CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-23914-1 — Gillespie Algorithms for Stochastic Multiagent Dynamics in Populations and Networks
Naoki Masuda , Christian L. Vestergaard
Excerpt
More Information

1 Introduction

We are compelled to understand and intervene in the dynamics of various complex systems in which different elements, such as human individuals, interact with each other. Such complex systems are often modeled by multiagent or network-based models that explicitly dictate how each individual behaves and influences other individuals. Stochastic processes are popular models for the dynamics of multiagent systems when it is realistic to assume random elements in how agents behave or in dynamical processes taking place in the system. For example, random walks have been successfully applied to describe locomotion and foraging of animals (Codling, Plank, & Benhamou, 2008; Okubo & Levin, 2001), dynamics of neuronal firing (Gabbiani & Cox, 2010; Tuckwell, 1988), and financial market dynamics (Campbell, Lo, & MacKinlay, 1997; Mantegna & Stanley, 2000) to name a few (see Masuda, Porter, and Lambiotte [2017] for a review). Branching processes are another major type of stochastic processes that have been applied to describe, for example, information spread (Eugster et al., 2004; Gleeson et al., 2021), spread of infectious diseases (Britton, 2010; Farrington, Kanaan, & Gay, 2003), cell proliferation (Jagers, 1975), and the abundance of species in a community (McGill et al., 2007) as well as other ecological dynamics (Black & McKane, 2012).

Stochastic processes in which the state of the system changes via discrete events that occur at given points in time are a major class of models for dynamics of complex systems (Andersson & Britton, 2000; Barrat, Barthélemy, & Vespignani, 2008; Daley & Gani, 1999; de Arruda, Rodrigues, & Moreno, 2018; Kiss, Miller, & Simon, 2017a; Liggett, 2010; Shelton & Ciardo, 2014; Singer & Spilerman, 1976; Van Mieghem, 2014). For example, in typical models for infectious disease spread, each infection event occurs at a given time t such that an individual transitions instantaneously from a healthy to an infectious state. Such processes are called *Markov jump processes* when they satisfy certain independence conditions (Hanson, 2007), which we will briefly discuss in Section 2.5. A jump is equivalent to a discrete event. In Markov jump processes, jumps occur according to *Poisson processes*. In this volume, we focus on how to simulate Markov jump processes. Specifically, we will introduce a set of exact and computationally efficient simulation algorithms collectively known as Gillespie algorithms. In the last technical section of this volume (i.e., Section 5), we will also consider more general, non-Markov, jump processes, in which the events are generated in more complicated manners than by Poisson processes. In the following text, we refer collectively to Markov jump processes and non-Markov jump processes as jump processes.

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-23914-1 — Gillespie Algorithms for Stochastic Multiagent Dynamics in Populations and Networks
Naoki Masuda , Christian L. Vestergaard
Excerpt
More Information

The Gillespie algorithms were originally proposed in their general forms by Daniel Gillespie in 1976 for simulating systems of chemical reactions (Gillespie, 1976), whereas several specialized variants had been proposed earlier; see Section 3.1 for a brief history review. Gillespie proposed two different variants of the simulation algorithm, the *direct method*, also known as Gillespie's *stochastic simulation algorithm* (SSA), or often simply *the Gillespie algorithm*, and the *first reaction method*. Both the direct and first reaction methods have found widespread use and in fields far beyond chemical physics. Furthermore, researchers have developed many extensions and improvements of the original Gillespie algorithms to widen the types of processes that we can simulate with them and to improve their computational efficiency.

The Gillespie algorithms are practical algorithms to simulate coupled Poisson processes exactly (i.e., without approximation error). Here "coupled" means that an event that occurs somewhere in the system potentially influences the likelihood of future events' occurrences in different parts of the same system. For example, when an individual in a population, v_i, gets infected by a contagious disease, the likelihood that a different healthy individual in the same population, v_j, will get infected in the near future may increase. If interactions were absent, it would suffice to separately consider single Poisson processes, and simulating the system would be straightforward.

