Perfect Simulation Of Processes With Long Memory: a 'Coupling Into And From The Past' Algorithm

Aurélien Garivier

Perfect Simulation Of Processes With Long Memory: A "Coupling Into And From The Past" Algorithm

Aurélien Garivier

March 8, 2013

Abstract

We describe a new algorithm for the perfect simulation of variable length Markov chains and random systems with perfect connections. This algorithm generalizes Propp and Wilson's simulation scheme, and is based on the idea of coupling into and from the past. It improves on existing algorithms by relaxing the conditions required on the kernel and by accelerating convergence, even in the simple case of finite order Markov chains. Chains of variable or infinite order are an old object of consideration that raised considerable interest recently because of their applications in applied probability, from information theory to bio-informatics and linguistics.

Keywords: Perfect simulation, Context trees, Markov Chains of infinite order, Coupling From The Past (CFTP), Coupling Into And From The Past (CIAFTP).

1 Introduction

Since the seminal paper [21] by Propp and Wilson, perfect simulation schemes for stationary Markov chains have been developed and applied in several fields of applied probabilities, from statistical physics to Bayesian statistics (see for example [18] and references therein, or [12] for a gentle introduction).

In 2002, Comets et al. [5] have proposed an extension to processes with long memory: they provided a perfect simulation algorithm for stationary processes called random systems with complete connections [19, 20] or *chains of infinite order* [13]; these processes are characterized by a transition kernel which specifies, given an infinite sequence of past symbols, the probability distribution of the next symbol. The idea was, after [16, 17, 2], to exploit regenerative structures of these processes. Their algorithm relies on renewal properties, under a summable memory decay condition. As a by-product, the authors obtained the existence of stationary process for a given kernel, together with uniqueness properties under suitable hypotheses.

However, it appeared that these conditions on the kernel were quite restrictive, and actually not necessary. Foss et al. [8] and Gallo [9] showed that different

coupling schemes could be designed under alternative assumptions that do not even require the kernel to be continuous. Besides, the coupling scheme described in [5] strongly relies on regeneration, not on coalescence. Contrary to Propp and Wilson's algorithm, it does not converge for all mixing Markov chains, and when it does converge the number of steps required is larger. Recently, De Santis and Piccioni [7] tried to conciliate the two algorithms, by providing a hybrid method that works with two regimes: coalescence for short memory, and regeneration on long scales.

In this paper, we fill the gap between long and short scales, by providing a relatively elaborated and yet elegant coupling procedure that purely relies on coalescence. When the process considered is a (first order) Markov chain, this procedure simply boils down to Propp and Wilson's algorithm. But it permits to handle more general, infinite memory processes characterized by a continuous transition kernel, as defined in Section 2.

The idea is to exploit *local* coalescence, instead of global loss of memory properties. From an abstract point of view, the algorithm depicted in Section 3 simply consists in running a Markov chain on an infinite, uncountable set, until the first hit time of a given subset of states. Its concrete implementation involves a dynamical system on a set of labeled trees described in Section 4.

Another way to consider this algorithm is to link it to the algorithm described in [15]. In this article, Kendall explains how to adapt Propp and Wilson's idea in order to sample exactly from area-interaction point processes, by perfectly simulating the equilibrium distribution of a spacial birth-and-death process. As in the present article, the idea is that if all possible initial patterns at time $t < 0$ lead, following the same birth and death transitions, to the same configuration at time 0, then this configuration has the expected distribution; whenever such a coalescence is observable, perfect simulation is possible. This algorithm was later generalized by Wilson, who named it 'coupling into and from the past' (CIAFTP, see [25] Section 7), a term that fits very well to the algorithm described in Section 4.

We show that this perfect simulation scheme converges under less restrictive hypotheses than were required previously. As they prove very useful in many applications (e.g. information Theory [22, 24] or bio-informatics [4]), we detail the case of finite, but large order Markov chains (or Variable order Markov Chains, see [3]): our algorithm compares favorably with Propp and Wilson's algorithm on the extended chain in terms of computational complexity, and it compares favorably with the procedure of [5] in terms of convergence speed.

The paper is organized as follows: Section 2 presents the notation and definitions required in the sequel. Section 3 contains the conceptual description of the perfect simulation schemes. The key tool is an update rule constructed in Section 3.2. Section 4 contains the detailed description of the algorithm. Then, Section 5 gathers elements of complexity analysis, while Section 6 illustrates the weakness of the assumptions required for the algorithm to converge, in comparison to other coupling schemes. At the end of the paper, the Appendix gathers some proofs of technical results.

2 Notation and definitions

The statement of the results, of the algorithm, and the proofs, require the introduction of some notation, which is given in the following. A notation section is always somewhat off-putting, but a large part of it is quite standard and should be read rapidly. Some specific notions, especially regarding trees, are required in the following: even if they may seem somewhat unusual here, they are central in this paper and necessary in order to expose the algorithm as clearly as possible.

2.1 Histories

As in [5], we denote by G a finite alphabet, and we denote its size by $|G|$. For $k \in \mathbb{N}$, we denote by G^{-k} the set of all sequences $(w_{-k}, \dots, w_{-1})$, and $G^* = \cup_{k \geq 0} G^{-k}$. By convention, ε denotes the empty sequence, and $G^0 = \{\varepsilon\}$. The set of G-valued sequences index by the set of negative integers is denoted by $G^{-\mathbb{N}^+}$ and called the space of *histories*. For $-\infty \leq a \leq b < 0$ and $w \in G^{-\mathbb{N}^+}$, the sequence $(w_a, \dots, w_b)$ is denoted by $w_{a:b}$. An element $w_{-\infty:-1} \in G^{-\mathbb{N}^+}$ will be denoted by $\underline{w}$. For $w \in G^{-k}$, we note $|w| = k$ and for $\underline{w} \in G^{-\mathbb{N}^+}, |\underline{w}| = \infty$. For every negative integer n, we define the projection $\Pi^n : G^{-\mathbb{N}^+} \to G^n$ by $\Pi^n(\underline{w}) = w_{n:-1}$.

A *trie* is a rooted tree with edges labeled by elements of G. An element $\underline{w} \in G^{-\mathbb{N}^+}$ can be represented as a path in the infinite, complete trie, starting from the root, and successively following the edges labeled by $w_{-1}, w_{-2}, \dots$ A finite sequence $s \in G^*$ is represented by an internal node of this infinite trie. This representation is illustrated, in the case of the binary alphabet $G = \{0, 1\}$, in Figure 1.

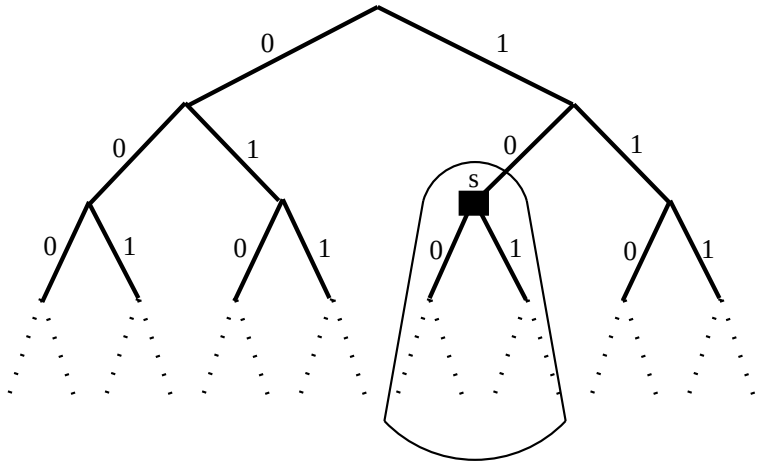

Figure 1: Trie representation of $G^{-\mathbb{N}^+}$ in the case of the binary alphabet $G = \{0, 1\}$. The square represents $s = (0, 1) \in G^{-2}$, and $\mathcal{T}(s)$ is circled. Note that the symbols $(s_{-2}, s_{-1}) = (0, 1)$ are to be read *from the bottom* (node s) *to the top* (root).

2.2 Concatenation, suffix

For two sequences $w_{a:b}$ and $z_{c:d}$, $-\infty \leq a \leq b < 0$, $-\infty < c \leq d < 0$, $w_{a:b}z_{c:d}$ denotes the concatenation of $w_{a:b}$ and $z_{c:d}$: $w_{a:b}z_{c:d} = (w_a, \ldots, w_b, z_c, \ldots, z_d)$. In particular, by taking $a = -\infty$, this defines the concatenation $\underline{z}s$ of a history $\underline{z}$ and an n-tuple $s \in G^{|s|}$. Note that this notation is different from the convention taken in [5]. If $a > b$, $w_{a:b}$ is the empty sequence ε.

