Computation of Substring Probabilities in Stochastic Grammars

Ana L. N. Fred

Instituto de Telecomunicações
Instituto Superior Técnico
IST-Torre Norte, Av. Rovisco Pais, 1049-001 Lisboa, Portugal
afred@lx.it.pt

Abstract. The computation of the probability of string generation according to stochastic grammars given only some of the symbols that compose it underlies pattern recognition problems concerning the prediction and/or recognition based on partial observations. This paper presents algorithms for the computation of substring probabilities in stochastic regular languages. Situations covered include prefix, suffix and island probabilities. The computational time complexity of the algorithms is analyzed.

1 Introduction

The computation of the probability of string generation according to stochastic grammars given only some of the symbols that compose it, underlies some pattern recognition problems such as prediction and recognition of patterns based on partial observations. Examples of this type of problems are illustrated in [2–4], in the context of automatic speech understanding, and in [5, 6], concerning the prediction of a particular physiological state based on the syntactic analysis of electro-encephalographic signals. Another potential application, in the area of image understanding, is the recognition of partially occluded objects based on their string contour descriptions.

Expressions for the computation of substring probabilities according to stochastic context-free grammars written in Chomsky Normal Form have been proposed [1–3]. In this paper algorithms for the computation of substring probabilities for regular-type languages expressed by stochastic grammars of the form

$$\sigma \rightarrow F_i \quad , \quad F_i \rightarrow \alpha \quad , \quad F_i \rightarrow \alpha F_j \quad , \quad \alpha \in \Sigma^* \quad , \quad \sigma, F_i, F_j \in V_N$$

(σ representing the grammar start symbol and V_N, Σ corresponding to the non-terminal and terminal symbols sets, respectively) are described. This type of grammars arises, for instance, in the process of grammatical inference based on Crespi-Reghizzi's method [7] when structural samples assume the form

$$[\ldots[dc[fgb[e[cd[ab]]]]$$

(meaning some sort of temporal alignment of sub-patterns).

The algorithms presented next explore the particular structure of these grammars, being essentially dynamic programming methods. Situations described include the recognition of fixed length strings (probability of exact recognition – section 3.1, highest probability interpretation – section 3.2) and arbitrary length strings (prefix – section 3.3, suffix – section 3.5 and island probabilities – section 3.4). The computational complexity of the methods is analyzed in terms of worst case time complexity. Minor and obvious modifications of the above algorithms enable the computation of probabilities according to the standard form of regular grammars.

2 Notation and Definitions

Let $\mathcal{G} = (V_N, \Sigma, R_s, \sigma)$ be a stochastic context-free grammar, where V_N is the finite set of non-terminal symbols or syntactical categories; Σ is the set of terminal symbols (vocabulary); R_s is a finite set of productions of the form $p_i : A \to \alpha$, $A \in V_N$, $\alpha \in (\Sigma \cup V_N)^*$, the star representing any combination of symbols in the set and p_i is the rule probability; and $\sigma \in V_N$ is the start symbol. When rules take the particular form $A \to aB$ or $A \to a$, with $A, B \in V_N$ and $a \in \Sigma$, then the grammar is designated as finite-state or regular.

Along the text the symbols A, G and H will be used to represent non-terminal symbols. The following definitions are also useful:

$w_1 \dots w_n \stackrel{\text{def}}{=}$ finite sequence of terminal symbols;

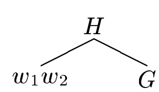
$H \stackrel{*}{\Rightarrow} \alpha \stackrel{\text{def}}{=}$ derivations of α from H through the application of an arbitrary number of rules;

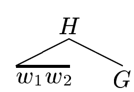
$C_T(\gamma) \stackrel{\text{def}}{=}$ number of terminal symbols in γ (with repetitions);

$C_N(\gamma) \stackrel{\text{def}}{=}$ number of non-terminal symbols in γ (with repetitions);

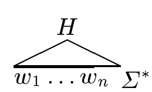
$n_{min(H,G)} \stackrel{\text{def}}{=} min\left\{n \ , \ \max_{\gamma} \ \{C_T(\gamma) : H \to \gamma G\}\right\}$

The following graphical notation is used to represent derivation trees:

– Arcs are associated with the direct application of rules. For instance, the rule $H \to w_1 w_2 G$ is represented by 

or 

– The triangle represents derivation trees having the top non-terminal symbol as root and leading to the string on the base of the triangle:

$H \stackrel{*}{\Rightarrow} w_1 \dots w_n \Sigma^* \equiv$ 

3 Algorithms Description

3.1 Probability of Derivation of Exact String – $\mathbf{Pr}(H \stackrel{*}{\Rightarrow} w_i \ldots w_{i+n})$

Let Pr $(H \stackrel{*}{\Rightarrow} w_i \ldots w_{i+n})$ be the probability of all derivation trees having H as root and generating exactly $w_i \ldots w_{i+n}$.

