CSI 3104 / 3504 , Winter 2011
Solution of Assignment 1

15) page 20
i) no w could be concatenation of words of S without being in S
ii) Since w Є T then w Є T* = S* = S**

17) page 20
(i) S= {aa, ab, ba, bb}. Since S has all possible words of length 2, S* is the set of all words of even length bigger or equal than 2.
(ii) S= {aaa, aab, aba, abb, baa, bab, bba, bbb, Λ}
(iii) Since any non empty word has either an odd length or an even length. And a non empty word with even length could be decomposed into a concatenation of two words of odd length, then S* is the set of all non empty words.

19) page 20
S= {aa, ba, aba, abaab}.
ababaaba is a word in S* however the algorithm will delete aba then aba an we are left with ab which in not in S. However w can be decomposed as a concatenation of aba, ba, aba.

20) page 20
Since T is closed and S ⊂ T, any factors in S concatenated together two at a time will be a word in T. Likewise, concatenating factors in S any number of times produces a word in T. That is any word in S* is also in T. However we are given that T ≠ S* so T contains some words that are not in S*. We can conclude that S* is a proper subset of T, in other words S* is smaller than T, and in symbols
S* ⊂ T.

19) page 30

. (i) Rule 1: Λ is in EVENSTRING
Rule 2: If w is in EVENSTRING., then so are waa, wab, wba, wbb.
Another equivalent definition
Rule 1: Λ, aa, ab, ba, bb are in EVENSTRING.
Rule 2: If u and w are in EVENSTRING., then so is uw.

(iii) Rule 1: aa is in AA.
Rule 2: If w is in AA, then so are aw, bw, wa and wb.

10) page 49
(**b*ab*ab*ab*)***

Exercises # 16 (ii); 17(iv) ; 18 , page 50
16. (ii) a*b is the language of all words that have exactly one *b* which is the last letter. Its Kleene closure, (a*b)*, is the language of all words that do not end in *a.* Concatenating the a* permits words to end in *a. So* (a*b)*a* gives all words over (a, b } . Symmetrically, ba* is all the words that have

one *b*, the first letter. Its closure is all words that begin with *b*. Concatenating the initial a* gives all words over {a, b}, so the regular expressions define the same language.

17) (iv)

(iv) Λ + a(a+b)* + (a+b)*aa(a+b)* and **((b*a)*ab*)***

a(ba+a)*b describes words that begin with a and end with *b*. In the body of any word Ys are always followed by an a (which prevents clumps of b's), however the clumps of a's are unrestricted. The associated language contains all strings where each b is surrounded by at least one a on either side and that ends in *b*.aa*b(aa*b)* describes the same language where each *b is* preceded by at least one a. Both expressions define the language of all words over { a, b }.

18)

(i) (a+b)*a(Λ+bbbb)

words that finishes with **a** or **abbbb**

(ii) **(a(a+bb)*)***

Λ word and words starting with a and where the b's appear in clumps of even length.

(iii) **(a(aa)*b(bb)*)***

Λ word and words composed of sequences of a's and sequences of b's of odd length, starting with an odd sequence of a's and finishing with an odd sequence of b's.

(iv) **(b(bb)*)*(a(aa)*b(bb)*)***

Λ word and words composed of sequences of a's and sequences of b's of odd length and finishing with an odd sequence of b's.

(v) **(b(bb)*)*(a(aa)*b(bb)*)*(a(aa)*)***

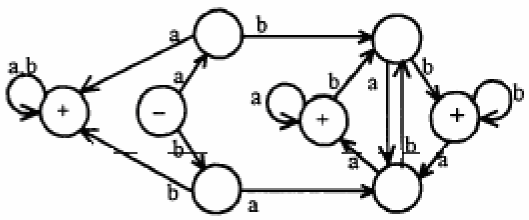
Λ word and words composed of sequences of a's and sequences of b's of odd length.

(vi) **((a+b)a)***

All words of even length that (excepts) a`s occupy all even positions.

7) page 71:

7.



Exercise # 17 page 73:

(i) All non empty words with odd length and where every odd position in the word (position 1, 3, 5, …) contains an **a.**

(ii) All non empty words where at least one even position in the word (position 2, 4, 6, …) contains an **a.**

(iii) All non empty words where all even positions in the word (position 2, 4, 6, …) contains an **a.**

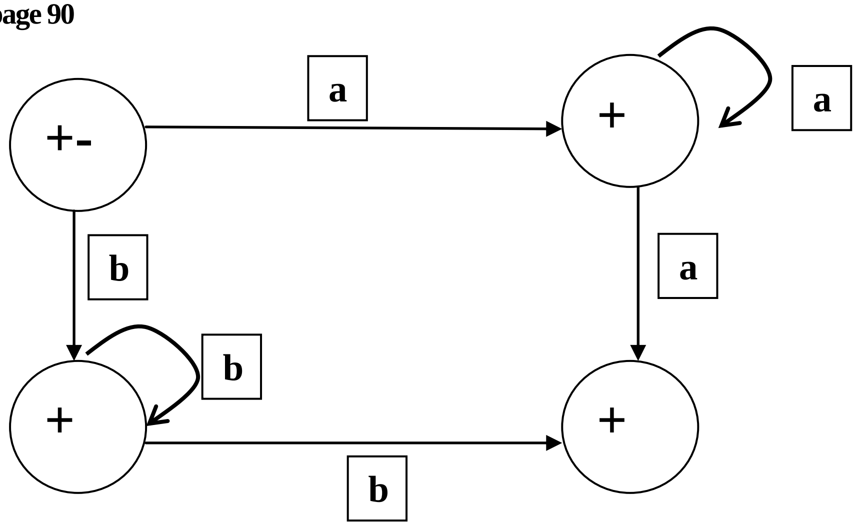
(iv) (a(a+b)*a (a+b)(b(a+b))*a ((a+b)(b(a+b))*a)* (a+b)a((a+b)a)*

Exercise # 18 page 74

18. We start in state 1 and remain there until we encounter an *a*. State 2 = we have just read an *a*. Scan any *a*'s and return to state l on reading *c*. State 3 = we have read a *b* following an a. Reading an a puts us back to state 2 and reading a b sets us back to state 1. However state 4 = we have just found a substring *abe,* and if the whole sequence was read the string is accepted.

Now states 4, 5 and 6 exactly mirror states l, 2 and 3. Returning to state l indicates that we just found another occurrence of the substring *abc.* Being in one of the first three states means that we have read an even number of *abc* substrings (if any) and are in the midst of finding another one. Ending in an accepting state, 4, 5 or 6, means that we have read an odd number of *abc*'s.

15) page 90



Deleting the transition **b** from the intial state will lead to **a***, deleting the transition **a** from the intial state will lead to **b***.

Deleting the other transitions **a** and **b** will lead to **a* + b***