Let $h \in G^* \cup G^{-\mathbb{N}^+}$. If $s \in G^*$ is such that $|h| \geq |s|$ and $h_{-|s|:-1} = s$, we say that s is a *suffix* of h and we denote $h \succeq s$, defining a partial order $\succeq$ on $G^* \cup G^{-\mathbb{N}^+}$.

2.3 Metric

Equipped the product topology, and with the ultra-metric distance δ defined by

$$\delta(\underline{w}, \underline{z}) = 2^{\sup\{k<0 : w_k \neq z_k\}},$$

$G^{-\mathbb{N}^+}$ is a complete and compact set. A ball $B \subset G^{-\mathbb{N}^+}$ is a set $\left\{\underline{z}s : \underline{z} \in G^{-\mathbb{N}^+}\right\}$ for some $s \in G^*$. In reference to the trie representation of $G^{-\mathbb{N}^+}$, we denote by $s = \mathcal{R}(B)$ the *root* of B, and by $\mathcal{T}(s) = B$ the *tail* of s (see Figure 1). Note that $\mathcal{T}(\varepsilon) = G^{-\mathbb{N}^+}$.

The set of probability distributions on G is denoted $\mathcal{M}(G)$, and is endowed with the total variation distance

$$|p - q|_{TV} = \frac{1}{2}\sum_{a \in G} |p(a) - q(a)| = 1 - \sum_{a \in G} p(a) \wedge q(a) \ ,$$

where $x \wedge y$ denotes the minimum of x and y.

2.4 Complete suffix Dictionaries

A (finite or infinite) set D of elements of G^* is called a *complete suffix dictionary* (CSD) if one of the following equivalent properties is satisfied:

- every sequence $\underline{w} \in G^{-\mathbb{N}^+}$ has a unique suffix in D:

$$\forall \underline{w} \in G^{-\mathbb{N}^+}, \exists! s \in D : \underline{w} \succeq s \ ;$$

- $\{\mathcal{T}(s) : s \in D\}$ is a partition of $G^{-\mathbb{N}^+}$; in that case, we write:

$$G^{-\mathbb{N}^+} = \bigsqcup_{s \in D} \mathcal{T}(s) \ .$$

A CSD can be represented as a trie, as illustrated in Figure 2. This representation suggests to define the *depth* of CSD D as the depth of this trie:

$$\mathrm{d}(D) = \sup\{|s| : s \in D\} \ .$$

Note that $\mathrm{d}(D) = +\infty$ if D is infinite. The smallest possible CSD is $\{\epsilon\}$ (its trie is reduced to the root): it has depth 0 and size 1. The second smallest is G, it has depth 1.

If a finite word $h \in G^*$ has a (unique) suffix in D, we write $h \succeq D$. If D and D' are two CSD such that $\forall s \in D', s \succeq D$, then we note $D' \succeq D$. This means that the trie representing D' entirely shadows that of D, as illustrated in Figure 2.

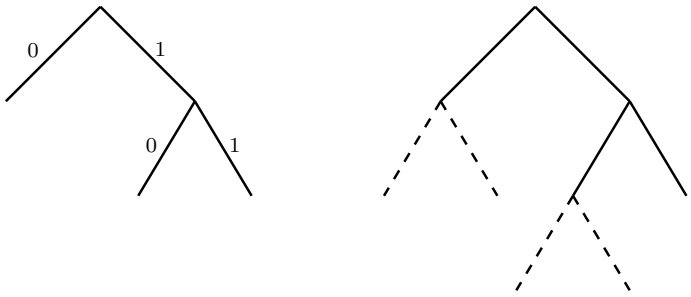

Figure 2: Trie representation of a CSD on the binary alphabet $G = \{0, 1\}$. Left: the trie representing the Complete Suffix Dictionary $D = \{0, 01, 11\}$. Right: $\{00, 10, 001, 101, 11\} \succeq \{0, 01, 11\}$.

2.5 Piecewise constant mappings

For a given CSD D, we say that a mapping f defined on $G^{-\mathbb{N}^+}$ is *D-constant* if

$$\forall s \in D, \forall \underline{w}, \underline{z} \in \mathcal{T}(s), f(\underline{w}) = f(\underline{z}) \ .$$

The mapping f is constant if and only if it is $\{\epsilon\}$-constant, and f is called *piecewise constant* if there exists a CSD D such that f is D-constant. For every $h \in G^*$ we define

$$f(h) = f\big(\mathcal{T}(h)\big) = \{f(z) : z \in \mathcal{T}(h)\} \ .$$

Note that, by definition, $f(h)$ is a set; however, if f is D-constant and if $h \succeq D$, then $f(h)$ is a singleton (that is, a set containing exactly one element).

Let f be a piecewise constant mapping; the set of all CSDs such that f is D-constant has a minimal element when ordered by the inclusion relation: we denote it D^f, the *minimal CSD* of f. The minimal CSD D^f is such that if $s \in D^f$, there exists $w \in G^*$ such that $s' = ws_{-|s|+1:-1} \in D^f$ and $f(s) \neq f(s')$. If f is D-constant, then D^f can be obtained by recursive pruning of D, that is, by pruning the nodes whose children are leaves with the same value for f as long as possible. A D-constant mapping f can be represented by the trie D, if each leaf s of D is labeled by the common value of the $f(\underline{w})$ for $w \in \mathcal{T}(s)$. Figure 3 illustrates the trie representation of a piecewise constant function, and pruning.

2.6 Probability transition kernels

A mapping $P : G^{-\mathbb{N}^+} \to \mathcal{M}(G)$ is called a *probability transition kernel*, and we denote the image of $\underline{w} \in G^{-\mathbb{N}^+}$ by $P(\cdot|\underline{w})$. We say that P is *continuous* if it is continuous as an application from $\left(G^{-\mathbb{N}^+}, \delta\right)$ to $(\mathcal{M}(G), |\cdot|_{TV})$. For $s \in G^*$, we define the *oscillation* of P on the ball $\mathcal{T}(s)$ as:

$$\eta_P(s) = \sup\Big\{ \big|P(\cdot|\underline{w}) - P(\cdot|\underline{z})\big|_{TV} : \underline{w}, \underline{z} \in \mathcal{T}(s) \Big\}.$$

We say that a process $(X_t)_{t\in\mathbb{Z}}$ with distribution ν on $G^{\mathbb{Z}}$ (equipped with the product topology and the product sigma-algebra) is *compatible* with kernel P

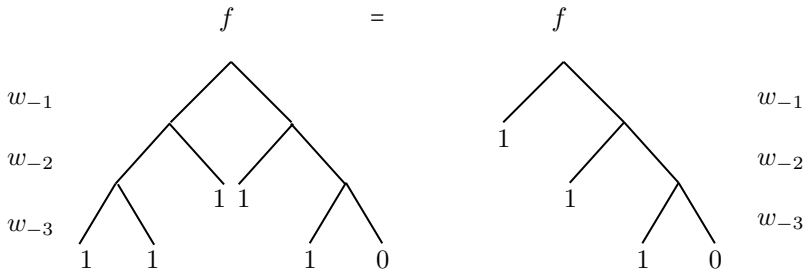

Figure 3: Two representations as labeled tries of the piecewise constant function f defined on the binary alphabet $\{0,1\}^{-\mathbb{N}^+}$ by $f(\underline{w}) = 0$ if $\underline{w} \in \mathcal{T}(111)$, and $f(\underline{w}) = 1$ otherwise. In the second representation, the trie is minimal: it has been obtained from the first trie by recursively pruning leaves with identical image.

if the latter is a version of the one-sided conditional probabilities of the former, that is:

$$\nu\left(X_i = g | X_{i+j} = w_j \ \forall j \in -\mathbb{N}^+\right) = P(g|\underline{w})$$

for all $i \in \mathbb{Z}, g \in G$ and ν-almost every $\underline{w}$. A classical but key remark is that $S_t = (\ldots, X_{t-1}, X_t), t \in \mathbb{Z}$, is a homogeneous Markov Chain on the compact ultra-metric state space $G^{-\mathbb{N}^+}$, with transition kernel Q given by:

$$\forall \underline{w}, \underline{z} \in G^{-\mathbb{N}^+}, \quad Q(\underline{z}|\underline{w}) = P(z_{-1}|\underline{w}) 1_{\bigcap_{i<0} z_{i-1}=w_i} \ .$$

2.7 Update rules

An application $\phi : [0,1[\times G^{-\mathbb{N}^+} \to G$ is called an *update rule* for a kernel P if, for all $\underline{w} \in G^{-\mathbb{N}^+}$ and for all $g \in G$, the Lebesgue measure of $\{u \in [0,1[: \phi(u, \underline{w}) = g\}$ is equal to $P(g|\underline{w})$. In other words, if U is a random variable uniformly distributed on $[0,1[$, then $\phi(U, \underline{w})$ has distribution $P(\cdot|\underline{w})$ for all $\underline{w} \in G^{-\mathbb{N}^+}$. For any continuous kernel P, Section 3.2 details the construction of an update rule ϕ_P such that:

$$\forall s \in G^*, 0 \le u < 1 - |G|\eta_P(s) \implies \phi_P(u, \cdot) \text{ is constant on } \mathcal{T}(s) \ . \qquad (1)$$

The following lemma (proved in the Appendix) is the basic observation that makes it possible to design an algorithm working in finite time even for kernels that are not piecewise continuous.