We believe that the Gillespie algorithms are important tools for students and researchers that study dynamic social systems, where social dynamics are broadly construed and include both human and animal interactions, ecological systems, and even technological systems. While there already exists a large body of references on the Gillespie algorithms and their variants, most are concise, mathematically challenging for beginners, and focused on chemical reaction systems.

Given these considerations, the primary aim of this volume is to provide a detailed tutorial on the Gillespie algorithms, with specific focus on simulating dynamic social systems. We will realize the tutorial in the first part of the Element (Sections 2 and 3). In this part, we assume basic knowledge of calculus and probability. Although we do introduce stochastic processes and explain the Gillespie algorithms and related concepts with much reference to networks, we do not assume prior knowledge of stochastic processes or of networks. To understand the coding section, readers will need basic knowledge of programming. The second part of this Element (Sections 4 and 5) is devoted to a survey of recent advancements of Gillespie algorithms for simulating social dynamics. These advancements are concerned with accelerating simulations and/or increasing the realism of the models to be simulated.

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-23914-1 — Gillespie Algorithms for Stochastic Multiagent Dynamics in Populations and Networks
Naoki Masuda , Christian L. Vestergaard
Excerpt
More Information

2 Preliminaries

We review in this section mathematical concepts needed to understand the Gillespie algorithms. In Sections 2.1 to 2.3, we introduce the types of models we will be concerned with, namely *jump processes*, and in particular a simple type of jump process termed *Poisson processes*. In Sections 2.4 to 2.6, we derive the main mathematical properties of Poisson processes. The concepts and results presented in Sections 2.1 to 2.6 are necessary for understanding Section 3, where we derive the Gillespie algorithms. In Sections 2.7 and 2.8, we review two simple methods for solving the models that predate the Gillespie algorithms and discuss some of their shortcomings. These two final subsections motivate the need for exact simulation algorithms such as the Gillespie algorithms.

2.1 Jump Processes

Before getting into the nitty-gritty of the Gillespie algorithms, we first explore which types of systems they can be used to simulate. First of all, with the Gillespie algorithms, we are interested in simulating a dynamic system. This can be, for example, epidemic dynamics in a population in which the number of infectious individuals varies over time, or the evolution of the number of crimes in a city, which also varies over time in general. Second, the Gillespie algorithms rely on a predefined and parametrized mathematical model for the system to simulate. Therefore, we must have the set of rules for how the system or the individuals in it change their states. Third, Gillespie algorithms simulate stochastic processes, not deterministic systems. In other words, every time one runs the same model starting from the same initial conditions, the results will generally differ. In contrast, in a deterministic dynamical system, if we specify the model and the initial conditions, the behavior of the model will always be the same. Fourth and last, the Gillespie algorithms simulate processes in which changes in the system are primarily driven by discrete events taking place in continuous time. For example, when a chemical reaction obeying the chemical equation $\mathrm{A} + \mathrm{B} \to \mathrm{C} + \mathrm{D}$ happens, one unit each of A and of B are consumed, and one unit each of C and of D are produced. This event is discrete in that we can count the event and say when the event happened, but it can happen at any point in time (i.e., time is not discretized but continuous).

We refer to the class of mathematical models that satisfy these conditions and may be simulated by a Gillespie algorithm as *jump processes*. In the remainder of this section, we explore these processes more extensively through motivating examples. Then, we introduce some fundamental mathematical definitions and results that the Gillespie algorithms rely on.

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-23914-1 — Gillespie Algorithms for Stochastic Multiagent Dynamics in Populations and Networks
Naoki Masuda , Christian L. Vestergaard
Excerpt
More Information

2.2 Representing a Population as a Network

Networks are an extensively used abstraction for representing a structured population, and Gillespie algorithms lend themselves naturally to simulate stochastic dynamical processes taking place in networks. In a network representation, each individual in the population corresponds to a node in the network, and edges are drawn between pairs of individuals that directly interact. What constitutes an interaction generally depends on the context. In particular, for the simulation of dynamic processes in the population, the interaction depends on the nature of the process we wish to simulate. For simulating the spread of an infectious disease, for example, a typical type of relevant interaction is physical proximity between individuals.

