Introduction to multi-tape S-attributed grammars

Jérôme Waldispühl

LIX, École Polytechnique
91128 Palaiseau Cedex
waldispu@lix.polytechnique.fr

A complete and technical description of multi-tape S-attributed grammars would be too long to do. We give in this section a brief overview of this part of language's theory but the reader will can find more informations in [1] or [2].

1 S-attributed context-free grammars

We recall the basic definitions of context-free grammars and of their natural extension to S-attributed grammars. These definitions are adapted to the fact that we use this formalism to model biological properties of sequences.

Definition 1 (**Context-free grammar**)
A context-free grammar $G = (V_T, V_N, P, S)$ consists of a finite set of terminals V_T, a finite set of nonterminals V_N such that $V_T \cap V_N = \emptyset$, a finite set of productions (rewriting rules) P and a start symbol $S \in V_N$. Let $V = V_T \cup V_N$ denote the vocabulary for the grammar. Each production in P has the form $A \to \alpha$, where $A \in V_N$ and $\alpha \in V^\star$. A is the left-hand side of the production and α its right-hand side.

The transitive closure of the derivation relation $\to$ is denoted $\stackrel{\star}{\to}$. A derivation tree is the planar representation of a sequence of derivations $A \to \alpha$ such that $A \in V_N$ and $\alpha \in V^\star$. The set of strings in $V_T^\star$ derived from S is called the language generated by G and it is denoted by $L(G)$. The empty string is denoted by ε.
In order to keep the number of derivation trees finite for a given word $\omega \in L(G)$, we assume that the grammar is non-circular, which means that no nonterminal A may verify $A \stackrel{+}{\to} A$. We also assume that the grammar is epsilon-free (i.e. it has no rules of the form $A \to \varepsilon$). An ambiguous grammar is a grammar for which there exists a string of symbols having at least two different derivation trees. For example, the grammar whose derivation trees describe t-RNA secondary structures has to be ambiguous because a given RNA has potentially several different secondary structures. Parsing is the process of finding a derivation tree for a string in $L(G)$, which is called the parse tree of the sequence.
S-attributed context-free grammars, which are a proper subset of attributed-grammars introduced by Knuth in his seminal paper [1], are an extension of context-free grammar allowing the assignment of a value (called attribute) to every vertex of a derivation tree.

Definition 2 (**S-attributed grammar**)
An S-attributed grammar, denoted by $G = (V_T, V_N, P, S\ , \mathcal{A}, S_\mathcal{A}, F_P)$, is an extension of the context-free grammar $G = (V_T, V_N, P, S)$; an attribute $x \in \mathcal{A}$ is attached to each symbol $X \in V$, and a string of attributes $\lambda \in \mathcal{A}^\star$ to each string $\alpha \in V^\star$. $S_\mathcal{A}$ is a function from V_T to $\mathcal{A}$ assigning attributes to

$A \to \alpha$ *in* P.

The attribute of a string $\alpha \in V^\star$, denoted by λ_α, is the concatenation of the attributes of the symbols in α. When a function $f_{A\to\alpha}$ is applied to the attribute λ_α derived from A it returns the attribute $x = f_{A\to\alpha}$ of $\mathcal{A}$. Thus, the functions of F_P determine the bottom-up computation of the attributes of nonterminals A in derivations $\alpha A\beta \stackrel{\star}{\to} u$, where u must belong to $L(G)$ so that the attribute of A be computable.

Example 1 We consider the following ambiguous context-free grammar for arithmetical expressions.

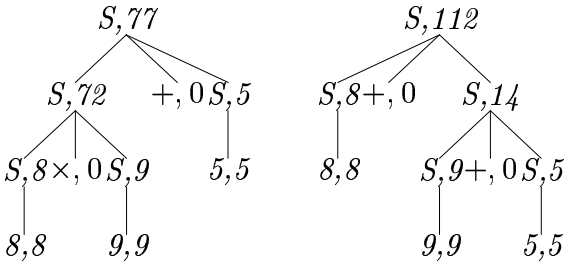
$$\begin{aligned} S &\to S + S \\ S &\to S \times S \\ S &\to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9. \end{aligned}$$

We extend it into an S-attributed grammar such that each attribute stores the value of the arithmetical expression corresponding to the derivation subtree rooted at this node.

$$\begin{aligned} \mathcal{A} &= \mathbb{N} \\ S_{\mathcal{A}}(0) &= 0 \\ &\vdots \\ S_{\mathcal{A}}(9) &= 9 \\ S_{\mathcal{A}}(+,\times) &= 0 \end{aligned}$$

$$F_P = \left\{ \begin{array}{l} f_{S\to S+S}(x0z) = x + z \\ f_{S\to S\times S}(x0z) = x \times z \\ f_{S\to a\in V_T}(x) = x \end{array} \right\}$$

Thus, the sequence $8 \times 9 + 5$ can be parsed in different ways corresponding to different derivation trees, hence producing different values



As previously noted, grammars used for biological sequence analysis are ambiguous because they aim at describing the space of all possible configurations. Attributes are designed to give an evaluation of the quality for each of them which gives a choice criterion. For example, if the attribute of a vertex is an energy or a probability, the criterion may be the selection of the derivation tree with the lowest energy or the highest probability at the root. (But attributes are not restricted to simple real values in general even if this paper deals mainly with this kind). The selection of the best possible value for an attribute is done by means of a function called optimization constraint.