Lemma 1. *For all $u \in [0,1[$ the mapping $\underline{w} \to \phi_P(u, \underline{w})$ is continuous, i.e, piecewise constant.*

3 Abstract description of the perfect simulation scheme

Given a continuous transition kernel P, two questions arise:

1. does there exist a stationary distribution ν compatible with P? In that case, is it unique?

2. if ν exists, how can we sample finite trajectories from that distribution?

In the past decade, [5, 7, 9] have contributed to answer these questions. Their approach is to show that there exists a simulation scheme drawing samples of ν, and this algorithm is based on the idea of coupling from the past. Following these authors, we address these questions by constructing a new perfect simulation scheme that requires looser conditions on the kernel, and that converges faster than existing algorithms. In this section, we describe the general principle of this algorithm, ignoring practical details of implementation. These details are given in Section 4 below.

3.1 Perfect simulation by coupling into the past

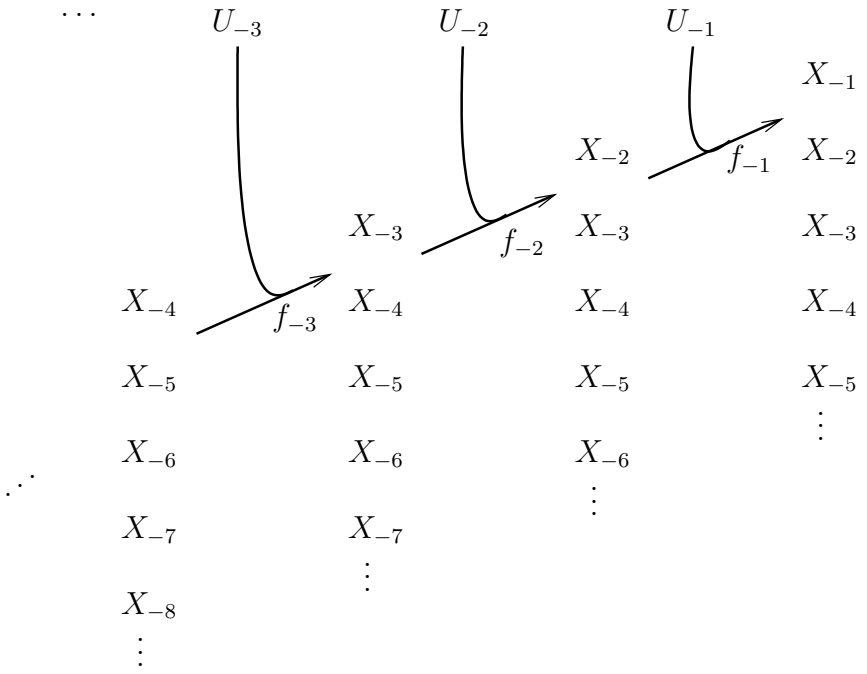

Figure 4: Perfect simulation scheme.

Let n be a negative integer. In order to draw $(X_n, \ldots, X_{-1})$ from a stationary distribution compatible with P, we use a semi-infinite sequence of independent random variables $(U_t)_{t<0}$ defined on a probability space $(\Omega, \mathcal{A}, P)$ and uniformly distributed on $[0, 1[$. The idea is to deduce X_t from U_t and from the past symbols $X_{t-1}, X_{t-2}, \ldots$ as depicted in Figure 4. Those past symbols are unknown, but the continuity of P makes it sometimes possible to compute X_t yet.

For each $t < 0$, let f_t be the random function $G^{-\mathbb{N}^+} \to G^{-\mathbb{N}^+}$ defined

by $f_t(\underline{w}) = \underline{w}\phi_P(U_t, \underline{w})$.[1] Beware the index shift: if $\underline{z} = f_t(\underline{w})$ then $z_{-1} = \phi_P(U_t, \underline{w})$ and $z_i = w_{i+1}$ for $i < -1$.

In addition, let $F_t = f_{-1} \circ \cdots \circ f_t$ and, for any negative integer n, $H_t^n = \Pi^n \circ F_t$. As will shall see in Proposition 1 below, the continuity of P implies that H_t^n is piecewise constant. Define

$$\tau(n) = \sup\{t < 0 : H_t^n \text{ is constant}\} ,$$

where by convention $\tau(n) = -\infty$ if for all $t < -1, H_t^n$ is not constant. When $\tau(n)$ is finite, the result of the procedure is $\{X_{n:-1}\}$, the image of the constant mapping $H_{\tau(n)}^n$: it is easily seen to have the expected distribution (see [21, 5]).

Remark 1. *For $t > n$, H_t^n cannot be constant, since for $n \leq k < t$, it holds that $(H_t^n(\underline{w}))_k = w_k$. Thus, $\tau(n) = \sup\{t \leq n : H_t^n \text{ is constant}\} \leq n$.*

Observe also that the sequence $(\tau(n))_n$ is a non-increasing sequence of stopping times with respect to the filtration $(\mathcal{F}_s)_s$, where $\mathcal{F}_s = \sigma(U_t : t \geq s)$, when s decreases.

From a theoretical point of view, this CIAFTP algorithm simply consists in running an instrumental Markov chain until a given hitting time. Indeed, the recursive definition given above shows that the sequence $(H_t^n)_{t \leq 1}$ is a homogeneous Markov chain on the set of functions $G^{-\mathbb{N}^+} \to G^n$. The algorithm terminates when this Markov chains hits the set of constant mappings. Such a procedure seems to be purely abstract, as it involves infinite, uncountable objects. But in Section 4, we show how this Markov chain on the set of functions $G^{-\mathbb{N}^+} \to G^n$ can be handled with a finite memory. Before we come to the detailed implementation of the algorithm, we present in Section 3.2 the construction of the update rule and we provide sufficient conditions for the stopping time $\tau(n)$ to be finite.

3.2 Construction of the update rule ϕ_P

The algorithm abstractly depicted above, and detailed in Section 4, crucially relies on the update rule ϕ_P that satisfies Equation (1). This section presents the construction of this update rule for a given, continuous kernel P. To put it in a nutshell, the construction of ϕ_P relies, for each k-tuple $z \in G^{-k}$, on a coupling of the conditional distributions $\big\{P(\cdot|\underline{z}) : \underline{z} \in \mathcal{T}(z)\big\}$. The simultaneous construction of all these couplings requires a few definitions and properties that are stated here and proved in the Appendix.

Provide G with any order $<$, so that $G^{-\mathbb{N}^+}$ can be equipped with the corresponding lexicographic order : $\underline{w} < \underline{z}$ if there exists $k \in -\mathbb{N}$ such that $\forall j > k, w_j = z_j$ and $w_k < z_k$. The continuity of P is locally quantified by some coupling factors that we define here together with coefficients that

[1] Regarding measurability issues: if the set $G^{-\mathbb{N}^+} \to G^{-\mathbb{N}^+}$ is equipped with the topology induced by the distance $\underline{\delta}$ defined by

$$\underline{\delta}(f_1, f_2) = \sum_{\underline{w} \in G^{-\mathbb{N}^+}} (2|G|)^{-|w|} \delta\big(f_1(w), f_2(w)\big) ,$$

and with the corresponding Borel sigma-algebra, then the measurability of f_t follows from Lemma 1.

are necessary for the construction of the update rule ϕ_P. For all $g \in G$, let $A_{-1}(\varepsilon) = a_{-1}(g|\varepsilon) = 0$; for all $k \in \mathbb{N}$ and all $z \in G^{-k}$, let

$$\begin{aligned} a_k(g|z_{-k:-1}) &= \inf\left\{P(g|\underline{w}) : \underline{w} \in \mathcal{T}\left(z_{-k:-1}\right)\right\}, \\ A_k(z_{-k:-1}) &= \sum_{g\in G} a_k(g|z_{-k:-1}), \\ A_k^- &= \inf_{s\in G^{-k}} A_k(s), \\ \alpha_k(g|z_{-k:-1}) &= A_{k-1}(z_{-k+1:-1}) + \sum_{h<g}\left\{a_k(h|z_{-k:-1}) - a_{k-1}(h|z_{-k+1:-1})\right\}, \end{aligned} \tag{2}$$

$$\beta_k(g|z_{-k:-1}) = A_{k-1}(z_{-k+1:-1}) + \sum_{h\le g}\left\{a_k(h|z_{-k:-1}) - a_{k-1}(h|z_{-k+1:-1})\right\}. \tag{3}$$

Note that, with our conventions, $a_0(g|\varepsilon) = \inf\{P(g|\underline{z}) : \underline{z} \in G^{-\mathbb{N}^+}\}$. Moreover, if $s, s' \in G^*$ are such that $s \succeq s'$, then for all $g \in G$ it holds that $a_k(g|s) \ge a_k(g|s')$, $A_k(s) \ge A_k(s')$ and the sequence $\left(A_k^-\right)_k$ is increasing.

