Example: Regular Grammar$_{JP}$

Enter a regular grammar that generates the following language L over the alphabet Σ = {a, b}:

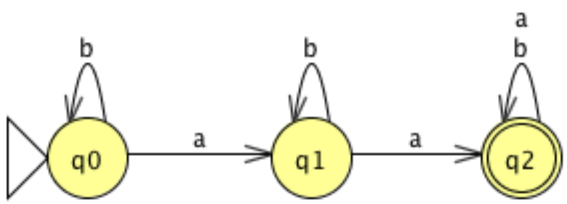
L = { *w* | *w* contains at least two **a** symbols }

Sample Solution (see `RG_XaXaX.jff`)

Consider that the only constraint is two appearances of the symbol **a** and that any number of symbols may appear before and between them.

Recall that a regular grammar may be specified in right-linear format. Thus all productions must have at most one variable in the right-hand side and that variable must be consistently to the right of any terminals.

Note that one possible DFA for this language is the following (see DFA_XaXaX.jff):



This suggests that the start symbol, in correspondence to state q0, should be able to produce any number of **b** symbols followed by a string that contains the required number of **a** symbols.

$$S \rightarrow bS$$

Further, once an **a** is produced, the remaining string must simply contain at least one **a** symbol, corresponding to state q1.

$$S \rightarrow aA$$

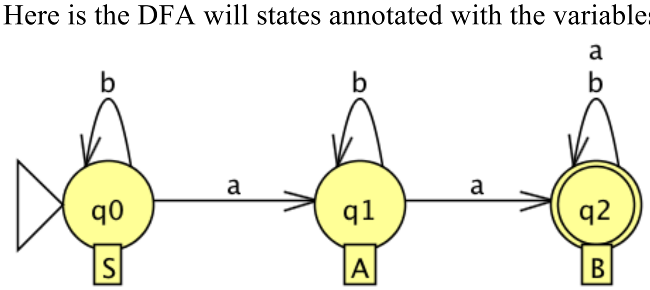
Similarly, the variable A should be able to produce any number of b symbols followed by a string that contains at least one a symbol.

$$A \rightarrow bA$$
$$A \rightarrow aB$$

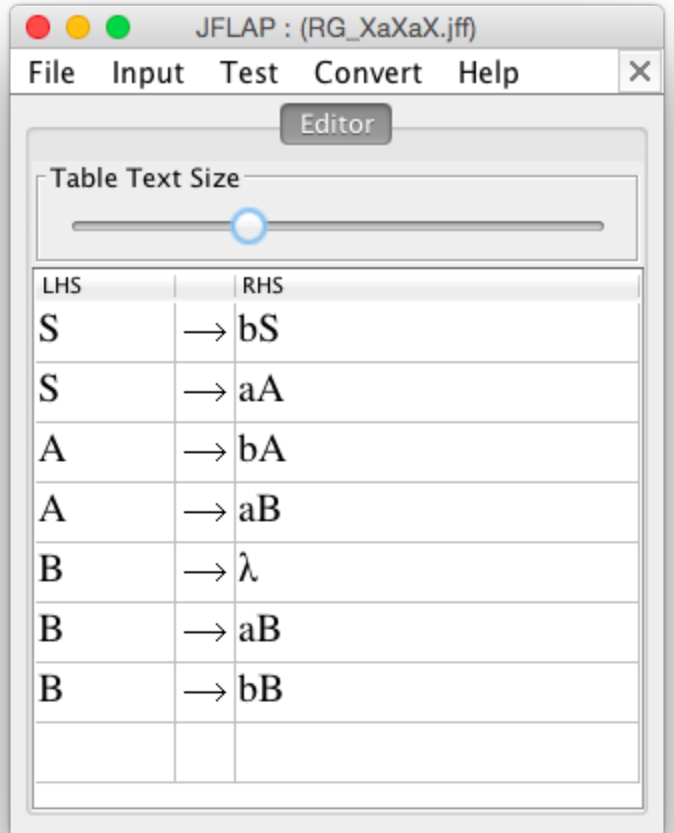
The remaining rules combine to produce all strings over {a, b}*.