According to the type of grammars considered the computation of this probability can be obtained as follows (see figure 1):

$$\Pr(\sigma \stackrel{*}{\Rightarrow} w_i \ldots w_{i+n}) = \sum_G \Pr(\sigma \to G)\Pr(G \stackrel{*}{\Rightarrow} w_i \ldots w_{i+n}) \tag{1}$$

$$\begin{aligned} &\Pr(H \stackrel{*}{\Rightarrow} w_i \ldots w_{i+n}) \quad , \quad H \neq \sigma \\ &= \Pr(H \to w_i \ldots w_{i+n}) + \\ &\quad + \sum_G \sum_{k=1}^{n_{min(H,G)}} \Pr(H \to w_i \ldots w_{i+k-1} G)\Pr(G \stackrel{*}{\Rightarrow} w_{i+k} \ldots w_{i+n}) \end{aligned} \tag{2}$$

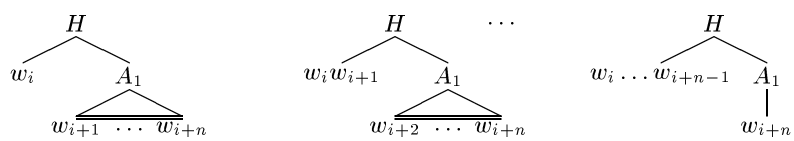


Fig. 1. Terms of the expression 2.

This can be summarized by the iterative algorithm:

1. For all non-terminal symbols $H \in V_N - \{\sigma\}$, determine

$$\Pr(H \to w_{i+n})$$

2. For $k = 1, \ldots n$ and for all non-terminal symbols $H \in V_N - \{\sigma\}$, compute:

$$\begin{aligned} &\Pr(H \stackrel{*}{\Rightarrow} w_{i+n-k} \ldots w_{i+n}) \\ &= \Pr(H \to w_{i+n-k} \ldots w_{i+n}) + \\ &\quad + \sum_G \sum_{j=1}^{n_{min(H,G)}} \Pr(H \to w_{i+n-k} \ldots w_{i+n-k+j-1} G) \times \\ &\quad \Pr(G \stackrel{*}{\Rightarrow} w_{i+n-k+j} \ldots w_{i+n}) \end{aligned} \tag{3}$$

3.

$$\Pr(\sigma \overset{*}{\Rightarrow} w_i \ldots w_{i+n}) = \sum_G \Pr(\sigma \to G)\Pr(G \overset{*}{\Rightarrow} w_i \ldots w_{i+n}) \qquad (4)$$

This corresponds to filling in the matrix of figure 2, column by column, from the right to the left. In this figure, each element corresponds to the probability of all derivation trees for the substring on the top of the column, with root on the non-terminal symbols indicated on the left of the row, i.e. Pr $(H_i \overset{*}{\Rightarrow} w_j \ldots w_{i+n})$.

	$w_i \ldots w_{i+n}$	$\ldots$	$w_{i+n-1} \ldots w_{i+n}$	w_{i+n}
H_1				Pr $(H_1 \to w_{i+n})$
H_2			Pr $(H_2 \overset{*}{\Rightarrow} w_{i+n-1} w_{i+n})$	
$\vdots$				
$H_{\lvert V_N \rvert}$				

Fig. 2. Array representing the syntactic recognition of strings according to the algorithm.

Notice that the calculations on step 2 are based only on direct rules and access to previously computed probabilities on columns to the right of the position under calculation.