Formally, we define a network as a graph $G = (V, E)$, where $V = \{1, 2, \ldots, N\}$ is the set of nodes, $E = \{(u, v)\colon u, v \in V\}$ is the set of edges, and each edge (u, v) defines a pair of nodes $u, v \in V$ that are directly connected. The pairs (u, v) may be ordered, in which case edges are directed (by convention from u to v), or unordered, in which case edges are undirected (i.e., v connects to u if and only if u connects to v). We may also add weights to the edges to represent different strengths of interactions, or we may even consider graphs that evolve in time (so-called temporal networks) to account for the dynamics of interactions in a population.

We will primarily consider simple (i.e., static, undirected, and unweighted) networks in our examples. However, the Gillespie algorithms apply to simulated jump processes in all kinds of populations and networks. (For temporal networks, we need to extend the classic Gillespie algorithms to cope with the time-varying network structure; see Section 5.4.)

2.3 Example: Stochastic SIR Model in Continuous Time

We introduce jump processes and explore their mathematical properties by way of a running example. We show how we can use them to model epidemic dynamics using the stochastic susceptible-infectious-recovered (SIR) model.[1] For more examples (namely, SIR epidemic dynamics in metapopulation networks, the voter model, and the Lotka–Volterra model for predator–prey dynamics), see Section 3.4.

We examine a stochastic version of the SIR model in continuous time defined as follows. We consider a constant population of N individuals (nodes). At any time, each individual is in one of three states: susceptible (denoted by S;

[1] The SIR model was incidentally one of the first applications of a Gillespie-type algorithm in a 1953 article (Bartlett, 1953).

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-23914-1 — Gillespie Algorithms for Stochastic Multiagent Dynamics in Populations and Networks
Naoki Masuda , Christian L. Vestergaard
Excerpt
More Information

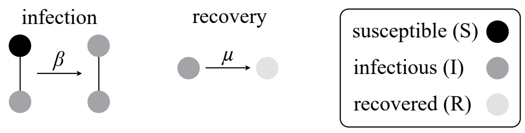

Figure 1 Rules of state changes in the SIR model. An infectious individual infects a susceptible neighbor at a rate β. Each infectious individual recovers at a rate μ.

meaning healthy), infectious (denoted by I), or recovered (denoted by R). The rules governing how individuals change their states are shown schematically in Fig. 1. An infectious individual that is in contact with a susceptible individual infects the susceptible individual in a stochastic manner with a constant *infection rate* β. Independently of the infection events, an infectious individual may recover at any point in time, with a constant *recovery rate* μ. If an infection event occurs, the susceptible individual that has been infected changes its state to I. If an infectious individual recovers, it transits from the I state to the R state. Nobody leaves or joins the population over the course of the dynamics. After reaching the R state, an individual cannot be reinfected or infect others again. Therefore, R individuals do not influence the increase or decrease in the number of S or I individuals. Because R individuals are as if they no longer exist in the system, the R state is mathematically equivalent to having died of the infection; once dead, an individual will not be reinfected or infect others.

We typically start the stochastic SIR dynamics with a single infectious individual, which we refer to as the *source* or *seed*, and $N_\mathrm{S} = N - 1$ susceptible individuals (and thus no recovered individuals). Then, various infection and recovery events may occur. The dynamics stop when no infectious individuals are left. In this final situation, the population is composed entirely of susceptible and/or recovered individuals. Since both infection and recovery involve an infectious individual, and there are no infectious individuals left, the dynamics are stuck. The final number of recovered nodes, denoted by N_R, is called the *epidemic size*, also known as the *final epidemic size* or simply the *final size*.[2] The epidemic size tends to increase as the infection rate β increases or as the recovery rate μ decreases. Many other measures to quantify the behavior of the SIR model exist (Pastor-Satorras et al., 2015). For example, we may be interested in the time until the dynamics terminate or in the speed at which the number of infectious individuals grows in the initial stage of the dynamics.

[2] The fraction N_R/N is typically also referred to as the epidemic size.

