state amounts to first counting all vertices of type x (for $x \in \{\bullet, \circ\}$), followed by counting all •—○ edges; performing this operation in one step, this amounts to either counting the x-type vertices separately (i.e., counting the pattern •—○ x), or counting the x-type vertices on the same location as counting the •—○ patterns, thus explaining the two contributions in (16).

and using the above operator relations, we find the rule-based DTMC generator

$$D_{AVM} = \frac{\alpha}{2N_E}\rho\left(\delta\left(\begin{smallmatrix}\bullet & & \circ\\ & \diagdown & \\ & & \bullet\end{smallmatrix} \leftharpoonup \begin{smallmatrix}\bullet\!-\!\circ\\ \bullet\end{smallmatrix}\right)\right)\frac{1}{(\hat{O}_{\bullet}-1)^*} + \frac{\alpha}{2N_E}\rho\left(\delta\left(\begin{smallmatrix}\bullet & & \circ\\ & \diagup & \\ \circ & & \end{smallmatrix} \leftharpoonup \begin{smallmatrix}\bullet\!-\!\circ\\ \circ\end{smallmatrix}\right)\right)\frac{1}{(\hat{O}_{\circ}-1)^*} \tag{17a}$$

$$+\frac{(1-\alpha)}{2N_E}\rho\left(\delta\left(\bullet\!-\!\bullet \leftharpoonup \bullet\!-\!\circ\right)\right) + \frac{(1-\alpha)}{2N_E}\rho\left(\delta\left(\circ\!-\!\circ \leftharpoonup \bullet\!-\!\circ\right)\right) \tag{17b}$$

$$+\frac{1}{N_E}\rho\left(\delta\left(\bullet\!-\!\bullet \leftharpoonup \bullet\!-\!\bullet\right)\right) + \frac{1}{N_E}\rho\left(\delta\left(\circ\!-\!\circ \leftharpoonup \circ\!-\!\circ\right)\right). \tag{17c}$$

Interestingly, the particular semantics for the "firing" of transitions as encoding in this DTMC generator is neither a variant of mass-action semantics (wherein each rule would fire with a rate dependent on its number of matches, and so that the overall firing rate of all rules would be utilized to normalize all state-dependent firing rates to probabilities as described in Definition 5), nor do all transitions have state-independent, i.e., constant rates. Instead, the *rewiring rules* (17a) have rates proportional to $N_{\bullet\!-\!\circ}$; according to (16), we have that $N_{\bullet\!-\!\circ} = N_{\bullet\!-\!\circ\, x}/(N_X-1)^*$ (for $x \in \{\bullet, \circ\}$), whence the firing rates of the rewiring transitions of the AVM model are found to be proportional to a state-dependent fraction of their mass-action rates! In contrast, the *adoption* transitions (17b) as well as the *inactive* transitions (17c) are found to follow standard mass-action semantics, which might raise the question of whether the overall "mixed" semantics chosen for the AVM model indeed reflect the intended intuitions and empirical findings.

Finally, in view of simulation experiments, we would like to advocate the use of dedicated and high-performance Gillespie-style stochastic graph transformation software such as in particular `SimSG` [12] (cf. Section 4). More concretely, one may take advantage of the rule-algebraic formulation of DTMCs and CTMCs to identify for a given DTMC with generator D a particular CTMC based upon the *uniform* infinitesimal generator $H_U := D - Id_{End(\hat{\mathbf{C}})}$, so that (if both limits exist) the two alternative probabilistic systems have the same limit distribution (i.e., $|\Phi_\infty\rangle = \lim\limits_{t\to\infty}|\Psi(t)\rangle\rangle$):

$$D\,|\Phi_\infty\rangle = |\Phi_\infty\rangle \quad \Leftrightarrow \quad (D - Id_{End(\hat{\mathbf{C}})})\,|\Phi_\infty\rangle = 0 \quad \leftrightarrow \quad \left(\tfrac{d}{dt}\,|\Psi(t)\rangle\rangle\right)\Big|_{t\to\infty} = (H_U\,|\Psi(t)\rangle\rangle)\Big|_{t\to\infty} = 0 \tag{18}$$