Computational complexity analysis For strings of length n, the computational cycle is repeated n times, all non-terminal symbols being considered. Associating to the terms in equation 3

A: $\Pr(H \to w_{i+n-k} \ldots w_{i+n})$

B: $\displaystyle\sum_G \sum_{j=1}^{n_{min(H,G)}} \Pr(H \to w_{i+n-k} \ldots w_{i+n-k+j-1} G)\Pr(G \overset{*}{\Rightarrow} w_{i+n-k+j} \ldots w_{i+n})$

the worst case time complexity is of the order

$$O\left(\underbrace{(|V_N|-1)\delta}_{\text{step 1}} + \underbrace{\sum_{k=1}^{n-1}(|V_N|-1)\left(\overbrace{\underbrace{}_{\alpha}}^{\text{computation of A}} + \overbrace{\underbrace{}_{\beta}}^{\text{computation of B}}\right)}_{\text{step 2}} + \underbrace{\theta}_{\text{step 3}}\right) = O\left(|V_N| n\right)$$

3.2 Maximum Probability of Derivation – $P_m(H \stackrel{*}{\Rightarrow} w_i \ldots w_{i+n})$

Let $\mathrm{P}_m(H \stackrel{*}{\Rightarrow} w_1 \ldots w_n)$ denote the highest probability over all derivation trees having H as root and producing exactly $w_1 \ldots w_n$ and define the matrix M_u as follows:

$$M_u[i,j] = \mathrm{P}_m(H_i \stackrel{*}{\Rightarrow} w_j \ldots w_n) \qquad i = 1, \ldots, |V_N| \quad j = 1, \ldots, n \tag{5}$$

Observing that

$$\mathrm{P}_m(\sigma \stackrel{*}{\Rightarrow} w_1 \ldots w_n) = \max_G \left\{ \mathrm{P}_m(G \stackrel{*}{\Rightarrow} w_1 \ldots w_n) \right\} \tag{6}$$

and

$$\begin{aligned} &\mathrm{P}_m(H \stackrel{*}{\Rightarrow} w_1 \ldots w_n) \quad , \quad H \neq \sigma \\ &= \max \{ \Pr(H \rightarrow w_1 \ldots w_n) \ , \\ &\quad \max_{i,G} \left\{ \Pr(H \rightarrow w_1 \ldots w_i G) P_m(G \stackrel{*}{\Rightarrow} w_{i+1} \ldots w_n) \right\} \Big\} \end{aligned} \tag{7}$$

the following algorithm computes the desired probability:

1. For $i = 1, \ldots, |V_N| \quad , H_i \neq \sigma$

$$M_u[i,n] = \begin{cases} \Pr(H_i \rightarrow w_n) & \text{if } (H_i \rightarrow w_n) \in R_s \\ 0 & \text{otherwise} \end{cases}$$

2. For $j = n-1, \ldots, 1$ and for $i = 1, \ldots, |V_N| \quad , H_i \neq \sigma$

$$\begin{aligned} M_u[i,j] = \max \{ \ & \Pr(H_i \rightarrow w_j \ldots w_n) \ , \\ & \max_{k>j \ ,l} \{ \Pr(H_i \rightarrow w_j \ldots w_k H_l) M_u[l, k+1] \} \ \ \} \end{aligned}$$

3. For $i : H_i = \sigma$

$$M_u[i,1] = \max_j \{ \Pr(\sigma \rightarrow H_j) M_u[j,1] \}$$

This algorithm corresponds to the computation of an array similar to the one in figure 2, but where probabilities refer to maximum values of single derivations instead of total probability of derivation when ambiguous grammars are considered.

Based on the similarity between this algorithm and the one developed in section 3.1 it is straightforward to conclude that it runs in $O(|V_N|n)$ time.

3.3 Prefix Probability – Pr $(H \stackrel{*}{\Rightarrow} w_i \dots w_{i+n}\Sigma^*)$

The probability of all derivation trees having H as root and generating arbitrary length strings with the prefix $w_i \dots w_{i+n}$ – Pr $(H \stackrel{*}{\Rightarrow} w_i \dots w_{i+n}\Sigma^*))$ – can be expressed as follows:

$$\Pr(\sigma \stackrel{*}{\Rightarrow} w_i \dots w_{i+n}\Sigma^*) = \sum_G \Pr(\sigma \to G)\Pr(G \stackrel{*}{\Rightarrow} w_i \dots w_{i+n}\Sigma^*) \qquad (8)$$