$$B \rightarrow \varepsilon \mid aB \mid bB$$

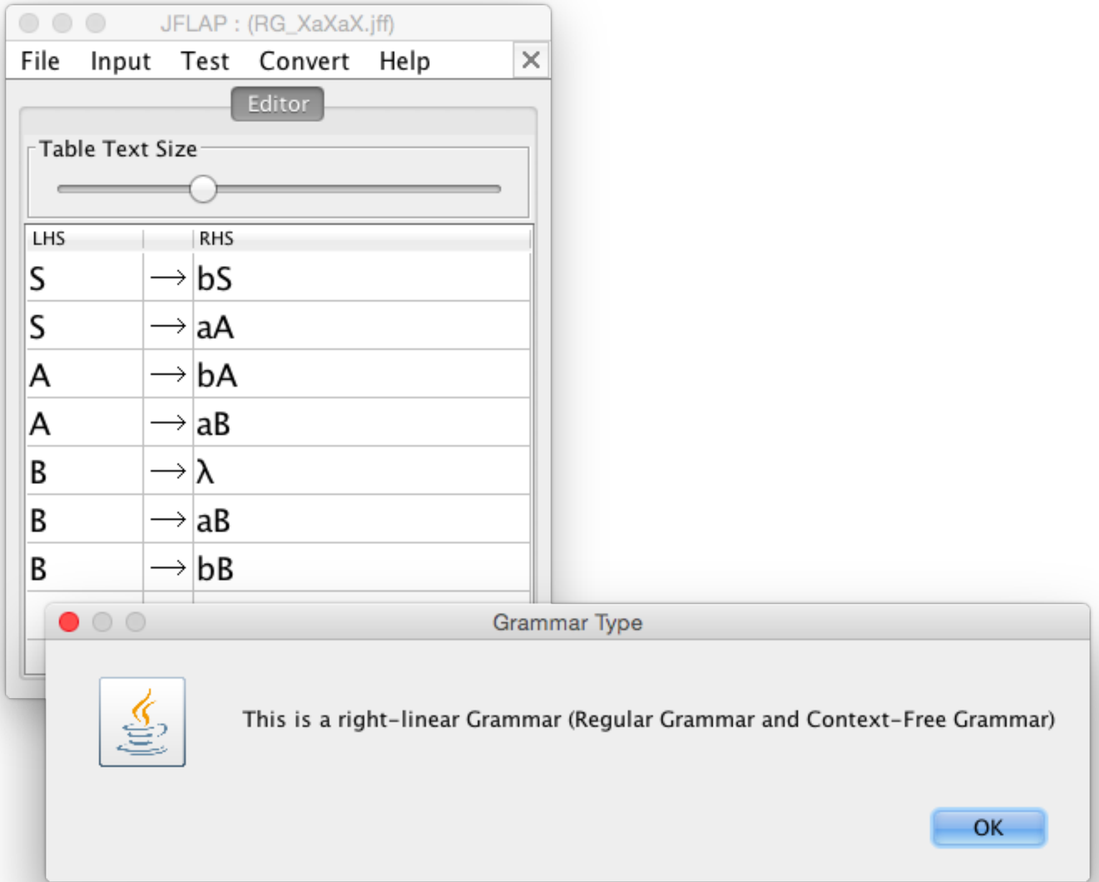
Here is the DFA will states annotated with the variables of this grammar.



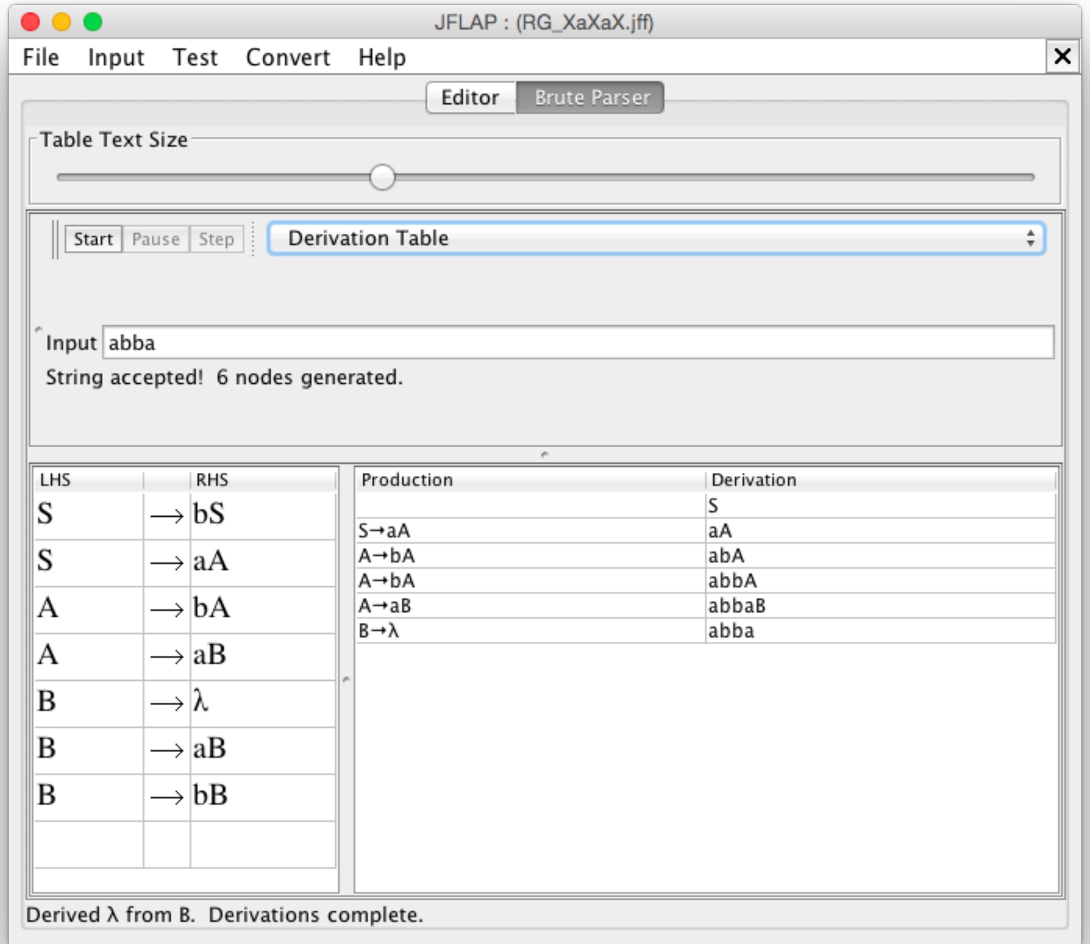
1. Enter this grammar into JFLAP.



2. Check the type of grammar using *Test* > *Test for Grammar Type.*



3. Verify known strings using the *Brute Force Parse.*


JFLAP : (RG_XaXaX.jff)
File
Input
Test
Convert
Help
Editor
Brute Parser
Table Text Size
Start
Pause
Step
Derivation Table
Input
abba
String accepted! 6 nodes generated.
LHS
RHS
S → bS
S → aA
A → bA
A → aB
B → λ
B → aB
B → bB
Production
Derivation
S
S→aA
aA
A→bA
abA
A→bA
abbA
A→aB
abbaB
B→λ
abba
Derived λ from B. Derivations complete.