Definition 3 (**Optimization constraint**)
Let G be an S-attributed grammar and let $a_1 \cdots a_n$ be a string of $L(G)$. Let x be the attribute of a derivation $A \to \alpha_1 \to \cdots \to \alpha_k \to a_{i+1} \cdots a_j$ and let x' be the attribute of another derivation $A \to \alpha'_1 \to \cdots \to \alpha'_k \to a_{i+1} \cdots a_j$. The optimisation constraint $\mathcal{C}_{\mathcal{A}}$ associated with the nonterminal A takes as an input the attributes x and x' and returns another attribute x''. We note $x'' = \mathcal{C}_{\mathcal{A}}(x, x')$.

In practice, $\mathcal{C}_{\mathcal{A}}$ will be a comparison function between x and x' which returns the optimal attribute (the lowest energy, if attributes are free energy). Finally, we denote by λ_ω the optimal attribute of a string $\omega \in L(G)$.

We extend now this formalism in order to use a "m-tape" alphabet.

Definition 4 (m**-tape index**)
A m-tape index $\vec{i}$ is a vector of $\mathbb{Z}^m$. the notation $\vec{i}^{(k)}$ will denote the k-th tape. We shall say that a m-tape index $\vec{i}$ is inferior to a m-tape index $\vec{j}$ if this order stands on every tape, and we shall denote this relation by $\vec{i} \leq \vec{j}$.

The m-tape index $\vec{1}$ has the value 1 on all tapes. The sum of two m-tapes indexes $\vec{i}_1$ and $\vec{i}_2$ is a m-tape index $\vec{j}$ defined by $\vec{j}^{(k)} = \vec{i}_1^{(k)} + \vec{i}_2^{(k)}$. Thus, $\vec{i} \leq \vec{j}$ also means $\vec{i} - \vec{j} \leq 0$.

Definition 5 (m**-tape input string**)
Given m alphabet $\Sigma^{(i)}$ $(1 \leq i \leq m)$, a m-tape input string is a vector of m strings $(a^{(1)}_{\vec{1}^{(1)}} \cdots a^{(1)}_{\vec{n}^{(1)}}, \cdots, a^{(m)}_{\vec{m}^{(m)}} \cdots a^{(m)}_{\vec{n}^{(m)}})$, where each string $a^{(i)}_{\vec{1}^{(i)}} \cdots a^{(i)}_{\vec{n}^{(i)}}$ belongs to $\left(\Sigma^{(i)}\right)^\star$. We shall denote the set of such defined input strings by $\langle \Sigma^\star \rangle = \bigotimes_{i=1\cdots n} \Sigma^{(i)^\star}$. As a shorthand, any m-tape input string $\left(a^{(1)}_{\vec{1}^{(1)}} \cdots a^{(1)}_{\vec{n}^{(1)}}, \cdots, a^{(m)}_{\vec{m}^{(m)}} \cdots a^{(m)}_{\vec{n}^{(m)}}\right)$ may be denoted by $a_{\vec{1}} \cdots a_{\vec{n}}$. Substrings of $a_{\vec{1}} \cdots a_{\vec{n}}$ will be denoted by $a_{\vec{i}} \cdots a_{\vec{j}} = \left(a^{(1)}_{\vec{i}^{(1)}} \cdots a^{(1)}_{\vec{j}^{(1)}}, \cdots, a^{(m)}_{\vec{i}^{(m)}} \cdots a^{(m)}_{\vec{j}^{(m)}}\right)$ with the usual conventions that $\vec{1} \leq \vec{i}$, $\vec{j} \leq \vec{n}$ and, if $\vec{i}^{(k)} > \vec{j}^{(k)}$, $a^{(k)}_{\vec{i}^{(k)}} \cdots a^{(k)}_{\vec{j}^{(k)}} = \varepsilon$.