However, CTMC simulations for rules that are not following mass-action semantics are non-standard to date, which is why it is also of interest to consider two particular mass-action type alternative CTMC models that are motivated by the original AVM model:

- A "standard" *mass-action semantics (MAS)* CTMC, where the transformation rules in D_{AVM} are defined to act in mass-action semantics (with the base rates $\kappa_\bullet, \kappa_\circ, \alpha_\bullet, \alpha_\circ \in \mathbb{R}_{>0}$ however not fixed in any evident way from the original AVM model interpretation):

  $$H_{MAS} = \kappa_\bullet \cdot \rho\left(\delta\left(\begin{smallmatrix}\bullet & & \circ\\ & \diagdown & \\ & & \bullet\end{smallmatrix} \leftharpoonup \begin{smallmatrix}\bullet\!-\!\circ\\ \bullet\end{smallmatrix}\right)\right) + \kappa_\circ \cdot \rho\left(\delta\left(\begin{smallmatrix}\bullet & & \circ\\ & \diagup & \\ \circ & & \end{smallmatrix} \leftharpoonup \begin{smallmatrix}\bullet\!-\!\circ\\ \circ\end{smallmatrix}\right)\right) \tag{19a}$$

  $$+\alpha_\bullet \cdot \rho\left(\delta\left(\bullet\!-\!\bullet \leftharpoonup \bullet\!-\!\circ\right)\right) + \alpha_\circ \cdot \rho\left(\delta\left(\circ\!-\!\circ \leftharpoonup \bullet\!-\!\circ\right)\right) \tag{19b}$$

  $$-\hat{O}_{\bullet\!-\!\circ}\left(\kappa_\bullet(\hat{O}_\bullet - 1) + \kappa_\circ(\hat{O}_\circ - 1) + \alpha_\bullet + \alpha_\circ\right). \tag{19c}$$

  Note that if we let $\rho(h_{MAS}) :=$ (19a) + (19b), then the terms in (19c) are indeed found to be equal to $-\hat{\mathbb{O}}(h_{AVM})$ (utilizing (16) to simplify terms) as required by Theorem 2.

- A *least-common-multiple (LCM)* variant which is computable from D_{AVM} via the method presented in Example 2 (with constant and operator-valued contributions to firing rates in orange):

$$\begin{aligned}
H_{LCM} = {} & \frac{\alpha}{2N_E}\rho\left(\delta\left(\begin{smallmatrix}\bullet & & \circ\\ & \diagdown & \\ & & \bullet\end{smallmatrix} \leftharpoonup \begin{smallmatrix}\bullet\!-\!\circ\\ \bullet\end{smallmatrix}\right)\right)(\hat{O}_{\circ}-1) + \frac{\alpha}{2N_E}\rho\left(\delta\left(\begin{smallmatrix}\bullet & & \circ\\ & \diagup & \\ \circ & & \end{smallmatrix} \leftharpoonup \begin{smallmatrix}\bullet\!-\!\circ\\ \circ\end{smallmatrix}\right)\right)(\hat{O}_{\bullet}-1) && \text{(20a)}\\
& + \frac{(1-\alpha)}{2N_E}\rho\left(\delta\left(\bullet\!-\!\bullet \leftharpoonup \bullet\!-\!\circ\right)\right)(\hat{O}_{\bullet}-1)(\hat{O}_{\circ}-1) && \\
& + \frac{(1-\alpha)}{2N_E}\rho\left(\delta\left(\circ\!-\!\circ \leftharpoonup \bullet\!-\!\circ\right)\right)(\hat{O}_{\bullet}-1)(\hat{O}_{\circ}-1) && \text{(20b)}\\
& - \frac{1}{N_E}\hat{O}_{\bullet-\circ}(\hat{O}_{\bullet}-1)(\hat{O}_{\circ}-1)\,. && \text{(20c)}
\end{aligned}$$