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE
Cambridge University Press & Assessment
978-1-009-23914-1 — Gillespie Algorithms for Stochastic Multiagent Dynamics in Populations and Networks
Naoki Masuda , Christian L. Vestergaard
Excerpt
More Information

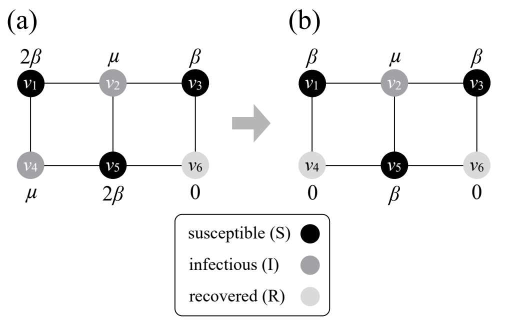

Figure 2 Stochastic SIR process on a square-grid network with six nodes. (a) Status of the network at an arbitrary time t. (b) Status of the network after v_4 has recovered. The values attached to the nodes indicate the rates of the events that the nodes may experience next.

Consider Fig. 2(a), where individuals are connected as a network. We generally assume that infection may only occur between pairs of individuals that are directly connected by an edge (called adjacent nodes). For example, the node v_4 can infect v_1 and v_5 but not v_3. The network version of the SIR model is fully described by the infection rate β, the recovery rate μ, the network structure, that is, which node pairs are connected by an edge, and the choice of source node to initialize the dynamics.

Mathematically, we describe the system by a set of coupled, constant-rate jump processes; constant-rate jump processes are known as *Poisson processes* (Box 1). Each possible event that may happen is associated to a Poisson process, that is, the recovery of each infectious individual is described by a Poisson process, and so is each pair of infectious and susceptible individuals where the former may infect the latter. The Poisson processes are coupled because an event generated by one process may alter the other processes by changing their rates, generating new Poisson processes, or making existing ones disappear. For example, after a node gets infected, it may in turn infect any of its susceptible neighbors, which we represent mathematically by adding new Poisson processes. This coupling implies that the set of coupled Poisson processes generally constitutes a process that is more complicated than a single Poisson process.

In the following subsections we develop the main mathematical properties of Poisson processes and of sets of Poisson processes. We will rely on these properties in Section 3 to construct the Gillespie algorithms that can simulate systems of coupled Poisson processes exactly. Note that the restriction to Poisson (i.e., constant-rate) processes is essential for the classic Gillespie

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-23914-1 — Gillespie Algorithms for Stochastic Multiagent Dynamics in Populations and Networks
Naoki Masuda , Christian L. Vestergaard
Excerpt
More Information

algorithms to work; see Section 5 for recent extensions to the simulation of non-Poissonian processes.

Box 1 Properties of Poisson Processes

A Poisson process is a jump process that generates events with a constant rate, λ.

Waiting-Time Distribution

The waiting times τ between consecutive events generated by a Poisson process are exponentially distributed. In other words, τ obeys the probability density

$$\psi(\tau) = \lambda e^{-\lambda\tau}. \tag{2.1}$$

Memoryless Property

The waiting time left until a Poisson process generates an event given that a time t has already elapsed since the last event is independent of t. This property is called the memoryless property of Poisson processes and is shown as follows:

$$\psi(t+\tau|t) = \frac{\psi(t+\tau)}{\Psi(t)} = \frac{\lambda e^{-\lambda(t+\tau)}}{e^{-\lambda t}} = \lambda e^{-\lambda\tau}, \tag{2.2}$$

where $\psi(t+\tau|t)$ represents the conditional probability density that the next event occurs a time $t+\tau$ after the last event given that time t has already elapsed; $\Psi(t) = \int_t^\infty \psi(\tau)\mathrm{d}\tau = e^{-\lambda t}$ is called the survival probability and is the probability that no event takes place for a time t. The first equality in Eq. (2.2) follows from the definition of the conditional probability. The second equality follows from Eq. (2.1).

Superposition Theorem

Consider a set of Poisson processes indexed by $i \in \{1, 2, \ldots, M\}$. The superposition of the processes is a jump process that generates an event whenever any of the individual processes does. It is another Poisson process whose rate is given by

$$\Lambda = \sum_{i=1}^{M} \lambda_i, \tag{2.3}$$

where λ_i is the rate of the ith Poisson process.

