KEY

Practice - Regular Expressions

[redacted] [redacted] Always show all work.

1. Alphabet = {a,b}
Find the shortest string that is not in the language represented by the regular expression a*(ab)*b* .

λ, a, b are all in the language.
aa, bb, ab are also in the language.
But ba cannot be generated.

2. Alphabet = {a,b}
For the two regular expressions: r1 = a* | b* r2 = ab* | ba* | b*a | (a*b)*
(a). Find a string that is generated by r2 but not by r1.
(b). Find a string that is generated by r1 and by r2.

(not the empty string)

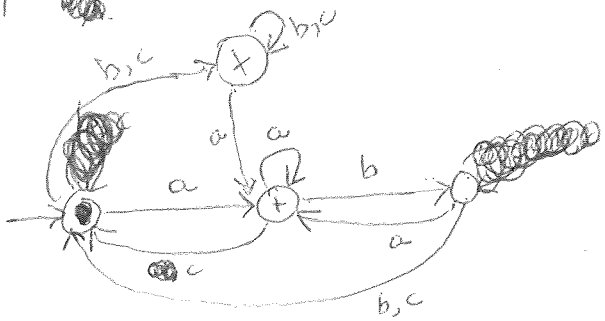
(a). r1 only generates just a's or b's, so anything with a mixture such as ab or ba.
(b). Only strings generated by r2 consisting of only a's or b's are a, b, or strings having only b's from (a*b)*.

3. Alphabet = {a,b,c}
Write a regular expression for the language of all strings that do not end in "ab" .

at least length one,

(a|b|c)*(a|bb|c|cb) | b (Hint: Consider what you can end with.)

aa
→ ab
ac
ba
bb
bc
ca
cb
cc



4. Alphabet = {a,b,c}
Write a regular expression for the language of all strings that do not contain the substring "ab" .

(b|c|a*c)*a* which simplifies to (b|a*c)*a*