Expanding terms in H_{LCM} via utilizing once again the representation property of ρ (not presented here for brevity; cf. [7, Thm. 3] for the details), we may write H_{LCM} in the equivalent form below, which exhibits this CTMC model as a particular instance of a model for which all rules have the same input motif (and thus as a model in the spirit of the "Potsdam approach" as in [20]):

$$\begin{aligned}
H_{LCM} = {} & \frac{\alpha}{2N_E}\rho\left(\delta\left(\begin{smallmatrix}\bullet & \circ\\ | & \\ \bullet & \circ\end{smallmatrix} \leftharpoonup \begin{smallmatrix}\bullet\!-\!\circ\\ \bullet\ \ \circ\end{smallmatrix}\right)\right) + \frac{\alpha}{2N_E}\rho\left(\delta\left(\begin{smallmatrix}\bullet & \circ\\ & | \\ \bullet & \circ\end{smallmatrix} \leftharpoonup \begin{smallmatrix}\bullet\!-\!\circ\\ \bullet\ \ \circ\end{smallmatrix}\right)\right) && \text{(21a)}\\
& + \frac{1-\alpha}{2N_E}\rho\left(\delta\left(\begin{smallmatrix}\bullet\!-\!\bullet\\ \bullet\ \ \circ\end{smallmatrix} \leftharpoonup \begin{smallmatrix}\bullet\!-\!\circ\\ \bullet\ \ \circ\end{smallmatrix}\right)\right) + \frac{1-\alpha}{2N_E}\rho\left(\delta\left(\begin{smallmatrix}\circ\!-\!\circ\\ \bullet\ \ \circ\end{smallmatrix} \leftharpoonup \begin{smallmatrix}\bullet\!-\!\circ\\ \bullet\ \ \circ\end{smallmatrix}\right)\right) && \text{(21b)}\\
& - \frac{1}{N_E}\hat{O}_{\begin{smallmatrix}\bullet-\circ\\ \bullet\ \ \circ\end{smallmatrix}} && \text{(21c)}
\end{aligned}$$

4 Simulating the Models

In Section 3.3 we presented several ways of converting a DTMC-based model into a CTMC-based one. The first (M1) is based on introducing weight factors correcting for the deviation of the probabilistic view of the DTMC from the CTMC mass action semantics. This allows us to use a modified stochastic simulation algorithm to recover the behavior of the DTMC. Then we have the mass action stochastic graph transformation system (M2) as introduced in Section 2, in our view the most natural and closest to reality due to its use of continuous time, but in order to relate to the original probabilistic formulation we have to sample rate constants experimentally until the behavior matches that observed in the given model. Finally we consider a model (M3) obtained by mutually extending the rules of the system to the same left-hand side. Then all rules have the same number of matches and hence rates directly reflect probabilities.

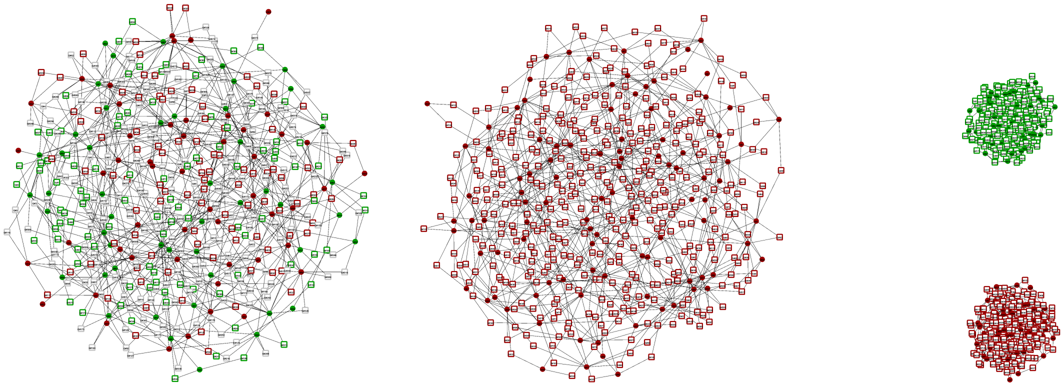
In this section we simulate these models with the CTMC-based SimSG tool [12] to see how, with suitably chosen parameters, their behavior matches that of the original formulation [11]. In particular, we want to answer the following questions.