The following propositions gather some elementary ideas that will be used in the sequel. They are proved in the Appendix.

Proposition 1. *The coupling factors of the kernel P satisfy the following inequalities: for all $s \in G^*$,*

$$1 - (|G| - 1)\eta_P(s) \le A_{|s|}(s) \le 1 - \eta_P(s). \tag{4}$$

Proposition 2. *The following assertions are equivalent:*

(i) the kernel P is continuous;

(ii) for every $\underline{w} \in G^{-\mathbb{N}^+}$, $\eta_P(w_{-k:-1})$ tends to 0 when k goes to infinity;

(iii) when k goes to infinity,

$$\sup\left\{\eta_P(s) : s \in G^{-k}\right\} \to 0;$$

(iv) $\forall \underline{w} \in G^{-\mathbb{N}^+}$, $A_k(w_{-k:-1}) \to 1$ as $k \to \infty$;

(v) $A_k^- \to 1$ as k goes to infinity.

Proposition 3. *Let P be a continuous kernel, and let $\alpha_k(\cdot|\cdot)$ and $\beta_k(\cdot|\cdot)$ be defined as in (2) and (3). Then, for every $\underline{w} \in G^{-\mathbb{N}^+}$,*

$$[0,1[= \bigsqcup_{g\in G, k\in\mathbb{N}} [\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})[\,.$$

In other words: for every $u \in [0,1[$ and every $\underline{w} \in G^{-\mathbb{N}^+}$, there exists a unique $k \in \mathbb{N}$ and a unique $g \in G$ such that $u \in [\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})[$.

Figure 5 illustrates Proposition 3 on a three-symbols alphabet. Thanks to Proposition 3, we can now define the following update rule and check that it satisfies Equation (1).

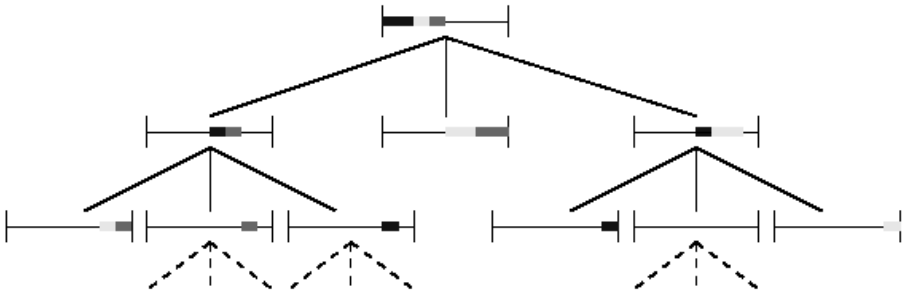

Figure 5: Graphical representation of an update rule ϕ_P on alphabet $\{0,1,2\}$: for each $w_{-k:-1}$, the intervals $[\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})[$ are represented in black ($g = 0$), light grey ($g = 1$) and medium grey ($g = 2$). For example, $P(1|1) = \alpha_0(1|\varepsilon) + \alpha_1(1|1) = 1/8 + 1/4$, and $P(0|00) = \alpha_0(0|\varepsilon) + \alpha_1(0|0) + \alpha_2(0|00) = 1/4 + 1/8 + 0$.

Definition 1. *Let $\phi_P : [0,1[\times G^{-\mathbb{N}^+} \to G$ be defined as follows:*

$$\phi_P(u, \underline{w}) = \sum_{g\in G, k\in\mathbb{N}} g 1_{[\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})[}(u) \ .$$

In words, for every $u \in [0,1[$ and for every $\underline{w} \in G^{-\mathbb{N}^+}$, $\phi_P(u, \underline{w})$ is the unique symbol $g \in G$ such that there exists $k \in \mathbb{N}$ satisfying:

$$u \in \big[\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})\big[\ .$$

Proposition 4. *The mapping ϕ_P of Definition 1 is an update rule satisfying Equation (1):*

$$\forall s \in G^*, \forall u \in [0,1], \forall \underline{w}, \underline{z} \in \mathcal{T}(s), \quad u < A_{|s|}(s) \implies \phi_P(u, \underline{w}) = \phi_P(u, \underline{z}) \ .$$

As a consequence, for every $s \in G^*$ and every $u < A_{|s|}(s)$, we can define $\phi_P(u, s)$ as the value common of the $\phi_P(u, \underline{w})$ for all $\underline{w} \in \mathcal{T}(s)$.

3.3 Convergence

In [5, 7], sufficient conditions on P are given in order to ensure that $\tau(n)$ is almost-surely finite (or even that $\tau(n)$ has bounded expectation). Besides, the authors prove that almost-sure finiteness for $\tau(n)$ is a sufficient condition to prove the existence and uniqueness of a stationary distribution ν compatible with P (see [5], Theorem 4.3 and corollaries 4.12 and 4.13). As a by-product, one obtains a simulation algorithm for sample paths of ν: if $U = (U_t)_{t\in\mathbb{Z}}$ is a sequence of independent, uniformly distributed random variables, one can define $\Phi : [0,1]^{\mathbb{Z}} \to G^{\mathbb{Z}}$ such that $\Phi(U)_t = \phi_P(U_{t-1}, \Phi(U)_{-\infty:t-1})$ for all t, and

$$\nu = \mathbb{P}(\Phi(U) \in \cdot),$$

the law of $\Phi(U)$, is stationary and compatible with P.

But the conditions on P required in [5] are quite restrictive: they require that

$$\sum_{m\geq 0}^{\infty}\prod_{k=0}^{m} A_k^- = \infty \ .$$

This condition requires in particular that the chain satisfies the Harris condition $A_0^- = \sum_{g\in G} \inf\left\{P(g|\underline{z}) : \underline{z} \in G^{-\mathbb{N}^+}\right\} > 0$. The authors prove that, under these conditions, the process regenerates, and that the stopping time

$$\tau'(n) = \sup\{t \leq n : H_t^t \text{ is constant}\}$$

is almost-surely finite, using a Kalikow-type decomposition of the kernel P as a mixture of Markov chains of all orders.

For $\tau(n)$ to be finite, this is obviously a sufficient but certainly not a necessary condition. Consider the example of order 1 Markov chains: while Propp and Wilson [21] have shown that the stopping time $\tau(n)$ of the optimal update rule is almost surely finite for every mixing chain (and, under some conditions, that $\tau(n)$ has the same order of magnitude as the mixing time of the chain), $\tau'(n)$ is almost surely infinite as soon as the Dobrushin coefficient A_0^- of the chain is 0. The contribution of this paper is precisely to fill the gap, by providing for general continuous kernels a Propp–Wilson procedure that may converge even if the process is not regenerating. In fact, for Markov chains of order 1, $\phi_P(u, \underline{w})$ depends only on w_{-1} and the algorithm presented in this paper is simply Propp and Wilson's exact sampling procedure.

Since the publication of [5], [9, 7] have generalized these results, relaxing the conditions on the kernel and proposing other particular conditions covering for different cases. Gallo [9] shows that P needs not be continuous to ensure the existence of ν, nor to ensure the finiteness of $\tau(n)$: he gives an example of a non-continuous regenerating chain (see also the final remark of Section 6). In [7], De Santis and Piccioni propose another algorithm which mixes the ideas of [5] and [21]: they propose a hybrid simulation scheme working with a Markov regime and a long-memory regime. Our approach is different and more general: we describe a single procedure that generalizes the sampling schemes of [5] and [21] in a single, unified framework.

4 The Coupling Into and From the Past Algorithm

In this section, we give a detailed description of the algorithm that permits to compute effectively the mappings H_t^n. The difficulty is that their domain is the infinite space $G^{-\mathbb{N}^+}$, so that no naive implementation is possible. The solution comes from the fact that, for each t, the mapping H_t^n is piecewise continuous, and thus can be represented by a random, but finite object: namely, by its trie representation defined in Section 2.5.

