7.7. Finite Automata to/from S-extended Regular Grammars

ALGORITHM 7.7.1.

Procedure: *from Deterministic or Nondeterministic Finite Automata to S-extended Right Linear or Left Linear Grammars.*

Input: a deterministic or nondeterministic finite automaton which accepts the language $L \subseteq \Sigma^*$.
Output: an S-extended 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). Add a new initial state q_0 with an ε-arc to the old initial state, which will no longer be the initial state. Add a new final state q_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 production:
$A \rightarrow a\,B$ *for the right linear grammar.* | $B \rightarrow A\,a$ *for the left linear grammar.*
Step (3). The symbol which occurs *only* on the *left* of a production, is the axiom, and the symbol which occurs *only* on the *right* of a production, has an ε-production, that is,

for the right linear grammar:	*for the left linear grammar*:
take q_0 as the axiom	take q_f as the axiom
add $q_f \rightarrow \varepsilon$	add $q_0 \rightarrow \varepsilon$

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. The form of the productions $A \rightarrow a\,B$ and $B \rightarrow A\,a$ for the right linear grammar and the left linear grammar, respectively, can be recalled by thinking at the boxed parts of the following diagrams of the arc $A \xrightarrow{a} B$:

for the right linear grammar:	*for the left linear grammar*:
$A\ \boxed{\xrightarrow{a} B}$	$\boxed{A \xrightarrow{a}}\ B$
$A \rightarrow \underline{a\,B}$	$B \rightarrow \underline{A\,a}$

Note that for the right linear grammar and the left linear grammar, the two symbols occurring on the right hand side of the production ($\underline{a\,B}$ and $\underline{A\,a}$, respectively), are *in the same order* in which they occur in the arc $A \xrightarrow{a} B$.

Note 3. We add exactly one production for every arc $A \xrightarrow{a} B$. With reference to what we have said on page 44, we have that:
(i) for the right linear grammar *every state encodes its future until a final state* and thus, $A \rightarrow a\,B$ tells us that the future of A is a followed by the future of B, and
(ii) for the left linear grammar *every state encodes its past from the initial state* and thus, $B \rightarrow A\,a$ tells us that the past of B is the past of A followed by a.

Note 4. At Step (3) the choice of the axiom and the addition of the ε-production make every symbol of the derived grammar, to be a *useful* symbol.

At Step (3) we add one ε-production only, and that ε-production forces an empty future of the final state q_f (for the right linear grammar), and an empty past of the initial state q_0 (for the left linear grammar).

At the end of Step (3) the grammar may have one or more unit productions.

ALGORITHM 7.7.2.

Procedure: *from S-extended Right Linear or Left Linear Grammars to Deterministic or Nondeterministic Finite Automata.*

Input: an S-extended right linear or a left linear grammar which generates the language $L \subseteq \Sigma^*$.
Output: a deterministic or nondeterministic finite automaton which accepts the language L.
Step (1). Add the new axiom symbol q_0 with production $q_0 \rightarrow S$, where S is the old axiom and eliminate the ε-production and the unit productions (and by doing so, the new axiom q_0 will not occur on the right hand side of the derived grammar). Add a new nonterminal symbol q_f and replace every production of the form $A \rightarrow a$ by the production:

$A \rightarrow a\ q_f$ *for the right linear grammar*	$A \rightarrow q_f\, a$ *for the left linear grammar*

Let $\langle V_T, V_N, P, q_0\rangle$ be the derived grammar.
Step (2). Let V_N be the set of states of the finite automaton.

For the right linear grammar:	*For the left linear grammar*:
for every production $A \rightarrow a\,B$	for every production $A \rightarrow B\,a$
add the arc $A \xrightarrow{a} B$	add the arc $B \xrightarrow{a} A$
State q_0 is initial. State q_f is final.	State q_f is initial. State q_0 is final.

Step (3). If $q_0 \rightarrow \varepsilon$ occurs in P, then the initial state is also a final state.
Step (4). If the derived automaton is nondeterministic, we can transform it into an equivalent deterministic automaton by applying the Powerset Construction (see Algorithm 2.3.11 on page 39).

Note 1. The replacement of every production of the form $A \rightarrow a$ preserves the generated language if we add the production $q_f \rightarrow \varepsilon$ when deriving the new grammar. We do *not* do so, but in the derived automaton we consider q_f to be the final (or initial) state for right (or left, respectively) linear grammars.