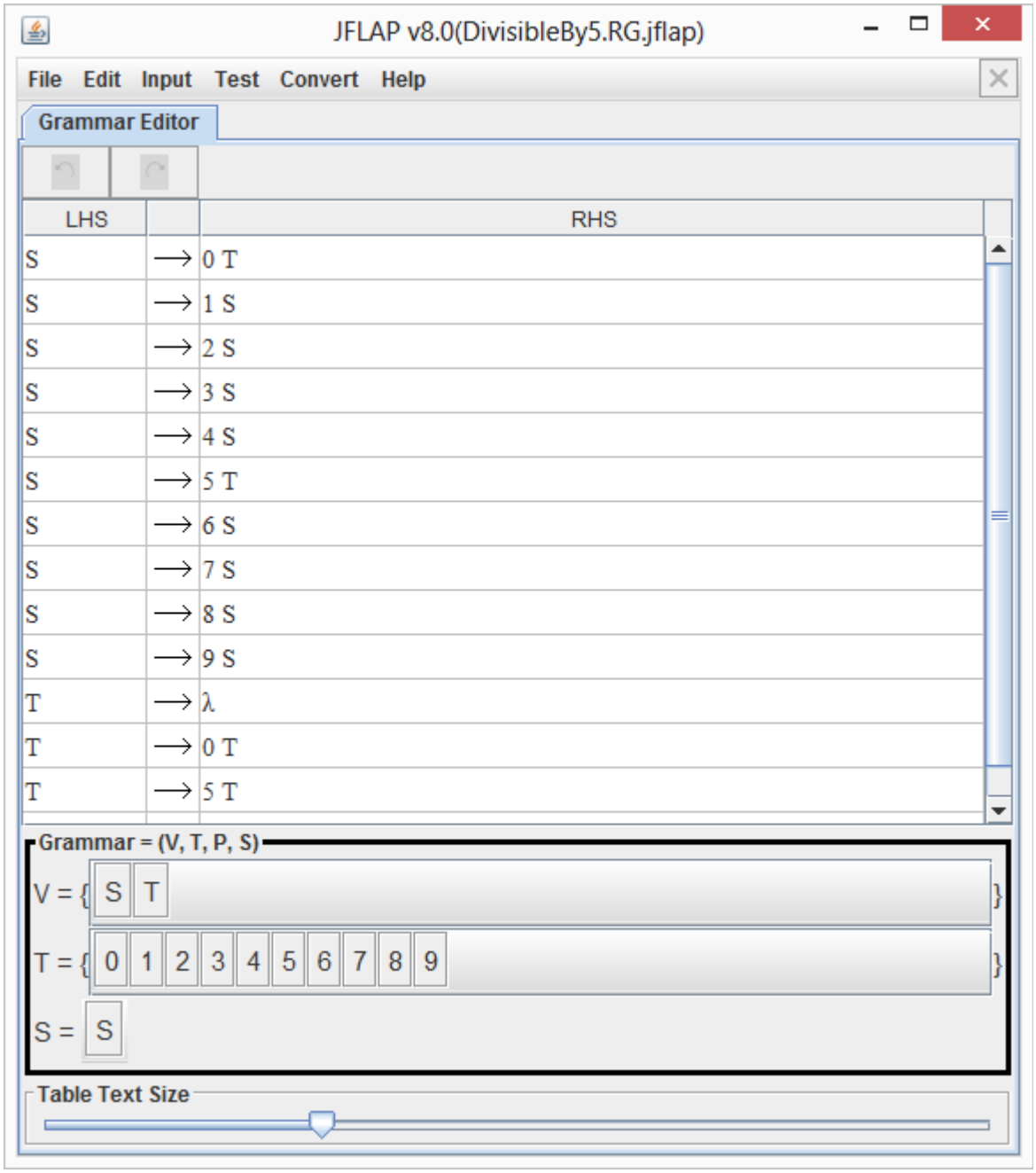
Additional Example to Practice with Converting a Regular Grammar to a DFA
Martha Kosa

In the module, you learned the process for converting a right-regular grammar to a DFA. Now you can practice some more.

Try It!