4.1 Description of the Algorithm

Consider a continuous kernel P and its update rule ϕ_P given by Definition 1. For each $u \in [0, 1[$, Proposition 1 shows that the mapping $\phi_P(u, \cdot)$ is piecewise constant; we denote by $D(u) = D^{\phi_P(u,\cdot)}$ its minimal CSD. Algorithm 1 shows how

the mappings H_t^n defined in Section 3 can be constructed recursively, using only finite memory. For simplicity, it is presented as a pseudo-code involving mathematical operations and ignoring specific data structures, but it is easy to deduce a real implementation from this pseudo-code. A matlab implementation is available on-line at `http://www.telecom-paristech.fr/~garivier/context/`. It contains a demonstration script illustrating the perfect simulation of the processes mentioned in Sections 5.2 and 6.

Algorithm 1: Coupling from and into the past for continuous kernels.

Input: update rule ϕ_P, size $-n$ of the path to sample

$t \leftarrow 0$;
$D_t^n \leftarrow G^n$;
$\forall s \in G^n, H_t^n(s) \leftarrow \{s\}$;
while $|D_t^n| > 1$ **do**
  $t \leftarrow t-1$;
  draw $U_t \sim \mathcal{U}([0,1[)$ independently;
  $D(U_t) \leftarrow$ the minimal trie of U_t;
  foreach $s \in D(U_t)$ **do**
    $\big\{g_t[s]\big\} \leftarrow \phi_P(U_t, s)$;
    if $sg_t[s] \succeq D_{t+1}^n$ **then**
      $E_t^n[s] \leftarrow \{s\}$;
    else
      $E_t^n[s] \leftarrow \big\{h \in G^* : hg_t[s] \in D_{t+1}^n\big(sg_t[s]\big)\big\}$;
  $E_t^n \leftarrow \bigcup_{s \in D(U_t)} E_t^n[s]$;
  Claim 1: E_t^n is a CSD;
  Claim 2: H_t^n is E_t^n-constant, and $\forall s \in E_t^n, H_t^n(s) = H_{t+1}^n\big(sg_t[s]\big)$ is a singleton;
  $D_t^n \leftarrow$ the minimal CSD of H_t^n obtained by pruning E_t^n

Output: $X_{n:-1}$ such that $\forall \underline{z} \in G^{-\mathbb{N}^+}, H_t^n(\underline{z}) = \{X_{n:-1}\}$

For every $t < 0$, the mapping H_t^n being piecewise constant, we can define $D_t^n = D^{H_t^n}$. Note that the definition of H_0^n in the initialization step is consistent with the general definition $H_t^n = \Pi^n \circ F_t$, as the natural definition of F_0 is the identity map on $G^{-\mathbb{N}^+}$. Algorithm 1 successively computes $H_{-1}^n, H_{-2}^n, \dots$ and stops for the first $t \le n$ such that H_t^n is constant.

The key step is the derivation of H_t^n and D_t^n from H_{t+1}^n, D_{t+1}^n and U_t: it is illustrated in Figure 6. It consists in three steps:

STEP 1: compute the minimal trie $D(U_t)$ of $\phi(U_t, \cdot)$.

STEP 2: compute the trie E_t^n such that H_t^n is E_t^n-constant, by completing $D(U_t)$ with portions of D_{t+1}^n. Namely, for every $s \in D(U_t)$, there are two cases:

- either $sg_t[s] \succeq D_{t+1}^n$, then knowing that $(X_{t-|s|}, \dots, X_{t-1}) = s$, together with U_t and H_t^n, is sufficient to determine $X_{n:-1}$ (see the dashed lines in Figure 6);

- or some additional symbols in the past are required by H^n_{t+1}, and a subtree of D^n_{t+1} has to be injected in the place of s (see the dotted circled subtree in Figure 6).

STEP 3: prune E^n_t in order to obtain the minimal trie D^n_t of H^n_t.

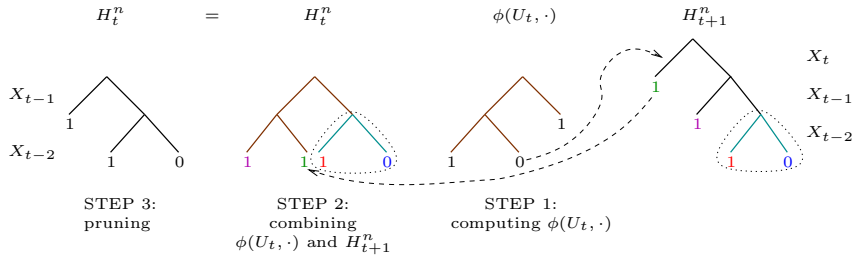

Figure 6: How to deduce the trie representation of H^n_t from that of H^n_{t+1} and from U_t, in three steps. Here, $G = \{0,1\}$ and $n = 1$. Recall that $\phi(U_t,\cdot)$ gives X_t in function of $(X_{t-1}, X_{t-2}, \dots)$, and that H^n_{t+1} gives X_{-1} in function of $(X_t, X_{t-1}, X_{t-2} \dots)$. Concerning Step 2: the dashed lines illustrates the first case (if $(X_{t-2}, X_{t-1}) = (0,1)$, then $X_t = \phi(U_t, \dots 01) = 0$ and $X_1 = H^n_{t+1}(\dots 010) = H^n_{t+1}(\dots 0) = 1$), while in the second case the subtree of D^n_{t+1} to be inserted into E^n_t is circled with a dotted line.

From a mathematical point of view, Algorithm 1 can be viewed as a run of an instrumental, homogeneous Markov chain on the set of $|G|$-ary trees whose leaves are labeled by G^n, which is stopped as soon as a tree of depth 0 is reached. One iteration of this chain, corresponding to one loop of of algorithm, is illustrated in Figure 6.

Algorithm 1 is thus very close to the high level method termed 'Coupling Into And From The Past' in [25] (see Section 7, in particular Figure 7). Indeed, in addition to coupling the trajectories starting *from* all possible states at past time t, one uses here a coupling of the conditional distributions before time t (that is, into the past). A small difference is that we want here to sample $X_{n:-1}$ and not only X_{-1}. The CSD D^n_t plays the role of the state denoted by X in [25], and H^n_t plays the role of F.

4.2 Correctness of Algorithm 1

Proving the correctness of Algorithm 1, that is, the correctness of the update rule deriving H^n_t from H^n_{t+1}, boils down to checking the two claims of lines 15 and 16.

Claim 1: E^n_t is a CSD.
Every $h \in E^n_t[s]$ is such that $h \succeq s$; let $\underline{w} \in \mathcal{T}(s)$, and let $\{hg_t[s]\} = D^n_{t+1}(\underline{w}g_t[s])$. Then $h \in E^n_t[s]$ and $\underline{w} \in \mathcal{T}(h)$, so that

$$\mathcal{T}(s) = \bigsqcup_{h \in E^n_t[s]} \mathcal{T}(h) \, .$$

The result follows, as $G^{-\mathbb{N}^+} = \sqcup_{s \in D(U_t)} \mathcal{T}(s)$.

Claim 2: H_t^n is E_t^n-constant, and $\forall s \in E_t^n, H_t^n(s) = H_{t+1}^n\big(sg_t[s]\big)$ is a singleton.
We prove that H_t^n is E_t^n-constant by induction on t, and the formula for $H_t^n(s)$ comes as a by-product of the proof. For $t = 0$, this is obvious if one denotes $E_0^n = D_0^n = G^n$. For $t < 0$, let $h \in E_t^n$. By construction $h \succeq D(U_t)$: denote by s the suffix of h in $D(U_t)$. Then $\phi_P(U_t, h)$ is the singleton $\{g_t[s]\}$. As, by construction, $hg_t[s] \succeq D_{t+1}^n$, $H_t^n(h) = H_{t+1}^n(hg_t[s])$ is a singleton by the induction hypothesis.

4.3 Computational complexity

For a given kernel, the random number of elementary operations performed by a computer during a run of Algorithm 1 is a complicated variable to analyze, as it depends not only on the number $\tau(n)$ of iterations, but also on the size of the trees D_t^n involved. Moreover, the basic operations on trees (traversal, lookup, node insertion or deletion, etc.) performed at each step have a computational complexity that depends on the implementation of the trees.

In first approximation, however, one can consider the cost of these operations to be a unit, so that the discussion on the computational complexity of the algorithm boils down to estimating (or bounding) their number in a run. Then a brief inspection of Algorithm 1 shows that the complexity of a run is proportional to the sum, for t from $\tau(n)$ to -1, of the number of nodes of D_t^n. Taking into account the complexity of the basic tree operations would typically lead to a complexity of order $O\left(\sum_{t=\tau(n)}^{-1} |D_t^n| \log |D_t^n|\right)$.