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-23914-1 — Gillespie Algorithms for Stochastic Multiagent Dynamics in Populations and Networks
Naoki Masuda , Christian L. Vestergaard
Excerpt
More Information

Box 1 (Continued)

Probability of a Given Process Generating an Event in a Superposition of Poisson Processes

Consider any given event generated by a superposition of Poisson processes. The probability Π_i that the ith individual Poisson process has generated this event is proportional to the rate of the ith process. In other words,

$$\Pi_i = \lambda_i/\Lambda. \tag{2.4}$$

2.4 Waiting-Time Distribution for a Poisson Process

We derive in this subsection the *waiting-time distribution* for a Poisson process, which characterizes how long one has to wait for the process to generate an event. It is often easiest to start from a discrete-time description when exploring properties of a continuous-time stochastic process. Therefore, we will follow this approach here. We use the recovery of a single node in the SIR model as an example in our development.

Let us partition time into short intervals of length δt. As δt goes to zero, this becomes an exact description of the continuous-time process. An infectious individual recovers with probability $\mu\delta t$ after each interval given that it has not recovered before.[3]

Formally, we define the SIR process in the limit $\delta t \to 0$. Then, you might worry that the recovery event is unlikely to ever take place because the probability with which it happens during each time-step, that is, $\mu\delta t$, goes toward 0 when the step size δt does so. However, this is not the case; because the number of time-steps in any given finite interval grows inversely proportional to δt, the probability to recover in finite time stays finite. For example, if we use a different step size $\overline{\delta t} = \delta t/10$, which is ten times smaller than the original δt, then the probability of recovery within the short duration of time $\overline{\delta t}$ is indeed 10 times smaller than $\mu\delta t$ (i.e., $= \mu\overline{\delta t}$). However, there are $\delta t/\overline{\delta t} = 10$ windows of size $\overline{\delta t}$ in one time window of size δt. So, we now have 10 chances for recovery to happen instead of only one chance. The probability for recovery to occur in any of these 10 time windows is equal to one minus the probability that it does not occur. The probability that the individual does not recover in time δt is equal

[3] To address a common misunderstanding, we emphasize that μ is a rate, not a probability, and thus can be larger than one. Note, however, that $\mu\delta t$ is a probability and thus cannot be greater than one.

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-23914-1 — Gillespie Algorithms for Stochastic Multiagent Dynamics in Populations and Networks
Naoki Masuda , Christian L. Vestergaard
Excerpt
More Information

to $(1-\mu\overline{\delta t})^{\delta t/\overline{\delta t}}$. Therefore, the probability that the individual recovers in any of the $\delta t/\overline{\delta t}$ windows is

$$p_{\mathrm{I}\to\mathrm{R}} = 1-(1-\mu\overline{\delta t})^{\delta t/\overline{\delta t}}. \tag{2.5}$$

Equation (2.5) does not vanish as we make $\overline{\delta t}$ small. In fact, the Taylor expansion of Eq. (2.5) in terms of $\overline{\delta t}$ yields $p_{\mathrm{I}\to\mathrm{R}} \approx (\delta t/\overline{\delta t}) \times \mu\overline{\delta t} = \mu\delta t$, where $\approx$ represents "approximately equal to". Therefore, to leading order, the recovery probabilities are the same between the case of a single time window of size δt and the case of $\delta t/\overline{\delta t}$ time windows of size $\overline{\delta t}$.

In the limit $\delta t \to 0$, the recovery event may happen at any continuous point in time. We denote by τ the waiting time from the present time until the time of the recovery event. We want to determine the probability density function (probability density or pdf for short) of τ, which we denote by $\psi_{\mathrm{I}\to\mathrm{R}}(\tau)$. By definition, $\psi_{\mathrm{I}\to\mathrm{R}}(\tau)\delta t$ is equal to the probability that the recovery event happens in the interval $[\tau, \tau+\delta t)$ for an infinitesimal δt (i.e., for $\delta t \to 0$). To calculate $\psi_{\mathrm{I}\to\mathrm{R}}(\tau)$, we note that the probability that the event occurs after $r = \tau/\delta t$ time windows, denoted by $p_{\mathrm{I}\to\mathrm{R}}(r)$, is equal to the probability that it did not occur during the first r time windows and then occurs in the $(r+1)$th window. This probability is equal to

$$p_{\mathrm{I}\to\mathrm{R}}(r) = (1-\mu\delta t)^r \times \mu\delta t = (1-\mu\delta t)^{\tau/\delta t}\mu\delta t. \tag{2.6}$$