Example 2 *(abba,dcd) is a 2-tape input string on $\Sigma^{(1)} = a, b$ and $\Sigma^{(2)} = c, d$. We shall also write this 2-tape input string as $\begin{smallmatrix} a & b & b & a \\ d & c & d \end{smallmatrix}$, which a somewhat more natural notation in the context of alignments. This 2-tape input string $a_{\binom{1}{1}\rightarrow} \cdots a_{\binom{4}{3}\rightarrow} = \begin{smallmatrix} a & b & b & a \\ d & c & d \end{smallmatrix}$ has a 2-tape input string $a_{\binom{2}{1}\rightarrow} \cdots a_{\binom{3}{2}\rightarrow} = \begin{smallmatrix} b & b \\ d & c \end{smallmatrix}$.*

Definition 6 (m**-tape alphabet**)
A m-tape alphabet Σ is a product of m alphabets $\Sigma^{(i)}$ augmented with the empty string: $\Sigma = \bigotimes_{i=1\cdots m}(\Sigma^{(i)} \cup \{\varepsilon\})$.

Definition 7 (m**-tape alignment**)
An element $a_1 \cdots a_l$ of the free monoid $\Sigma^\star$, generated by formal concatenation of m-tape elements of Σ, is called a m-tape alignment of length l. The empty alignment of $\Sigma^\star$ is denoted by ε.

Definition 8 (ε**-deletion**)
Given any m-tape alignment $a_1 \cdots a_l$, we get a m-tape input string $a_{\vec{1}} \cdots a_{\vec{n}}$ by concatenation of symbols of the projection of $a_1 \cdots a_l$ on every tape.

$$\begin{array}{rcl} \Sigma^\star & \rightarrow & \langle \Sigma^\star \rangle \\ a_1 \cdots a_l & \rightarrow & a_{\vec{1}} \cdots a_{\vec{l}} = \langle a_1 \cdots a_l \rangle \end{array}$$

Example 3 *The 2-tape input string $\begin{smallmatrix} a & b & b & a \\ d & c & d \end{smallmatrix}$ may be defined as an ε-deletion of the alignments $\left\langle \begin{bmatrix} \varepsilon \\ b \end{bmatrix} \begin{bmatrix} a \\ \varepsilon \end{bmatrix} \begin{bmatrix} b \\ \varepsilon \end{bmatrix} \begin{bmatrix} b \\ c \end{bmatrix} \begin{bmatrix} a \\ d \end{bmatrix} \right\rangle$ or $\left\langle \begin{bmatrix} a \\ \varepsilon \end{bmatrix} \begin{bmatrix} b \\ d \end{bmatrix} \begin{bmatrix} \varepsilon \\ c \end{bmatrix} \begin{bmatrix} b \\ \varepsilon \end{bmatrix} \begin{bmatrix} a \\ d \end{bmatrix} \right\rangle$.*

Once the m-tape alphabets and strings have been defined, we naturally extend a context-free grammar into a m-tape context-free grammar.

Definition 9 (m**-tape context-free grammar**) *A m-tape context-free grammar $G = (V_T, V_N, P, S)$ is classical context-free grammar such that V_T* **is a subset of a** m**-tape alphabet**.

Notice that in m-tape case, $L(G)$ is not the set of derivable strings but the set of ε-deleted such strings.

with a's:

$$S \rightarrow \begin{bmatrix} (\\ (\end{bmatrix} S \begin{bmatrix}) \\ ) \end{bmatrix} \mid \begin{bmatrix} a \\ a \end{bmatrix} \mid \begin{bmatrix} a \\ \varepsilon \end{bmatrix} \mid \begin{bmatrix} \varepsilon \\ a \end{bmatrix} \mid SS$$

In this MTSAG, the structure defined by parentheses must be the same on both tapes, but substrings of a may be aligned with gaps (denoted with ε).

Then, as in mono-tape case, a m-tape context-free grammar can be extended into a m-tape S-attributed grammar by addition of attributes and functions to compute the non-terminal attributes.

Definition 10 (**m-tape S-attributed grammar**) *A m-tape S-attributed grammaris denoted by $G = (V_T, V_N, P, S, \mathcal{A}, S_{\mathcal{A}}, F_P)$. It is an extension of a m-tape context-free grammar $G = (V_T, V_N, P, S)$, where an attribute $x \in \mathcal{A}$ is attached to each symbol $X \in V$ and a string of attributes $\lambda \in \mathcal{A}^\star$ to each string $\alpha \in V^\star$. $S_{\mathcal{A}}$ is a function from V_T to $\mathcal{A}$ assigning attributes to terminals. F_P is a set of functions from $\mathcal{A}^\star$ to $\star A$. A function $f_{A\rightarrow\alpha}$ is in F_P iff $A \rightarrow \alpha$ is in P.*

References

[1] D.E. Knuth. Semantic of context-free languages. *Mathematical Systems theory*, 2:127–145, 1968. Correction: Mathematical Systems theory 5:95-96, 1971.

[2] F. Lefebvre. A grammar-based unification of several alignment and folding algorithms. In AAAI press, editor, *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, pages 143–154, 1996.