Thus, analyzing the computational complexity of Algorithm 1 amounts to bounding, at the same time, the number of iterations $\tau(n)$ and the size of the trees D_t^n. For a general kernel P, this seems to be a very challenging task that overpasses the ambition of this paper, involving not only the mixing properties of the corresponding process, but also the oscillation of the kernel directly. However, some elements of analysis are provided in the next section, where both questions are considered successively. First a crude bound on $\tau(n)$ is given; then, a bound on the size of D_t^n is proved for finite memory processes.

5 Bounding the size of D_t^n

We first give sufficient conditions for the algorithm to terminate, together with bound on the expectation of the depth of D_t^n. Then, we focus on the special but important case of (finite) variable length Markov Chains.

5.1 Almost sure termination of the coupling scheme

In general, the CSD D_t^n can be arbitrarily large with positive probability. In [5], conditions are given that ensure the finiteness of $\tau(n)$ defined above, from which bounds on D_t^n can be deduced. However, these conditions are quite restrictive: in particular, it is necessary that $A_0(\varepsilon) > 0$. [7] somewhat relaxes these conditions using an hybrid simulation scheme, allowing for $A_0(\varepsilon) = 0$.

A crude bound, ignoring the coalescence possibilities of the algorithm, is the following: denoting $L_t^n = \mathrm{d}(D_t^n)$ the depth of the current tree at time t, an

immediate inspection of Algorithm 1 yields:

$$\begin{cases} L_t^t \leq \max\{X_t, L_{t+1}^{t+1} - 1, 1\} & \text{if } t < n \text{ , and} \\ L_t^n \leq \max\{X_t, L_{t+1}^n - 1\} & \text{if } t \geq n \text{ .} \end{cases}$$

where the $X_t = \mathrm{d}(D(U_t))$ are i.i.d. random variables such that $\forall k \in \mathbb{N}, \mathbb{P}(X_t \leq k) = A_k^-$. Thus,

$$\mathbb{P}(L_t^n \leq k) \geq \mathbb{P}(L_{t+1}^n \leq k+1) A_k^- \geq \prod_{j=k}^{k-t-1} A_j^- \ .$$

This bound permits to show, as in [5], that $\tau(n)$ is almost-surely finite as soon as

$$\sum_{m \geq 0}^{\infty} \prod_{k=0}^{m} A_k^- = \infty \ .$$

Moreover, one obtains:

$$\mathbb{E}[L_t^n] \leq \sum_{k=1}^{n} \left(1 - \prod_{j=k}^{\infty} A_j^- \right) \ .$$

5.2 The case of Finite Context Trees

There is at least one case where it is easy to upper-bound the size of D_t^n independently of $t \leq n$: when the kernel P actually defines a *finite Context Tree*, that is, when the mapping $\underline{w} \to P(\cdot|\underline{w})$ is piecewise constant. In other words, denoting by D the minimal CSD of this mapping, $P(\cdot|s)$ is a singleton for each $s \in D$.

Even in that case, the simulation scheme described above is useful: although the "plain" Propp-Wilson algorithm could be applied on the first order Markov chain $(X_{t+1:t+\mathrm{d}(D)})_{t \in \mathbb{Z}}$ on the extended state space $G^{\mathrm{d}(D)}$, the computational complexity of such an algorithm might well become rapidly intractable if the depth $\mathrm{d}(D)$ is large, whereas the following property shows that our algorithm keeps a possibly much more limited complexity. Such qualities of *parsimony* are precisely the reason why finite context trees have proved successful in many applications, from Information Theory and universal coding (see [22, 24, 6, 11]) to biology ([1, 4]) or linguistics [10].

We say that a CSD D is *prefix-closed* if every prefix of any sequence in D is the suffix of an element of D:

$$\forall s \in D, \forall k \leq |s|, \exists w \in D : w \succeq s_{-|s|:-k} \ .$$

A prefix-closed CSD satisfies the following property:

Lemma 2. *If D is a prefix-closed CSD, then for all $h \in D$ (or, equivalently, for all $h \succeq D$) and for all $a \in G$, $ha \succeq D$.*

Proof: If $h \in G^*$ is such that for some $a \in G$, $ha \succeq D$ does not hold, then (as D is a CSD) there exists $s \in D$ and $s' \in G^* \backslash \{\epsilon\}$ such that $s = s'ha$. But then $s'h$ is a prefix of s and, by the prefix-closure property, there exists $w \in D$ such that $w \succeq s'h$. Thus, one cannot have $h \succeq D$.

We define the *prefix closure* $\overleftarrow{D}$ of a CSD D as the minimal prefix-closed set containing D, that is, the set of maximal elements (for the partial order $\succeq$) of

$$\tilde{D} = \left\{ s_{-|s|:-k} : s \in D,\ k \leq |s| \right\} .$$

In other words, $\overleftarrow{D}$ is the smallest set such that for all $w \in \tilde{D}$ there exists $s \in \overleftarrow{D}$ such that $s \succeq w$.

Obviously, $|\overleftarrow{D}| \leq |\tilde{D}| \leq |D| \times \mathrm{d}(D)$. This bound is pessimistic in general: many CSDs are already prefix-closed, and for most CSDs $|\overleftarrow{D}|$ is of the same order of magnitude as $|D|$. But in fact, for each positive integer n, one can show that there exists a CSD D of size n such that $|\overleftarrow{D}| \geq c|D|^2$ for some constant $c \approx 0.4$.

Now, assume that $D \neq \{\epsilon\}$, i.e., that P is not memoryless.

Proposition 5. *For each $t \leq n$, $\forall k < t$, $\overleftarrow{D} \succeq D_t^n$. Thus,*

$$|D_t^n| \leq \left|\overleftarrow{D}\right| \leq |D| \times \mathrm{d}(D) .$$

Proof: We show that $\overleftarrow{D} \succeq D_t^t$ by induction on t. First, as P is not memoryless, $\overleftarrow{D} \succeq D_{-1}^{-1} = G$. Second, assume that $\overleftarrow{D} \succeq D_{t+1}^n$: it is sufficient to prove that H_t^n (or H_t^t, if $t \geq n$) is $\overleftarrow{D}$-constant. Observing that $\overleftarrow{D} \succeq D$, for every $U_t \in [0,1[$ and for every $s \in \overleftarrow{D}$ it holds that $\phi_P(U_t, s)$ is a singleton $\{g_t[s]\}$. Using successively the lemma and the induction hypothesis, $sg_t[s] \succeq \overleftarrow{D} \succeq D_t^n$, thus $H_t^n(s) = H_{t+1}^n(sg_t[s])$ is also a singleton. Finally, for $t = n$, H_t^t is $\overleftarrow{D}$-constant because $\overleftarrow{D} \neq \{\epsilon\}$.

It has to be emphasized that Proposition 5 provides only an upper-bound on the size of D_t^n: in practice, D_t^n can be observed to be often much smaller. Even for non-sparse, large order Markov Chains of order d, Algorithm 1 can thus be faster than the Propp-Wilson algorithm on G^d which, in general, requires the consideration of $|G|^d$ states at each iteration. Interested readers may want to run the matlab experiments available at `http://www.telecom-paristech.fr/~garivier/context/`.

6 Example: a continuous process with long memory

This section briefly illustrates the strengths of Algorithm 1 in comparison with the other existing CFTP algorithms for infinite memory processes. It focuses on a process that cannot be simulated by other methods. Of course, Algorithm 1 is also relevant for all the processes mentioned in [5, 7], which we refer to for further examples.

The example we consider involves a non-regenerating kernel on the binary alphabet $G = \{0, 1\}$. It is such that $a_0 = 0$, and that the convergence of the coupling coefficients is slow, so that neither the perfect simulation scheme of [5], nor its improvement by [7] can be applied; yet, a probabilistic upper-bound on the stopping time τ of Algorithm 1 can be given, which proves that there exists a compatible stationary process. For all $k \geq 0$, let

$$P(0|01^k) = 1 - 1/\sqrt{k} . \tag{5}$$

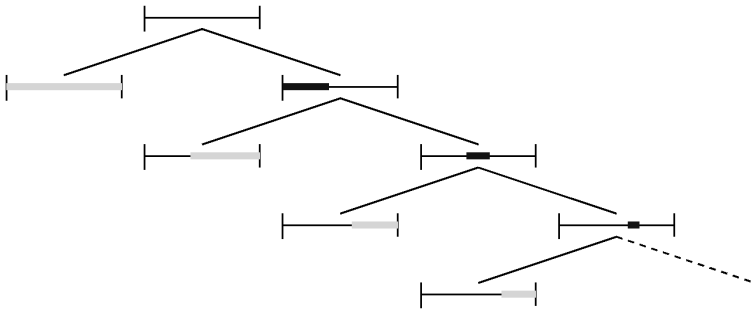

Figure 7: Graphical representation of the update rule for the kernel defined in (5) (dark is for 0, light grey is for 1)

The coupling coefficients of P are shown in Figure 7 Observe that $P(1|0) = \lim_{k\to\infty} P(0|01^k) = 1$, so that $a_0 = 0$. Besides, for $k \geq 0$ it holds that $A_{k+1} = A_k(01^k) = 1 - 1/\sqrt{k}$, so that

$$\sum_n \prod_{k=2}^{n} A_k^- < \infty$$

and the continuity conditions of [5, 7] do not apply.