RQ1: Can we reproduce the results of [11] by simulating (M1) in SimSG?

RQ2: What rates do we have to use to reproduce the behavior of [11] in (M2)?

RQ3: Does (M3) with rates reflecting the probabilities of [11] lead to the same overall behavior?

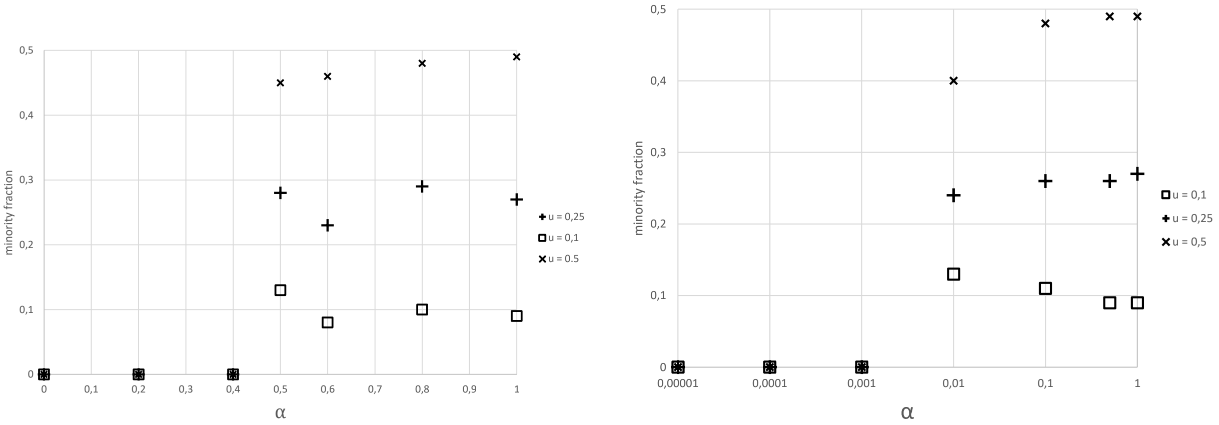
Note that in (M1) and (M2) rates are derived analytically based on the theory in Section 3.3 while they are determined experimentally for (M3). As in [11] initial graphs are generated randomly based on a fraction u of agents voting 0 (where $u = 0.5$ gives a 50/50 split between opinions) by deciding for each pair of nodes with probability v if they should be linked. This means that, with 100 voters and $v = 0.04$, we create



(a) Initial graph (u=0.5) (b) Giant component (c) Fragmented graph

Figure 4: Graphs before and after simulation

an Erdös-Rényi graph with 400 links and average degree 8. According to [11] we expect to see one of two behaviors, depending on the probability α of rewiring (vs. $1-\alpha$ for adoption) for a given discordant link. With $\alpha > 0.43$ rewiring causes a fragmentation of the graph into homogeneous connected components; otherwise adoption leads to the dominance of a single opinion (usually the initial majority) in a giant component. Visually we indicate opinion 1 in green and opinion 0 in red. Homogeneous components are highlighted in their respective color, whereas heterogeneous components are shown in gray. For example, Figure 4(a) shows the initial graph. In the limit, if the rates favor the adopt rules we obtain a graph as in Figure 4(b), while Figure 4(c) shows a possible result of dominant rewiring where the graph has split split into two components, one for each opinion.