$$\begin{aligned}
&\Pr(H \stackrel{*}{\Rightarrow} w_i \dots w_{i+n}\Sigma^*) \quad , \quad H \neq \sigma \\
&= \Pr(H \to w_i \dots w_{i+n}) + \\
&\quad + \sum_G \sum_{k=1}^{n_{min(H,G)}} \Pr(H \to w_i \dots w_{i+k-1}G)\Pr(G \stackrel{*}{\Rightarrow} w_{i+k} \dots w_{i+n}\Sigma^*) + \\
&\quad + \sum_G \Pr(H \to w_i \dots w_{i+n}G) + \\
&\quad + \sum_G \sum_{k \in \mathcal{N}^+} \Pr(H \to w_i \dots w_{i+n}v_1 \dots v_k G) \\
&= \Pr(H \to w_i \dots w_{i+n}) + \\
&\quad + \sum_G \sum_{k=1}^{n_{min(H,G)}} \Pr(H \to w_i \dots w_{i+k-1}G)\Pr(G \stackrel{*}{\Rightarrow} w_{i+k} \dots w_{i+n}\Sigma^*) + \\
&\quad + \sum_{G, k \in \mathcal{N}_0^+} \Pr(H \to w_i \dots w_{i+n}v_1 \dots v_k G) \qquad (9)
\end{aligned}$$

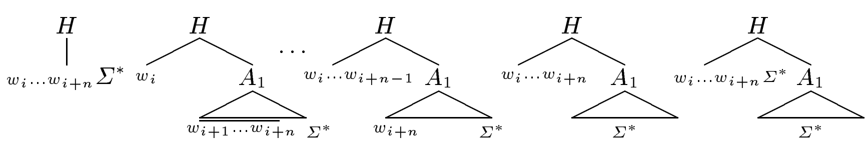


Fig. 3. Terms of the expression 9.

The previous expression suggest the following iterative algorithm:

1. For all nonterminal symbols $H \in V_N - \{\sigma\}$, determine

$$\Pr(H \to w_{i+n}) + \sum_{k \in \mathcal{N}^+} \Pr(H \to w_{i+n}v_1 \dots v_k)$$

2. For $k = 1, \dots, n$ and for all nonterminal symbols $H \in V_N - \{\sigma\}$, compute

$$\Pr(H \stackrel{*}{\Rightarrow} w_{i+n-k} \dots w_{i+n}\Sigma^*)$$

$$
\begin{aligned}
&= \Pr(H \to w_{i+n-k} \ldots w_{i+n}) + \\
&+ \sum_{G} \sum_{j=1}^{k_{min(H,G)}} \Pr(H \to w_{i+n-k} \ldots w_{i+n-k+j-1} G) \times \\
&\qquad \times \Pr(G \stackrel{*}{\Rightarrow} w_{i+n-k+j} \ldots w_{i+n} \Sigma^*) + \\
&+ \sum_{G} \sum_{j \in \mathcal{N}_0^+} \Pr(H \to w_{i+n-k} \ldots w_{i+n} v_1 \ldots v_j G)
\end{aligned} \tag{10}
$$

3.

$$
\Pr(\sigma \stackrel{*}{\Rightarrow} w_i \ldots w_{i+n} \Sigma^*) = \sum_{G} \Pr(\sigma \to G) \Pr(G \stackrel{*}{\Rightarrow} w_i \ldots w_{i+n} \Sigma^*) \tag{11}
$$

This algorithm, like the one of section 3.1, corresponds to filling in a parsing matrix similar to the one in figure 2.

The similarity with the algorithm in section 3.1 leads to the conclusion that this algorithm has $O(|V_N| n)$ time complexity.

3.4 Island Probability – Pr $(H \stackrel{*}{\Rightarrow} \Sigma^* w_i \ldots w_{i+n} \Sigma^*)$

The island probability consists of Pr $(H \stackrel{*}{\Rightarrow} \Sigma^* w_i \ldots w_{i+n} \Sigma^*)$, the probability of all derivation trees with root in H that generate arbitrary length sequences containing the subsequence $w_i \ldots w_{i+n}$.

