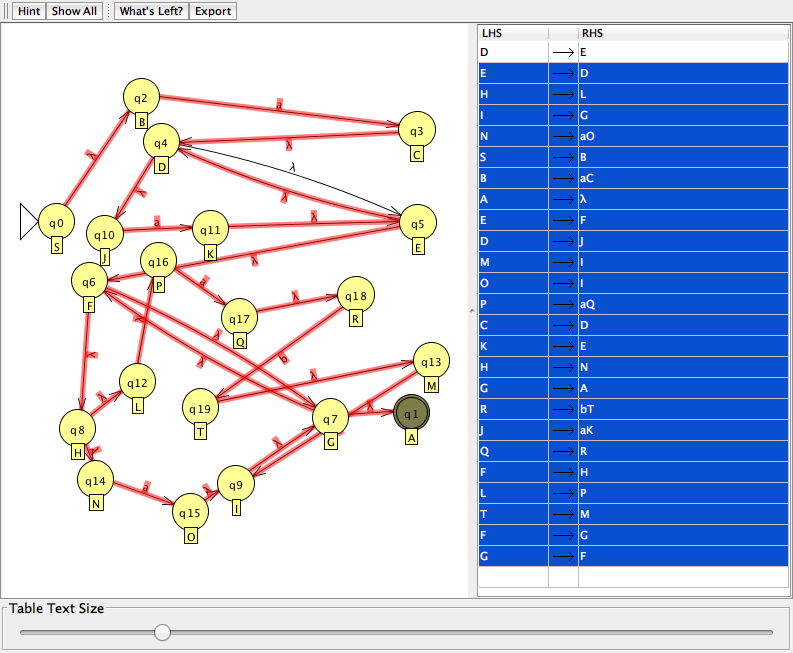
*“Use JFLAP to convert the regular expression to an NFA and then convert the NFA to a regular grammar… Compare that with the results of your answer you obtained by hand…”*

Results from converting the regular expression ***aa\*(ab+a)\**** 🡪 NFA then to 🡪 grammar in JFLAP.



Results from 3.3 #3 I did by hand.

G (V, T, S, P)

P:

S 🡪 aA

A 🡪 aA | abB | aB | λ

B 🡪 abB | aB | λ

T:

{a,ab}

V:

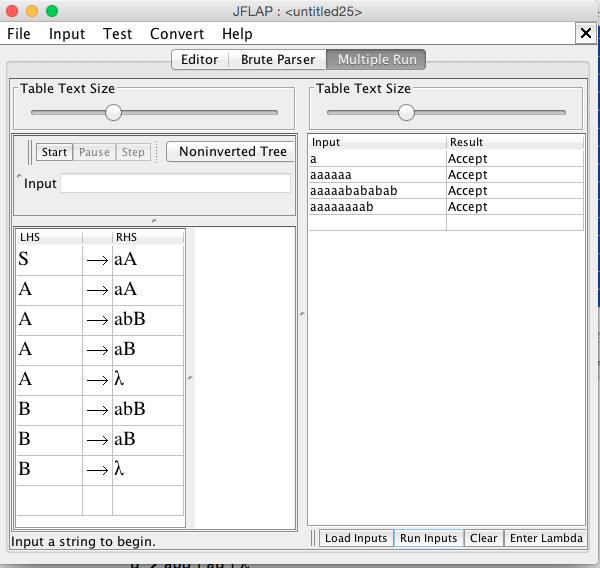
{S, A, B}

S:

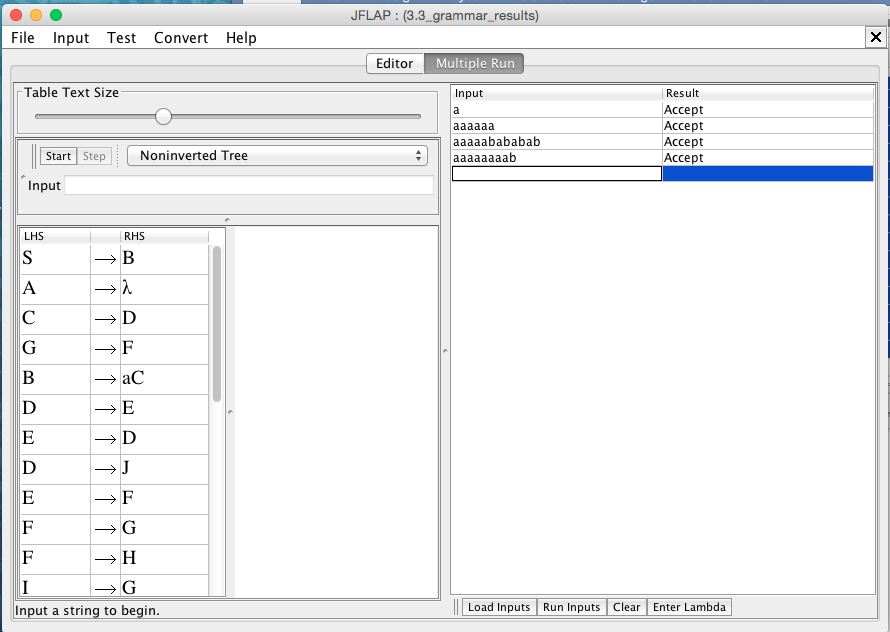
S

Some interesting things I found were that my grammar and the grammar generated by JFLAP under several input strings were equally generated. As show in the following images.

My Grammar



JFLAP’s Grammar



*Note: Both accepted the same critical strings.*

∴ Comparatively, I found that my grammar was much smaller and more concise then the one generated by JFLAP, but they are two ways of expressing the same thing. This shows me that the algorithms are very helpful as a fall back when you cannot determine the solution on your own, but they are not always best if you can “*eye ball”* the solution and get something much more familiar and easy to deal with.