The first factor on the right-hand side of Eq. (2.6) is the probability that the event has not happened before the $(r+1)$th window; it is simply equal to the probability that the event has not happened during a single window, raised to the power of r. The second factor is the probability that the event happens in the $(r+1)$th window. By applying the identity $\lim_{x\to 0}(1+x)^{1/x} = e$, known from calculus (see Appendix), with $x = -\mu\delta t$ to Eq. (2.6), we obtain the pdf of the waiting time as follows:

$$\begin{aligned}
\psi_{\mathrm{I}\to\mathrm{R}}(\tau) &= \lim_{\delta t\to 0}\frac{p_{\mathrm{I}\to\mathrm{R}}(\tau/\delta t)}{\delta t} \\
&= \mu \lim_{\delta t\to 0}(1-\mu\delta t)^{\tau/\delta t} \\
&= \mu\left[\lim_{\delta t\to 0}(1-\mu\delta t)^{1/(-\mu\delta t)}\right]^{-\mu\tau} \\
&= \mu e^{-\mu\tau}.
\end{aligned} \tag{2.7}$$

Equation (2.7) shows the intricate connection between the Poisson process and the exponential distribution: the waiting time of a Poisson process with rate μ (here, specifically the recovery rate) follows an exponential distribution with rate μ (Box 1). This fact implies that the mean time we have to wait for the

© in this web service Cambridge University Press & Assessment www.cambridge.org

CAMBRIDGE

Cambridge University Press & Assessment
978-1-009-23914-1 — Gillespie Algorithms for Stochastic Multiagent Dynamics in Populations and Networks
Naoki Masuda , Christian L. Vestergaard
Excerpt
More Information

recovery event to happen is $1/\mu$. The exponential waiting-time distribution actually completely characterizes the Poisson process. In other words, the Poisson process is the only jump process that generates events separated by waiting times that follow a fixed exponential distribution.

If we consider the infection process between a pair of S and I nodes in complete isolation from the other infection and recovery processes in the population, then exactly the same argument (Eq. (2.7)) holds true. In other words, the time until infection takes place between the two nodes is exponentially distributed with rate β, that is,

$$\psi_{\mathrm{S}\to\mathrm{I}}(\tau) = \beta e^{-\beta\tau}. \tag{2.8}$$

However, in practice the infection process is more complicated than the recovery process because it is coupled to other processes. Specifically, if another process generates an event before the infection process does, then Eq. (2.8) may no longer hold true for the infection process in question. For example, consider a node v_1 that is currently susceptible and an adjacent node v_2 that is infectious, as in Fig. 2. For this pair of nodes, two events are possible: v_2 may infect v_1, or v_2 may recover. As long as neither of the events has yet taken place, either of the two corresponding Poisson processes may generate an event at any point in time, following Eqs. (2.8) and (2.7), respectively. However, if v_2 recovers before it infects v_1, then the infection event is no longer possible, and so Eq. (2.8) no longer holds. We explore in the following two subsections how to mathematically deal with this coupling.

2.5 Independence and Interdependence of Jump Processes

Most models based on jump processes and most simulation methods, including the Gillespie algorithms, implicitly assume that different concurrent jump processes are independent of each other in the sense that the internal state of one process does not influence another. This notion of independence may be a source of confusion because a given process may depend on the events generated earlier by other processes, that is, the processes may be *coupled*, as we saw is the case for the infection processes in the SIR model. In this section, we sort out the notions of independence and coupling and what they mean for the types of jump processes we want to simulate. We will also explore another type of independence of Poisson processes, which is their independence of the past, called the *memoryless* property.

We can state the independence assumption as the condition that different processes are only allowed to influence each other by changing the state of the system. In other words, at any point in time each process generates an event

© in this web service Cambridge University Press & Assessment www.cambridge.org