(a) DTMC Version of the Voter Model (b) CTMC Version of the Voter Model

Figure 5: Simulation results

To answer RQ1 we use the rules as presented in Section 2, but modify the simulation algorithm by the weight factors derived in Section 3.3. This produces a simulation matching that of a discrete time Markov chain. While fixing the model size, we perform a nested parameter sweep over α, the static probability of rewiring, and over the fraction u of agents voting 0. The results are shown in Figure 5(a). On the y-axis we plot the fraction of voters with the minority opinion in the final graph, while the x-

axis shows the α value for each simulation. Furthermore, each set of symbols in the figure represents a different opinion ratio u in the initial graph. As we can see, independently of the u there is a clear point after which an increase in α leads to a separation of voters by opinion, which implies a segmentation of the graph. In turn, if α is low, the minority opinion disappears. Thus, Figure 5(a) reflects very closely the findings of [11], with only a minor deviation of the value of α separating the two outcomes.

For RQ2, we execute the same rules with our standard implementation simulation algorithm reflecting mass action CTMC semantics. While fixing the rate of the adopt rules as 1, we perform a nested parameter sweep over α, in this case denoting the rate of the rewiring rules, and the minority fraction u. The results are shown in Figure 5(b), again with the y-axis showing the fractions of voters in the minority in each final graph and the x-axis tracking the α values. As before, each set of symbols represents a different initial fraction u. In contrast to Figure 5(a), here the threshold of α at which the final graph becomes segmented is orders of magnitudes smaller. This is a result of the CTMC-based simulation algorithm, that obtains the rule application probability by multiplying a rule's match count with its static rate. Intuitively, the set of matches for each of the adopt rules consists of all connected pairs of voters $v1, v2$ of different opinion. Instead, the set of matches of the rewire rules is made up by the Cartesian product of this first set with the set of voters sharing the opinion of $v1$. To find the same balancing point as in the probabilistic model, this "unfair advantage" of the rewire rules is compensated for by a very low rate.

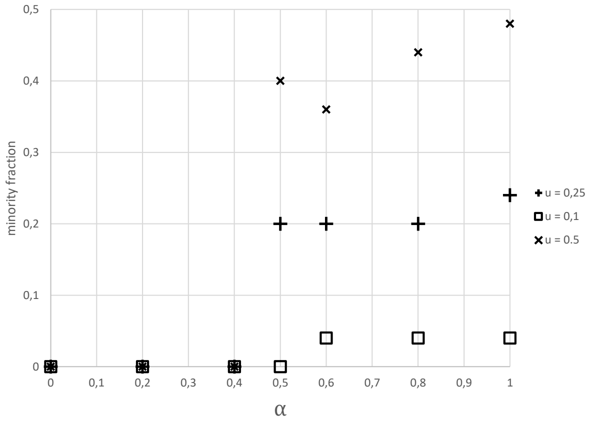


Figure 6: Simulation results of the alternative CTMC Version described by the rules in equation (21)

Finally, to address RQ3 we perform simulations using adopt and rewire rules extended to a common left-hand side consisting of two connected voters $v1, v2$ of conflicting opinions plus one extra voter each of opinion 0 and 1. By the same argument as above this leads to a large increase in the number of matches and hence poses scalability challenges to the simulation. To address this we used a smaller initial graph of 50 Voters and 200 groups and executed only 10 simulation runs per parameter configuration rather than 40 as in the other two models. (In each case, configurations range across 3 values for u and 7 values for α totaling 21 parameter configurations for each model.) The results are shown in Figure 6 using the same layout as before. It is interesting that, despite the high degree of activity in this model due to the very large match count, it displays almost the same behavior as the DTMC version (M1), even down to the α threshold.