Let

$$
\begin{aligned}
\mathrm{P}_R(H \to G) &\stackrel{\text{def}}{=} \sum_{\gamma} \Pr(H \to \gamma G) \quad , \ \gamma \in \Sigma^* \\
&= \text{probability of rewriting } H \text{ by sequences with} \\
&\quad\ \text{the nonterminal symbol } G \text{ as suffix}
\end{aligned} \tag{12}
$$

One can write

$$
\Pr(\sigma \stackrel{*}{\Rightarrow} \Sigma^* w_i \ldots w_{i+n} \Sigma^*) = \sum_{G} \Pr(\sigma \to G) \Pr(G \stackrel{*}{\Rightarrow} \Sigma^* w_i \ldots w_{i+n} \Sigma^*) \tag{13}
$$

$$
\begin{aligned}
&\Pr(H \stackrel{*}{\Rightarrow} \Sigma^* w_i \ldots w_{i+n} \Sigma^*) \quad , \quad H \neq \sigma \\
&= \sum_{G} \mathrm{P}_R(H \to G) \Pr(G \stackrel{*}{\Rightarrow} \Sigma^* w_i \ldots w_{i+n} \Sigma^*) + \\
&\quad + \sum_{G} \sum_{j=1}^{n_{min(H,G)}} \mathrm{P}_R(H \to w_i \ldots w_{i+j-1} G) \Pr(G \stackrel{*}{\Rightarrow} w_{i+j} \ldots w_{i+n} \Sigma^*) + \\
&\quad + \sum_{G} \mathrm{P}_R(H \to w_i \ldots w_{i+n} G) \underbrace{\Pr(G \stackrel{*}{\Rightarrow} \Sigma^*)}_{=1} +
\end{aligned}
$$

$$+\sum_G \sum_{j,k} \Pr(H \to v_1 \dots v_k w_i \dots w_{i+n} z_1 \dots z_j G) \underbrace{\Pr(G \overset{*}{\Rightarrow} \Sigma^*)}_{=1} +$$

$$+\sum_G \sum_{j,k} \Pr(H \to v_1 \dots v_k w_i \dots w_{i+n} z_1 \dots z_j) \tag{14}$$

For strings sufficiently long ($n >^{max}_{G} \{C_T(\gamma) : G \to \gamma\}$) the last three terms do not exist, therefore we will ignore them from now on.

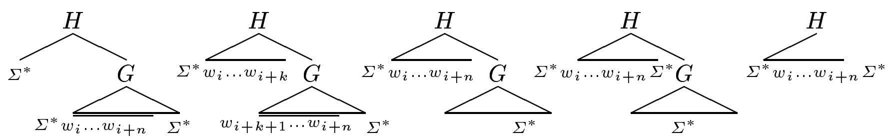


Fig. 4. Terms of the expression 14.

$$\begin{aligned}
&\Pr(H \overset{*}{\Rightarrow} \Sigma^* w_i \dots w_{i+n} \Sigma^*) \quad , \quad H \neq \sigma \\
&= \sum_G \mathrm{P}_R(H \to G) \Pr(G \overset{*}{\Rightarrow} \Sigma^* w_i \dots w_{i+n} \Sigma^*) + \\
&\quad + \sum_G \sum_{j=1}^{n_{min(H,G)}} \mathrm{P}_R(H \to w_i \dots w_{i+j-1} G) \Pr(G \overset{*}{\Rightarrow} w_{i+j} \dots w_{i+n} \Sigma^*)
\end{aligned} \tag{15}$$

Recursively applying the above expression and after some manipulation (details can be found in [6]) one obtains:

$$\begin{aligned}
&\Pr(H \overset{*}{\Rightarrow} \Sigma^* w_i \dots w_{i+n} \Sigma^*) \quad , \quad H \neq \sigma \\
&= \sum_A Q_R(H \Rightarrow A) \sum_G \sum_{j \in \mathcal{N}^+} \mathrm{P}_R(A \to w_i \dots w_{i+j-1} G) \Pr(G \overset{*}{\Rightarrow} w_{i+j} \dots w_{i+n} \Sigma^*) \\
&\quad + \sum_G \sum_{j \in \mathcal{N}^+} \mathrm{P}_R(H \to w_i \dots w_{i+j-1} G) \Pr(G \overset{*}{\Rightarrow} w_{i+j} \dots w_{i+n} \Sigma^*)
\end{aligned} \tag{16}$$

with

$$\begin{aligned}
&Q_R(H \Rightarrow G) = \Pr(H \overset{*}{\Rightarrow} \Sigma^* G) \\
&= \mathrm{P}_R(H \to G) + \\
&\quad + \sum_A \mathrm{P}_R(H \to A) \mathrm{P}_R(A \to G) + \sum_{A_1 A_2} \mathrm{P}_R(H \to A_1) \mathrm{P}_R(A_1 \to A_2) \times \\
&\quad \times \mathrm{P}_R(A_2 \to G) + \dots
\end{aligned} \tag{17}$$

