Given

r1: *ab\*aa*

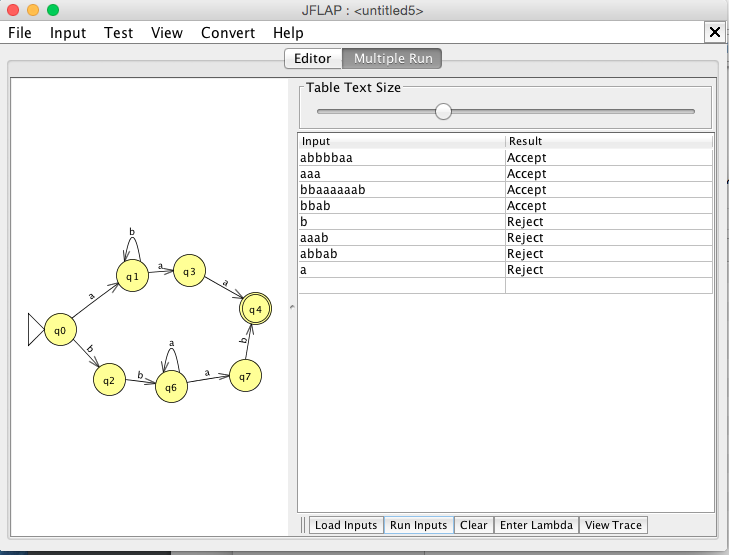
r2: *bb\*ab*

Find a simpler NFA for the language L (r1 + r2)

Using the *“eyeball”* method to *“simplify”* make part a simpler.

Considering we have no intersection between our languages , we cannot merge the two language representations in a machine M(r3), where r3 is a new representation of r1 and r2 with some new shared states, because they are uniquely independent from one another in their representation as a language. This merging of M(r1) and M(r2) would damage the union of our languages, which was our goal.

However, by removing λ-transitions we can simplify our machine to still accept the same language L (r1 + r2).



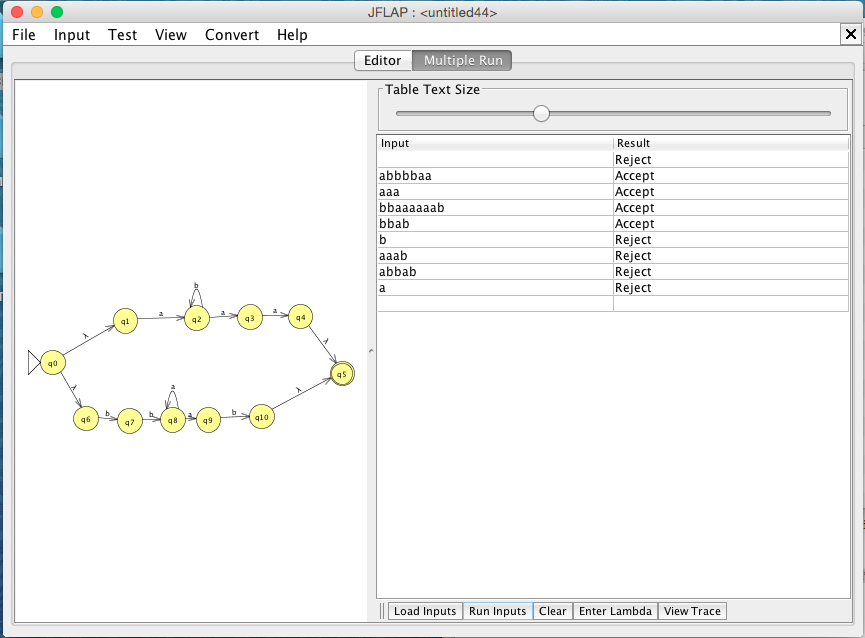
∴ A simplified NFA representing , or L(r1 + r2).

NOTE: From the construction algorithm in the book we can also create an eyeball nfa that is pretty similar to the one constructed above.

Let our upper half of the image be M(r1) which accepts ab\*aa.

Let our lower half of the image be M(r2) which accepts bba\*ab.

Then we can separate machines M(r1) and M(r2) with λ-transitions by *Theorem 3.1* as expressed in *Figure 3.3* and unite them to a common final state with a λ-transition.



The Machine above represents the canonical approach from *Theorem 3.1* as suggested in *Figure 3.1, 3.2, and 3.3* for the language *L(ab\*aa + bba\*ab).*