We will show that the algorithm described above can be used to simulate samples of a process X have specification P (so that, in particular, such a process exists; uniqueness is straightforward). It is sufficient to show that the stopping time $\tau(1)$ is almost surely finite. Actually, $-\tau(1)$ is stochastically upper-bounded by three times a geometric variable of parameter $3/(2\sqrt{2}) - 1$. To simplify notations, denote $H_t = H_t^{-1}$, and $\underline{0} = (\ldots, 0, 0) \in G^{-\mathbb{N}^+}$. For every $t < -2$ if $U_{t-1} \leq 1 - 1/\sqrt{2}$, if $U_t > 1 - 1/\sqrt{2}$ and if $U_{t+1} \leq 1 - 1/\sqrt{2}$ then for every $\underline{w} \in G^{-\mathbb{N}^+}$ we have:

- $U_{t+1} \leq 1 - 1/\sqrt{2}$ implies $f_{t+1}(\underline{w}1) = \underline{w}10$ and $H_{t+1}(\underline{w}1) = H_{t+2}(\underline{0})$;
- $U_t > 1 - 1/\sqrt{2}$ implies $f_t(\underline{w}01) = \underline{w}011$ and $H_t(\underline{w}01) = H_{t+1}(\underline{w}011) = H_{t+2}(\underline{0})$ on the other hand, $f_t(\underline{w}0) = \underline{w}01$ and $H_t(\underline{w}0) = H_{t+1}(\underline{w}01) = H_{t+2}(\underline{0})$;
- $U_{t-1} \leq 1 - 1/\sqrt{2}$ implies $f_{t-1}(\underline{w}1) = \underline{w}10$ and $f_{t-1}(\underline{w}0) = \underline{w}01$, so that $H_{t-1}(\underline{w}0) = H_{t-1}(\underline{w}1) = H_{t+2}(\underline{0})$, and $\tau \geq t - 1$.

For every negative integer k let $E_k = \{U_{3k-1} \leq 1 - 1/\sqrt{2}\} \cap \{U_{3k} > 1 - 1/\sqrt{2}\} \cap \{U_{3k+1} \leq 1 - 1/\sqrt{2}\}$. The $(E_k)_{k<0}$ are independent events of probability $3/(2\sqrt{2}) - 1$, which gives the result.

Thus, the algorithm converges fast. However, the dictionaries involved in the simulation can be very large: in fact, it is easy to see that the depth X_t of $D(U_t)$ has no expectation: $\mathbb{P}(X_t \geq k) = 1/\sqrt{k}$. Of course, as D_t^n has a very special shape, ad hoc modifications of the algorithm would easily allow, here, to draw arbitrary long samples with low computational complexity. Moreover, paths of renewal processes can be simulated directly. Yet, this example illustrates the weakness of the conditions required by Algorithm 1, and the fact that neither

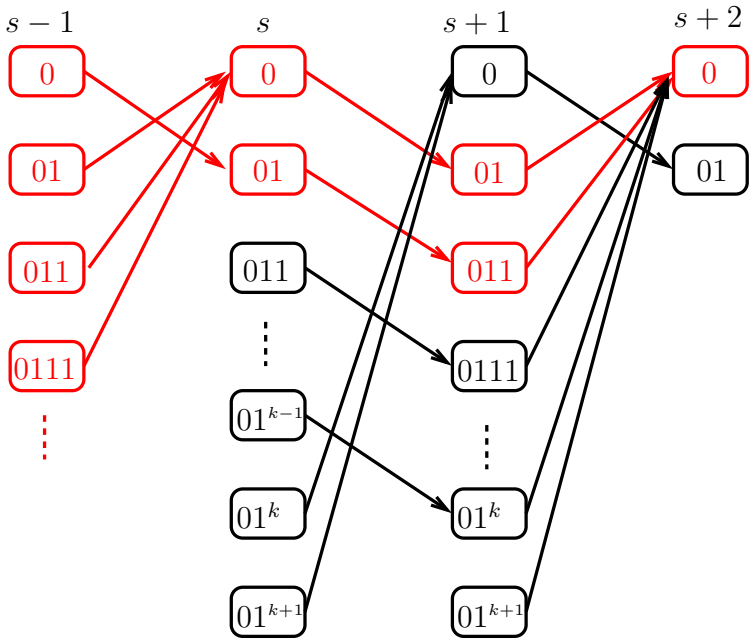

Figure 8: Convergence of the simulation scheme

regeneration, nor a rapid decreasing of the coupling coefficients are necessary conditions for perfect simulation. More complicated variants are easy to imagine where no other sampling method is known.

We conclude this section by the following remark: a simple modification of this example shows that continuity is absolutely not necessary to ensure convergence, as the proof also applies to any kernel P' such that for $0 \leq k \leq 2$, $P'(0|01^k) = 1 - 1/\sqrt{k}$, and for any $\underline{w} \in G^{-\mathbb{N}^+}, P'(0|\underline{w}11) \geq 1 - 1/\sqrt{2}$. Such a phenomenon has been studied in [9]: Gallo gives sufficient conditions on the shape of the trees, together with bounds on transition probabilities, that ensure convergence of his coupling scheme. However, his approach is quite different and does not cover the examples presented here.

Acknowledgments

I thank the referees of the paper for their very useful help for improving the redaction of this paper, and for pointing me to Kendall's 'Coupling From and Into The Past' method (the name is by Wilson). I also thank Sandro Gallo, Antonio Galves and Florencia Leonardi (Numec, Sao Paulo) for stimulating discussions on chains of infinite memory. This work was supported by USP-COFECUB (grant 2009.1.820.45.8).

References

[1] G. Bejerano and G. Yona. Variations on probabilistic suffix trees: statistical modeling and prediction of protein families. *Bioinformatics*, 17(1):23–43, 2001.

[2] Henry Berbee. Chains with infinite connections: uniqueness and Markov representation. *Probab. Theory Related Fields*, 76(2):243–253, 1987.

[3] P. Bühlmann and A. J. Wyner. Variable length Markov chains. *Ann. Statist.*, 27:480–513, 1999.

[4] J. R. Busch, P. A. Ferrari, A. G. Flesia, R. Fraiman, S. P. Grynberg, and F. Leonardi. Testing statistical hypothesis on random trees and applications to the protein classification problem. *Annals of applied statistics*, 3(2), 2009.

[5] Francis Comets, Roberto Fernández, and Pablo A. Ferrari. Processes with long memory: regenerative construction and perfect simulation. *Ann. Appl. Probab.*, 12(3):921–943, 2002.

[6] I. Csiszár and Z. Talata. Context tree estimation for not necessarily finite memory processes, via BIC and MDL. *IEEE Trans. Inform. Theory*, 52(3):1007–1016, 2006.

[7] Emilio De Santis and Mauro Piccioni. Backward coalescence times for perfect simulation of chains with infinite memory. *J. Appl. Probab.*, 49(2):319–337, 2012.

[8] S. G. Foss, R. L. Tweedie, and J. N. Corcoran. Simulating the invariant measures of Markov chains using backward coupling at regeneration times. *Probab. Engrg. Inform. Sci.*, 12(3):303–320, 1998.

[9] Sandro Gallo. Chains with unbounded variable length memory: perfect simulation and visible regeneration scheme. *J. Appl. Probab.*, 43(3):735–759, 2011.

[10] A. Galves, C. Galves, J. Garcia, N.L. Garcia, and F. Leonardi. Context tree selection and linguistic rhythm retrieval from written texts. *ArXiv: 0902.3619*, pages 1–25, 2010.

[11] A. Garivier. Consistency of the unlimited BIC context tree estimator. *IEEE Trans. Inform. Theory*, 52(10):4630–4635, 2006.

[12] Olle Häggström. *Finite Markov chains and algorithmic applications*, volume 52 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, 2002.