$\mathrm{Q}_R(H \Rightarrow G)$ obeys the equation

$$\mathrm{Q}_R(H \Rightarrow G) = \sum_A \mathrm{P}_R(H \to A) \mathrm{Q}_R(A \Rightarrow G) + \mathrm{P}_R(H \to G) \tag{18}$$

Defining the matrices

$$\mathcal{P}_R[H,G] = \mathrm{P}_R(H \to G) \tag{19}$$
$$\mathcal{Q}_R[H,G] = \mathrm{Q}_R(H \Rightarrow G) \tag{20}$$

$\mathcal{Q}$ is given by [6]:

$$\mathcal{Q}_R = \mathcal{P}_R[I - \mathcal{P}_R]^{-1} \tag{21}$$

The algorithm can thus be described as:

1. Off-line computation of $\mathcal{Q}_R = \mathcal{P}_R[I - \mathcal{P}_R]^{-1}$.
2. On-line computation of $\Pr(G \stackrel{*}{\Rightarrow} w_n \Sigma^*)$, $\Pr(G \stackrel{*}{\Rightarrow} w_{n-1} w_n \Sigma^*)$, ... , $\Pr(G \stackrel{*}{\Rightarrow} w_2 \dots w_n \Sigma^*)$ for all nonterminal symbols $G \in V_N - \{\sigma\}$ using the algorithm in section 3.3.
3. For all nonterminal symbols $H \in V_N - \{\sigma\}$ compute

$$\begin{aligned}
&\Pr(H \stackrel{*}{\Rightarrow} \Sigma^* w_i \dots w_{i+n} \Sigma^*)\\
&= \sum_A Q_R(H \Rightarrow A) \sum_G \sum_{j \in \mathcal{N}+} \mathrm{P}_R(A \to w_i \dots w_{i+j-1} G) \times\\
&\qquad\qquad \times \Pr(G \stackrel{*}{\Rightarrow} w_{i+j} \dots w_{i+n} \Sigma^*) +\\
&\quad + \sum_G \sum_{j \in \mathcal{N}+} \mathrm{P}_R(H \to w_i \dots w_{i+j-1} G) \Pr(G \stackrel{*}{\Rightarrow} w_{i+j} \dots w_{i+n} \Sigma^*)
\end{aligned}$$

4.

$$\Pr(\sigma \stackrel{*}{\Rightarrow} \Sigma^* w_i \dots w_{i+n} \Sigma^*) = \sum_G \Pr(\sigma \to G) \Pr(G \stackrel{*}{\Rightarrow} \Sigma^* w_i \dots w_{i+n} \Sigma^*)$$

The required on-line computations have the following time complexity:

$$\begin{aligned}
&O\left(\underbrace{(|V_N|-1)(n-1)\alpha}_{\text{step 2}} + \underbrace{(|V_N|-1)|V_N|\beta}_{\text{step 3}} + \underbrace{\delta}_{\text{step 4}}\right) =\\
&= max(O\left(|V_N| n\right), O\left(|V_N|^2\right)
\end{aligned}$$

3.5 Suffix Probability – $\Pr(H \stackrel{*}{\Rightarrow} \Sigma^* w_i \dots w_{i+n})$

Let Pr $(H \stackrel{*}{\Rightarrow} \Sigma^* w_i \dots w_{i+n})$ be the probability of all derivation trees with root in H generating arbitrary length strings having $w_i \dots w_{i+n}$ as a suffix. This probability can be derived as follows:

$$\Pr(\sigma \stackrel{*}{\Rightarrow} \Sigma^* w_i \dots w_{i+n}) = \sum_G \Pr(\sigma \to G) \Pr(G \stackrel{*}{\Rightarrow} \Sigma^* w_i \dots w_{i+n}) \tag{22}$$

$$
\begin{aligned}
&\Pr(H \stackrel{*}{\Rightarrow} \Sigma^* w_i \ldots w_{i+n}) \quad , \quad H \neq \sigma \\
&= \sum_G \mathrm{P}_R(H \to G)\Pr(G \stackrel{*}{\Rightarrow} \Sigma^* w_i \ldots w_{i+n}) + \\
&\quad + \sum_G \sum_{j=1}^{n_{min(H,G)}} \mathrm{P}_R(H \to w_i \ldots w_{i+j-1} G)\Pr(G \stackrel{*}{\Rightarrow} w_{i+j} \ldots w_{i+n}) + \\
&\quad + \sum_G \mathrm{P}_R(H \to w_i \ldots w_{i+n})
\end{aligned} \tag{23}
$$