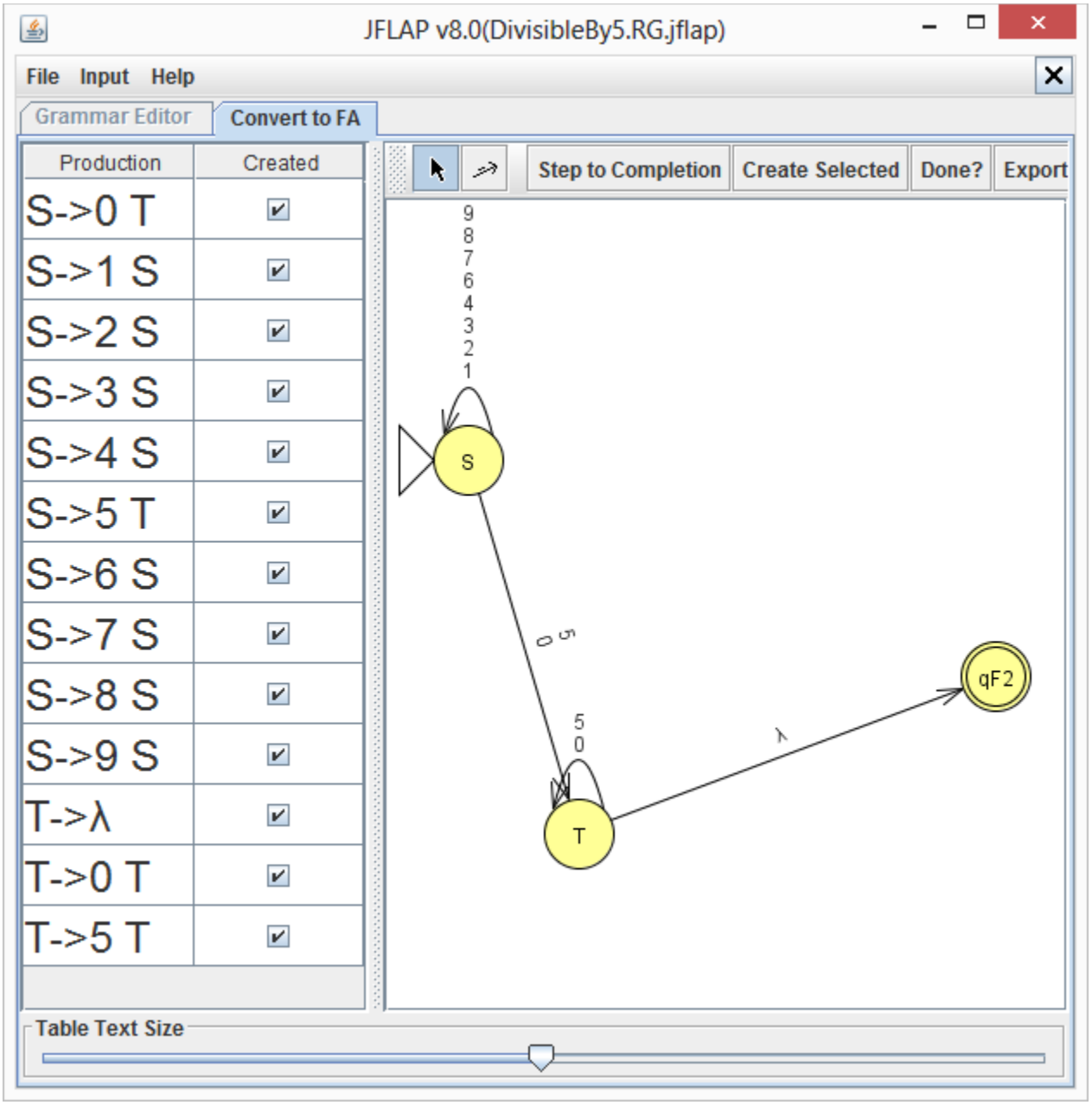
Start JFLAP and select ***File > Open*** to load the file **DivisibleBy5.RG.jflap**. This grammar generates all strings of decimal digits such that the represented integer is divisible by 5. For an explanation of how this grammar was developed, please see the module on regular grammars and its associated examples. The file should open with this window:



The technique given in the module was demonstrated for rules of the form $A \rightarrow w$, where $w \in T^*$. We have one rule with λ on the right-hand side, namely, $T \rightarrow \lambda$. How can we handle this? One way is to make the state corresponding to grammar variable T be a final state.

Create the DFA corresponding to the grammar by using the JFLAP Automaton Editor and manually creating the states and associated transitions. Save your automaton with a descriptive file name.

With the DivisibleBy5.RG.jflap file that you have open, select ***Convert > Convert Regular Grammar to FSA***. A new JFLAP screen appears, with the grammar in the left-hand pane. Click **Step to Completion** to see a screen similar to the following:



Questions to Think About:

1. Is the generated automaton a DFA? Why or why not? If you are having trouble answering this question, click **Export** and select ***Test > Highlight Nondeterminism*** in the new window that appears.
2. If it is not a DFA, how can you easily modify it to be a DFA?
3. Export the corresponding DFA, and compare its equivalence to the DFA that you developed by selecting ***Test > Compare Equivalence*** and selecting your DFA. The DFA should be equivalent to yours.
4. Describe any differences between the corresponding DFA and your DFA.