cosc-120
midterm, quizzes

A language recognizing machine is a FSM. Instead of halting states, it has one or more "accepting" states. If the machine runs out of input and is in an accepting state, the input string is "recognized". There is a "grammar" associated with each such machine that expresses the language in a different way. It has rules for generating the strings. Such a grammar is a "regular expression". You use regular expressions all the time to search. Here is a regular expression for a language:

(ab)+(cd)+[a|c]*

"(ab)+" means 1 or more "ab"s, "(cd)+" means 1 or more "cd"s.
"(ab)+(cd)+" mean one or more "ab"s followed by one or more "cd"s.
"[a|c]" means "a" or "c", and "[a|c]*" means choosing "a" or "c" repeatedly, including the empty string "".

Here are sample strings from the language:

abcd
abcdaa
ababcdcdcdcdccaa

Show the state-transition diagram for a FSM that accepts this language, assume the input symbol set is {a, b, c, d}.
Here are preliminary questions to help you get started:
(Q) How many state transitions from each state?
(Q) What does the start state mean?
(Q) Show a FSM accepting the language (ab)+ (assume the same symbol set).
(Q) Show a FSM accepting the language (cd)+.
(Q) Show a FSM accepting the language (ab)+(cd)+.
(Q) Show a FSM accepting the language [a|c]*.

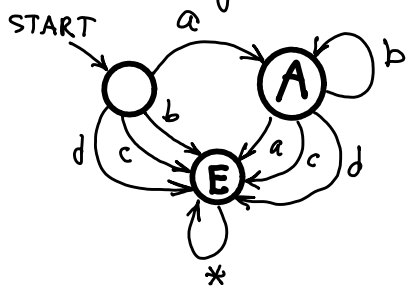
4, one for each symbol in {a,b,c,d}

"No symbols read"

See next page

Show accepting states like this (A).

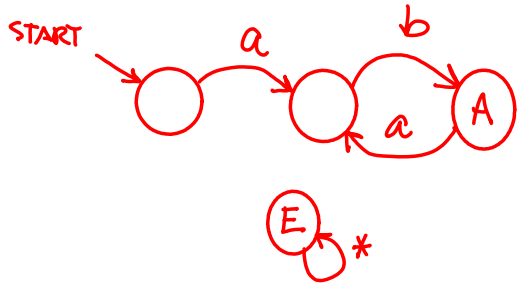
Here's a machine accepting (a)(b*):

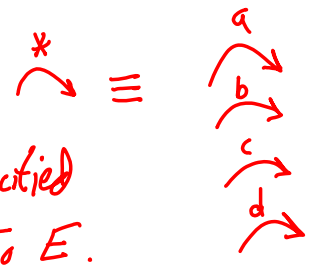


E is a non-accepting error state.

Check that it accepts "a" and "abbb" but not "aba".

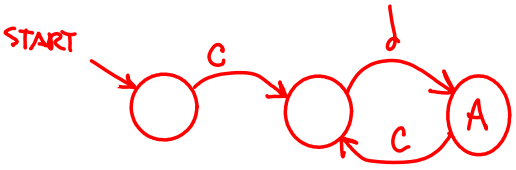
$(ab)+$ ⇒ Language = { ab, abab, ababab, ... } A = "accepting state"



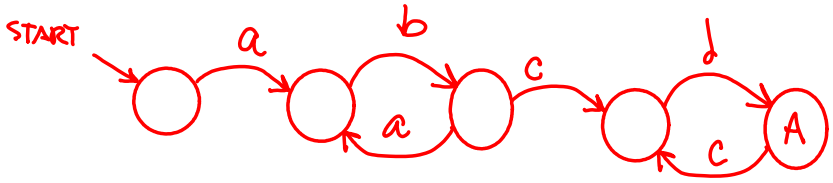


Note: all unspecified transitions go To E.

$(cd)+$ ⇒ Language = { cd, cdcd, cdcdcd, ... }



$(ab)+(cd)+$ ⇒ { abcd, ababcd, abcdcd, ababab cd, ababcdcd, ... }



$[a|c]^*$ ⇒ { "", a, c, aa, ac, ca, cc, aaa, aac, ... }

$(ab)+(cd)+[a|c]^*$