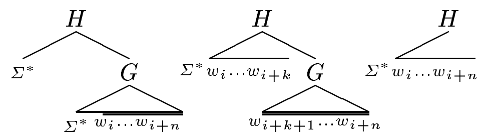


Fig. 5. Terms of the expression 23.

For strings sufficiently long ($n > \stackrel{max}{G} \{C_T(\gamma) : G \to \gamma\}$) the third term does not exist, so it will be ignored henceforth. Recursively applying the resulting expression to the second part of the first term one obtains:

$$
\begin{aligned}
&\Pr(H \stackrel{*}{\Rightarrow} \Sigma^* w_i \ldots w_{i+n}) \quad , \quad H \neq \sigma \\
&= \sum_{A_1} \sum_{j \in \mathcal{N}^+} \mathrm{P}_R(H \to w_i \ldots w_{i+j-1} A_1)\Pr(A_1 \stackrel{*}{\Rightarrow} w_{i+j} \ldots w_{i+n}) + \\
&\quad + \sum_{A_1, A_2} \sum_{j \in \mathcal{N}^+} \mathrm{P}_R(H \to A_1)\mathrm{P}_R(A_1 \to w_i \ldots w_{i+j-1} A_2). \\
&\quad .\Pr(A_2 \Rightarrow w_{i+j} \ldots w_{i+n}) + \\
&\quad + \ldots + \\
&\quad + \sum_{A_1, \ldots, A_k} \mathrm{P}_R(H \to A_1)\mathrm{P}_R(A_1 \to A_2) \ldots \\
&\quad \ldots \mathrm{P}_R(A_{k-1} \to w_i \ldots w_{i+j-1} A_k)\Pr(A_k \stackrel{*}{\Rightarrow} w_{i+j} \ldots w_{i+n}) + \ldots
\end{aligned} \tag{24}
$$

$$
\begin{aligned}
&= \sum_A Q_R(H \Rightarrow A) \sum_G \sum_{j \in \mathcal{N}^+} \mathrm{P}_R(A \to w_i \ldots w_{i+j-1} G)\Pr(G \stackrel{*}{\Rightarrow} w_{i+j} \ldots w_{i+n}) + \\
&\quad + \sum_G \sum_{j \in \mathcal{N}^+} \mathrm{P}_R(H \to w_i \ldots w_{i+j-1} G)\Pr(G \stackrel{*}{\Rightarrow} w_{i+j} \ldots w_{i+n})
\end{aligned} \tag{25}
$$

The algorithm is then:

1. Off-line computation of $\mathcal{Q}_R = \mathcal{P}_R[I - \mathcal{P}_R]^{-1}$.

2. On-line computation of $\Pr(G \stackrel{*}{\Rightarrow} w_n)$, $\Pr(G \stackrel{*}{\Rightarrow} w_{n-1}w_n)$, ... , $\Pr(G \stackrel{*}{\Rightarrow} w_2 \dots w_n)$ for all non-terminal symbols $G \in V_N - \{\sigma\}$ using the algorithm in section 3.1.
3. For all non-terminal symbols $H \in V_N - \{\sigma\}$ compute

$$\begin{aligned}
&\Pr(H \stackrel{*}{\Rightarrow} \Sigma^* w_i \dots w_{i+n}) \\
&= \sum_A Q_R(H \Rightarrow A) \sum_G \sum_{j \in \mathcal{N}^+} \mathrm{P}_R(A \to w_i \dots w_{i+j-1} G) \Pr(G \stackrel{*}{\Rightarrow} w_{i+j} \dots w_{i+n}) \\
&\quad + \sum_G \sum_{j \in \mathcal{N}^+} \mathrm{P}_R(H \to w_i \dots w_{i+j-1} G) \Pr(G \stackrel{*}{\Rightarrow} w_{i+j} \dots w_{i+n})
\end{aligned}$$

4.

