1. From Finite Automata to Right Linear and Left Linear Grammars

The algorithm presented here is not in the book because it uses the procedures for the elimination of ε-productions and unit productions which in the book are presented later, when describing simplifications of the context-free grammars.

ALGORITHM 1.1. *From Finite Automata to Right-Linear or Left-Linear Grammars.*

Input: a finite automaton which accepts the language L.
Output: a right-linear or a left-linear grammar which generates the language L.

If the finite automaton has no final states, then the right-linear or the left-linear grammar has an empty set of productions. If the finite automaton has at least one final state, then we perform the following steps.
Step (1). (1.1) Add a new initial state S with an ε-arc to the old initial state, which will no longer be the initial state. (1.2) Add a new final state F with ε-arcs from the old final state(s) which will no longer be final state(s).

Step (2). For every arc $A \xrightarrow{a} B$, with $a \in \Sigma \cup \{\varepsilon\}$, add the following production.

For right-linear grammars:	*For left-linear grammars*:
$A \to a\,B$	$B \to A\,a$
(the future of A is	(the past of B is
a followed by the future of B)	the past of A followed by a)

Step (3). Fix the beginning (the axiom) and the end (the ε-production).
(3.1) The only symbol which occurs only on the *left* of a production, is the axiom. (3.2) The only symbol which occurs only on the *right* of a production, has an ε-production.

For right-linear grammars:	*For left-linear grammars*:
choose S as the axiom	choose F as the axiom
add $F \to \varepsilon$	add $S \to \varepsilon$
(the final state F has an empty future)	(the initial state S has an empty past)

Step (4). Eliminate by unfolding the ε-production and the unit productions.

Note 1. If the given automaton has no final states, then the language accepted by that automaton is empty, and both the left-linear and right-linear grammars we want to construct, have an empty set of productions.

Note 2. After the introduction of the new initial state and the new final state, never the initial state is also a final state. Moreover, no arc goes to the initial state and no arc departs from the final state.

Note 3. In the above algorithm we add exactly one production for every transition $A \xrightarrow{a} B$. The above algorithm deals in a symmetric way with the cases of right-linear and the left-linear grammars.

Note 4. The choice of the axiom and the addition of the ε-production have the effect of making *useful* every symbol of the grammar produced at the end of Step (2).

We add one ε-production only. That ε-production forces an empty future of the final state F (for right-linear grammars), and an empty past of the initial state S (for left-linear grammars).

Step (3) is related to Step (1) in the sense that: (i) we choose the axiom because of the new initial state S, and (ii) we add an ε-production because of the new final state F.

Note 5. At the end of Step (3) the grammar has one ε-production and one or more unit productions. □