The limited scalability of this approach is in part due to the global nature of the requirement that all rules share the same left-hand side. In the MDP-based approach of [20] this applies only to the subsets of rules defining the same action. One could then consider the four alternatives of rewire and adopt as cases of the same action of resolving a conflict edge, leading to a probabilistic graph transformation system with one rule. An analysis of the relation of mass-action CTMC with MDP-based semantics, however, is beyond the scope of this paper.

Social network analysis of the voter model and its variants is limited to a theoretical level, exploring mechanisms resulting in certain emergent phenomena that are observable in real-life networks, but are not quantified using real data. For a deeper quantitative analysis of phenomena such as the spread of opinions or the fragmentation of networks, the parameters of our rule-based models need to be matched to real social network data where such is available. This means, in particular, to determine the rates of the rules, e.g., from observations of pattern frequencies.

Such approaches are well established in chemical and biological modeling where statistical methods are used to derive rates of reaction rules from concentrations, i.e., relative pattern frequencies describing the ratios of the different types molecules [19, 13]. Instead, the derivation of an equational model from the operational rule-based description based on the rule algebra approach described in this paper could allow for an analytical approach where rates are determined by solving a system of equations. The precise conditions under which this is possible are a subject of future research.

5 Conclusion

In this paper we analyzed the non-standard semantics behind the formulation of adaptive system models in the literature. For a simple but prototypical example, we focused specifically on the voter models of opinion formation in social networks. We analyzed the non-standard semantics behind these models and established that they can be seen as Discrete-Time Markov Chains (DTMCs). In order to start the study of such models using the concepts, theories and tools of stochastic graph transformations (SGTS), we formalized the semantic relation between the rule-based specification by SGTS of mass-action Continuous-Time Markov Chains (CTMCs) with the DTMC-based semantics, identifying two systematic ways by which an SGTS can be derived from a DTMC-based model while preserving the behavior in the limit.

In the first derivation, this leads to a generalized notion of SGTS with weight factors correcting for the mass-action component represented by the dependency of the jump rate on the number of matches for each rule. This new type of SGTS is supported by a generalized simulation algorithm supporting the analysis of DTMC-based probabilistic graph transformation systems which, to our knowledge, is original. The second derivation converts the rules of the system by extending them to the same left-hand sides, producing a model resembling Markov-Decision Process (MDP)-based probabilistic graph transformation rules.

Apart from the theoretical analysis, we validate both resulting systems through simulations, establishing that they reproduce the expected behavior of the model. We also create a direct model of the same system as an SGTS with standard mass-action CTMC semantics and succeed in determining its parameters experimentally to match the expected behavior. We conclude that we can use standard mass-action stochastic GTS to model the phenomena expressed by the voter models in the literature, providing a starting point for further more elaborate modeling and analyses.

In future work we want to study more complex adaptive networks, including variations on the voter model with groups of more than two members, opinion profiles for a set of topics instead of a single opinion per voter, and including concepts such as influencers (who actively try to link to and persuade others), or zealots (who do not change their opinions). We are also planning to study how to match models to social network data. On the foundational side, we are planning optimizations to the simulation algorithm and a study of the relation between mass-action CTMC and MDP semantics of probabilistic graph transformations. We can also derive and study differential equations (ODEs) from our SGTS using the rule-algebra formalism. This provides a more scalable solution to analyzing their behavior complementing the simulation-based approach.

References

[1] Fabian Baumann, Philipp Lorenz-Spreen, Igor M. Sokolov & Michele Starnini (2020): *Modeling Echo Chambers and Polarization Dynamics in Social Networks*. *Phys. Rev. Lett.* 124, 048301, doi:10.1103/PhysRevLett.124.048301.

[2] Nicolas Behr (2019): *Sesqui-Pushout Rewriting: Concurrency, Associativity and Rule Algebra Framework*. In Rachid Echahed & Detlef Plump, editors: *Proceedings of theTenth International Workshop on Graph Computation Models (GCM 2019) in Eindhoven, The Netherlands*, *Electronic Proceedings in Theoretical Computer Science* 309, Open Publishing Association, pp. 23–52, doi:10.4204/eptcs.309.2.