$$\Pr(\sigma \stackrel{*}{\Rightarrow} \Sigma^* w_i \dots w_{i+n}) = \sum_G \Pr(\sigma \to G) \Pr(G \stackrel{*}{\Rightarrow} \Sigma^* w_i \dots w_{i+n})$$

On-line computations have the time complexity:

$$O\left(\underbrace{(|V_N|-1)(n-1)\alpha}_{\text{step 2}} + \underbrace{(|V_N|-1)\beta}_{\text{step 3}} + \underbrace{\delta}_{\text{step 4}}\right) = O\left(|V_N| n\right)$$

4 Conclusions

This paper described several algorithms for the computation of substring probabilities according to grammars written in the form

$$\sigma \to F_i \quad , \quad F_i \to \alpha \quad , \quad F_i \to \alpha F_j \quad , \quad \alpha \in \Sigma^* \quad , \quad \sigma, F_i, F_j \in V_N$$

Table 1 summarizes the probabilities considered here, and the order of complexity of the associated algorithms.

	Expression	Algorithm time Complexity
Fixed length	$\Pr(H \stackrel{*}{\Rightarrow} w_1 \dots w_n)$	$O(\lvert V_N \rvert n)$
strings	$\mathrm{P_m}(H \stackrel{*}{\Rightarrow} w_1 \dots w_n)$	$O(\lvert V_N \rvert n)$
Arbitrary	$\Pr(H \stackrel{*}{\Rightarrow} w_1 \dots w_n \Sigma^*)$	$O(\lvert V_N \rvert n)$
length	$\Pr(H \stackrel{*}{\Rightarrow} \Sigma^* w_1 \dots w_n \Sigma^*)$	$max(O(\lvert V_N \rvert n), O(\lvert V_N \rvert^2)$
strings	$\Pr(H \stackrel{*}{\Rightarrow} \Sigma^* w_1 \dots w_n)$	$O(\lvert V_N \rvert n)$

Table 1. Summary of proposed algorithms for the computation of sub-string probabilities.

More general algorithms for the computation of sub-string probabilities according to stochastic context-free grammars, written in Chomsky Normal Form, can be found in [1–3]. However, the later have $O(n^3)$ time complexity [2]. The herein proposed algorithms, exhibiting linear time complexity in string's length, represent a computationally appealing alternative to be used whenever the application at hand can adequately be modeled by the types of grammars described above. Examples of application of the algorithms described in this paper can be found in [5, 6, 8, 9].

References

1. F. Jelinek, J. D. Lafferty and R. L. Mercer. Basic Methods of Probabilistic Context Free Grammars. In *Speech Recognition and Understanding. Recent Advances*, pages 345–360. Springer-Verlag, 1992.
2. A. Corazza, R. De Mori, R. Gretter, and G. Satta. Computation of Probabilities for an Island-Driven Parser. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, No. 9, pages 936–949, 1991.
3. A. Corazza, R. De Mori, and G. Satta. Computation of Upper-Bounds for Stochastic Context-Free Languages. In proceedings *AAAI-92*, pages 344–349, 1992.
4. A. Corazza, R. De Mori, R. Gretter, and G. Satta. Some Recent results on Stochastic Language Modelling. In *Advances in Structural and Syntactic Pattern Recognition*, World-Scientific, pages 163–183, 1992.
5. A. L. N. Fred, A. C. Rosa, and J. M. N. Leitão. Predicting REM in sleep EEG using a structural approach. In E. S. Gelsema and L. N. Kanal, editors, *Pattern Recognition in Practice IV*, pages 107 – 117. Elsevier Science Publishers, 1994.
6. A. L. N. Fred, *Structural Pattern Recognition: Applications in Automatic Sleep Analysis,* PhD Thesis, Technical University of Lisbon, 1994.
7. K. S. Fu and T. L. Booth. Grammatical inference: Introduction and survey – part I and II. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-8:343–359, May 1986.
8. Ana L. N. Fred and T. Paiva. Sleep Dynamics and Sleep Disorders: a Syntactic Approach to Hypnogram Classification. Em *10th Nordic-Baltic Conference on Biomedical Engineering and 1st International Conference on Bioelectromagnetism*, pp 395–396, Tampere, Finlândia Junho, 1996.
9. A. L. N. Fred, J. S. Marques, P. M. Jorge, Hidden Markov Models vs Syntactic Modeling in Object Recognition, *Proc. Intl. Conference on Image Processing,*ICIP'97, 1997.