[13] Theodore E. Harris. On chains of infinite order. *Pacific J. Math.*, 5:707–724, 1955.

[14] Mark Huber. Fast perfect sampling from linear extensions. *Discrete Math.*, 306(4):420–428, 2006.

[15] Wilfrid S. Kendall. Perfect simulation for the area-interaction point process. In *Probability towards 2000 (New York, 1995)*, volume 128 of *Lecture Notes in Statist.*, pages 218–234. Springer, New York, 1998.

[16] S. P. Lalley. Regenerative representation for one-dimensional Gibbs states. *Ann. Probab.*, 14(4):1262–1271, 1986.

[17] S. P. Lalley. Regeneration in one-dimensional Gibbs states and chains with complete connections. *Resenhas*, 4(3):249–281, 2000.

[18] D. J. Murdoch and P. J. Green. Exact sampling from a continuous state space. *Scand. J. Statist.*, 25(3):483–502, 1998.

[19] Octav Onicescu and Gheorghe Mihoc. Sur les chaînes de variables statistiques. *Bull. Sci. Math.*, 59:174–192, 1935.

[20] Octav Onicescu and Gheorghe Mihoc. Sur les chaînes statistiques. *Comptes Rendus de l'Académie des Sciences de Paris*, 200:511–512, 1935.

[21] James Gary Propp and David Bruce Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. In *Proceedings of the Seventh International Conference on Random Structures and Algorithms (Atlanta, GA, 1995)*, volume 9, pages 223–252, 1996.

[22] J. Rissanen. A universal data compression system. *IEEE Trans. Inform. Theory*, 29(5):656–664, 1983.

[23] Walter R. Rudin. *Principles of Mathematical Analysis, Third Edition*. McGrawHill, 1976.

[24] F.M.J. Willems, Y.M. Shtarkov, and T.J. Tjalkens. The context-tree weighting method: Basic properties. *IEEE Trans. Inf. Theory*, 41(3):653–664, 1995.

[25] David Bruce Wilson. How to couple from the past using a read-once source of randomness. *Random Structures Algorithms*, 16(1):85–113, 2000.

Appendix

Proof of Lemma 1

let $u \in [0,1[$; the uniform continuity of P implies that there exists ϵ such that if $\delta(\underline{w},\underline{z}) \leq \epsilon$, then $\left|P(\cdot|\underline{w}) - P(\cdot|\underline{z})\right|_{TV} < (1-u)/|G|$. But then Equation (1) implies that $\phi_P(u,\underline{w}) = \phi_P(u,\underline{z})$.

Proof of Proposition 1

For the upper-bound, observe that

$$\begin{aligned}
\eta_P(s) &= \sup\left\{\left|P(\cdot|\underline{w}) - P(\cdot|\underline{z})\right|_{TV} : \underline{w}, \underline{z} \in \mathcal{T}(s)\right\} \\
&= \sup\left\{1 - \sum_{a\in G} P(a|\underline{w}) \wedge P(a|\underline{z}) : \underline{w}, \underline{z} \in \mathcal{T}(s)\right\} \\
&= 1 - \inf\left\{\sum_{a\in G} P(a|\underline{w}) \wedge P(a|\underline{z}) : \underline{w}, \underline{z} \in \mathcal{T}(s)\right\} \\
&\leq 1 - \sum_{a\in G} \inf\left\{P(a|\underline{w}) \wedge P(a|\underline{z}) : \underline{w}, \underline{z} \in \mathcal{T}(s)\right\} \\
&= 1 - \sum_{a\in G} \inf\left\{P(a|\underline{w}) : \underline{w} \in \mathcal{T}(s)\right\} \\
&= 1 - A_{|s|}(s) \ .
\end{aligned}$$

For the lower-bound, let $\epsilon > 0$, let $\underline{w} \in \mathcal{T}(s)$ and $b \in G$ be such that $\forall \underline{z} \in \mathcal{T}(s), \forall a \in G, P(a|\underline{z}) \geq P(b|\underline{w}) - \epsilon$. Then for all $z \in G^{-\mathbb{N}^+}$ and all $a \neq b, P(a|\underline{z}) \geq P(a|\underline{w}) - \eta_P(s)$ and one gets:

$$\begin{aligned}
&A_{|s|}(s) = \sum_{a\in G} P(a|\underline{w}) + \inf\left\{P(a|\underline{z}) - P(a|\underline{w}) : \underline{z} \in \mathcal{T}(s)\right\} \\
&\geq 1 + \inf\left\{P(b|\underline{z}) - P(b|\underline{w}) : \underline{z} \in \mathcal{T}(s)\right\} + \sum_{a\neq b} \inf\left\{P(a|\underline{z}) - P(a|\underline{w}) : \underline{z} \in \mathcal{T}(s)\right\} \\
&= 1 - \epsilon - (|G| - 1)\eta_P(s) \ ,
\end{aligned}$$

and, as ϵ is arbitrary, the result follows.

Proof of Proposition 2

The equivalence of (i) and (ii) is obvious by definition. The equivalence with (iii) is a simple consequence of Proposition 1. Similarly, (iii) follows from (i): if P is continuous on the compact set $G^{-\mathbb{N}^+}$, then it is uniformly continuous, and

$$\varphi(k) = \sup_{s\in G^{-k}} \eta_P(s) \to 0$$

as k goes to infinity. But by Proposition 1, $A_k^- \geq 1 - |G|\varphi(k)$. Finally, (iii) implies (ii).

The equivalence of (ii) and (iii) can also be proved as a consequence of Dini's theorem (see[23], Theorem 7.13 on page 150): defining $\tilde{A}_k(\underline{w}) = A_k(w_{-k:-1})$, the sequence $\left(\tilde{A}_k\right)_k$ is an increasing sequence of continuous functions simply converging to the (continuous) constant function 1, thus the convergence is uniform.

Proof of Proposition 3

Let m (resp. M) be the minimal (resp. maximal) element of G. Then, for all integer k, $\alpha_k(m|w_{-k:-1}) = A_{k-1}(w_{-k+1:-1})$, $\beta_k(M|w_{-k:-1}) = A_k(w_{-k:-1})$, and

$$[A_{k-1}(w_{-k+1:-1}), A_k(w_{-k:-1})[= \bigsqcup_{g\in G} [\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})[\ .$$

The result follows from the continuity assumption: $A_{-1}(\varepsilon) = 0$ and $A_k(w_{-k:-1}) \to 1$ as k goes to infinity.

Proof of Proposition 4

We need to prove that if $U \sim \mathcal{U}([0,1[)$, then for all $\underline{w} \in G^{-\mathbb{N}^+}$ the random variable $\phi_P(U, \underline{w})$ has distribution $P(\cdot|\underline{w})$. It is sufficient to prove that for all $g \in G$,

$$\sum_{l=0}^{\infty} \beta_l(g|w_{-l:-1}) - \alpha_l(g|w_{-l:-1}) = P(g|\underline{w}).$$

For all integer k, it holds that

$$\begin{aligned}\sum_{l=0}^{k} \beta_l(g|w_{-l:-1}) - \alpha_l(g|w_{-l:-1}) &= \sum_{l=0}^{k} a_l(g|w_{-l:-1}) - a_{l-1}(g|w_{-l+1:-1}) \\ &= a_k(g|w_{-k:-1}) \ .\end{aligned}$$

As an increasing sequence upper-bounded by $P(g|\underline{w})$, $a_k(g|w_{-k:-1})$ has a limit $Q(g|\underline{w}) \leq P(g|\underline{w})$ as k tends to infinity. By continuity,

$$\begin{aligned}\sum_{g\in G} Q(g|\underline{w}) &= \sum_{g\in G} \lim_{k\to\infty} a_k(g|w_{-k:-1}) \\ &= \lim_{k\to\infty} \sum_{g\in G} a_k(g|w_{-k:-1}) = \lim_{k\to\infty} A_k(w_{-k:-1}) = 1,\end{aligned}$$

and as $\sum_{g\in G} P(g|\underline{w}) = 1$ this implies that for all $g \in G$, $Q(g|\underline{w}) = P(g|\underline{w})$. The last part of the proposition is immediate: for $\underline{w}, \underline{z} \in \mathcal{T}(s), \forall k \leq |s|, w_{-k:-1} = z_{-k:-1}$ and

$$\begin{aligned}&\bigsqcup_{g\in G, k\leq|s|} [\alpha_k(g|w_{-k:-1}), \beta_k(g|w_{-k:-1})[\\ &\qquad = \bigsqcup_{g\in G, k\leq|s|} [\alpha_k(g|z_{-k:-1}), \beta_k(g|z_{-k:-1})[= [0, A_{|s|}(s)[\ .\end{aligned}$$