[3] Nicolas Behr (2021): *On Stochastic Rewriting and Combinatorics via Rule-Algebraic Methods*. In Patrick Bahr, editor: *Proceedings 11th International Workshop on Computing with Terms and Graphs (TERMGRAPH 2020)*, 334, Open Publishing Association, pp. 11–28, doi:10.4204/eptcs.334.2.

[4] Nicolas Behr, Vincent Danos & Ilias Garnier (2016): *Stochastic mechanics of graph rewriting*. In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science - LICS '16*, ACM Press, pp. 46–55, doi:10.1145/2933575.2934537.

[5] Nicolas Behr, Vincent Danos & Ilias Garnier (2020): *Combinatorial Conversion and Moment Bisimulation for Stochastic Rewriting Systems*. *Logical Methods in Computer Science* Volume 16, Issue 3. doi:10.4204/EPTCS.323.4.

[6] Nicolas Behr & Jean Krivine (2021): *Compositionality of Rewriting Rules with Conditions*. *Compositionality* 3, doi:10.32408/compositionality-3-2.

[7] Nicolas Behr, Jean Krivine, Jakob L. Andersen & Daniel Merkle (2021): *Rewriting theory for the life sciences: A unifying theory of CTMC semantics*. *Theoretical Computer Science* 884, pp. 68–115, doi:10.1016/j.tcs.2021.07.026.

[8] Nicolas Behr & Pawel Sobocinski (2020): *Rule Algebras for Adhesive Categories (extended journal version)*. *Logical Methods in Computer Science* Volume 16, Issue 3. doi:10.23638/LMCS-16(3:2)2020.

[9] Paolo Bolzern, Patrizio Colaneri & Giuseppe De Nicolao (2020): *Opinion Dynamics in Social Networks: The Effect of Centralized Interaction Tuning on Emerging Behaviors*. *IEEE Transactions on Computational Social Systems* 7(2), pp. 362–372, doi:10.1109/TCSS.2019.2962273.

[10] Andrea Corradini, Tobias Heindel, Frank Hermann & Barbara König (2006): *Sesqui-Pushout Rewriting*. In A. Corradini, H. Ehrig, U. Montanari, L. Ribeiro & G. Rozenberg, editors: *Graph Transformations (ICGT 2006)*, *Lecture Notes in Computer Science* 4178, Springer Berlin Heidelberg, pp. 30–45, doi:10.1007/11841883_4.

[11] Richard Durrett, James P. Gleeson, Alun L. Lloyd, Peter J. Mucha, Feng Shi, David Sivakoff, Joshua E. S. Socolar & Chris Varghese (2012): *Graph fission in an evolving voter model*. *Proceedings of the National Academy of Sciences* 109(10), pp. 3682–3687, doi:10.1073/pnas.1200709109.

[12] Sebastian Ehmes, Lars Fritsche & Andy Schürr (2019): *SimSG: Rule-based Simulation using Stochastic Graph Transformation*. *J. Object Technol.* 18(3), pp. 1:1–17, doi:10.5381/jot.2019.18.3.a1.

[13] Hoda Eydgahi, William W Chen, Jeremy L Muhlich, Dennis Vitkup, John N Tsitsiklis & Peter K Sorger (2013): *Properties of cell death models calibrated and compared using Bayesian approaches*. *Molecular Systems Biology* 9(1), p. 644, doi:10.1038/msb.2012.69.

[14] Daniel T. Gillespie (1977): *Exact stochastic simulation of coupled chemical reactions*. *The Journal of Physical Chemistry* 81(25), pp. 2340–2361, doi:10.1021/j100540a008.

[15] Thilo Gross & Bernd Blasius (2008): *Adaptive coevolutionary networks: a review*. J. R. Soc. Interface 5(20), pp. 259–271, doi:10.1098/rsif.2007.1229.

[16] Reiko Heckel, Georgios Lajios & Sebastian Menge (2004): *Stochastic Graph Transformation Systems*. In Hartmut Ehrig, Gregor Engels, Francesco Parisi-Presicce & Grzegorz Rozenberg, editors: Graph Transformations, Springer Berlin Heidelberg, pp. 210–225, doi:10.1007/978-3-540-30203-2_16.

[17] David Kempe, Jon M. Kleinberg & Éva Tardos (2015): *Maximizing the Spread of Influence through a Social Network*. Theory Comput. 11, pp. 105–147, doi:10.4086/toc.2015.v011a004.

[18] Pascal P. Klamser, Marc Wiedermann, Jonathan F. Donges & Reik V. Donner (2017): *Zealotry effects on opinion dynamics in the adaptive voter model*. Phys. Rev. E 96, 052315, doi:10.1103/PhysRevE.96.052315.

[19] David J. Klinke (2014): *In silico model-based inference: A contemporary approach for hypothesis testing in network biology*. Biotechnology Progress 30, doi:10.1002/btpr.1982.

[20] Christian Krause & Holger Giese (2012): *Probabilistic Graph Transformation Systems*. In Hartmut Ehrig, Gregor Engels, Hans-Jörg Kreowski & Grzegorz Rozenberg, editors: Graph Transformations, Lecture Notes in Computer Science 7562, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 311–325, doi:10.1007/978-3-642-33654-6_21.

[21] Erhan Leblebici, Anthony Anjorin & Andy Schürr (2014): *Developing eMoflon with eMoflon*. In Davide Di Ruscio & Dániel Varró, editors: Theory and Practice of Model Transformations, Lecture Notes in Computer Science 8568, Springer International Publishing, Cham, pp. 138–145, doi:10.1007/978-3-319-08789-4_10.

[22] Weiyi Liu, Kun Yue, Hong Wu, Jin Li, Donghua Liu & Duanping Tang (2016): *Containment of competitive influence spread in social networks*. Knowl. Based Syst. 109, pp. 266–275, doi:10.1016/j.knosys.2016.07.008.

[23] Mark Newman (2008): *The physics of networks*. Physics Today 61(11), pp. 33–38, doi:10.1063/1.3027989.

[24] Jonathan M Read, Ken T.D Eames & W. John Edmunds (2008): *Dynamic social networks and the implications for the spread of infectious disease*. Journal of The Royal Society Interface 5(26), pp. 1001–1007, doi:10.1098/rsif.2008.0013.

[25] Fabián Riquelme, Pablo Gonzalez Cantergiani, Xavier Molinero & Maria J. Serna (2018): *Centrality measure in social networks based on linear threshold model*. Knowl. Based Syst. 140, pp. 92–102, doi:10.1016/j.knosys.2017.10.029.

[26] Dániel Varró, Gábor Bergmann, Ábel Hegedüs, Ákos Horváth, István Ráth & Zoltán Ujhelyi (2016): *Road to a reactive and incremental model transformation platform: three generations of the VIATRA framework*. Software & Systems Modeling 15(3), pp. 609–629, doi:10.1007/s10270-016-0530-4.

[27] Gergely Varró & Frederik Deckwerth (2013): *A Rete Network Construction Algorithm for Incremental Pattern Matching*. In Keith Duddy & Gerti Kappel, editors: Theory and Practice of Model Transformations (ICMT 2013), Lecture Notes in Computer Science 7909, Springer Berlin Heidelberg, pp. 125–140, doi:10.1007/978-3-642-38883-5_13.

[28] Gerd Zschaler, Gesa A. Böhme, Michael Seißinger, Cristián Huepe & Thilo Gross (2012): *Early fragmentation in the adaptive voter model on directed networks*. Phys. Rev. E 85, 046107, doi:10.1103/PhysRevE.85.046107.