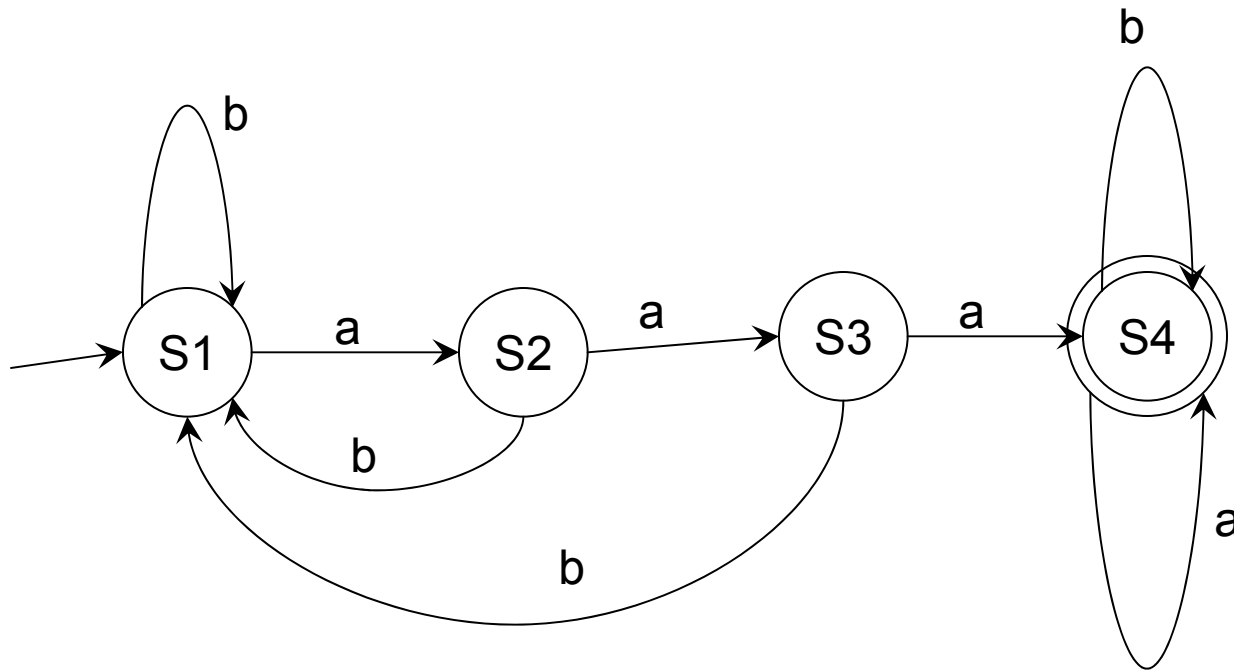


DFA Construction Examples

DFA Construction

- 1. Strings of a's and b's such that there are at least 3 consecutive a's

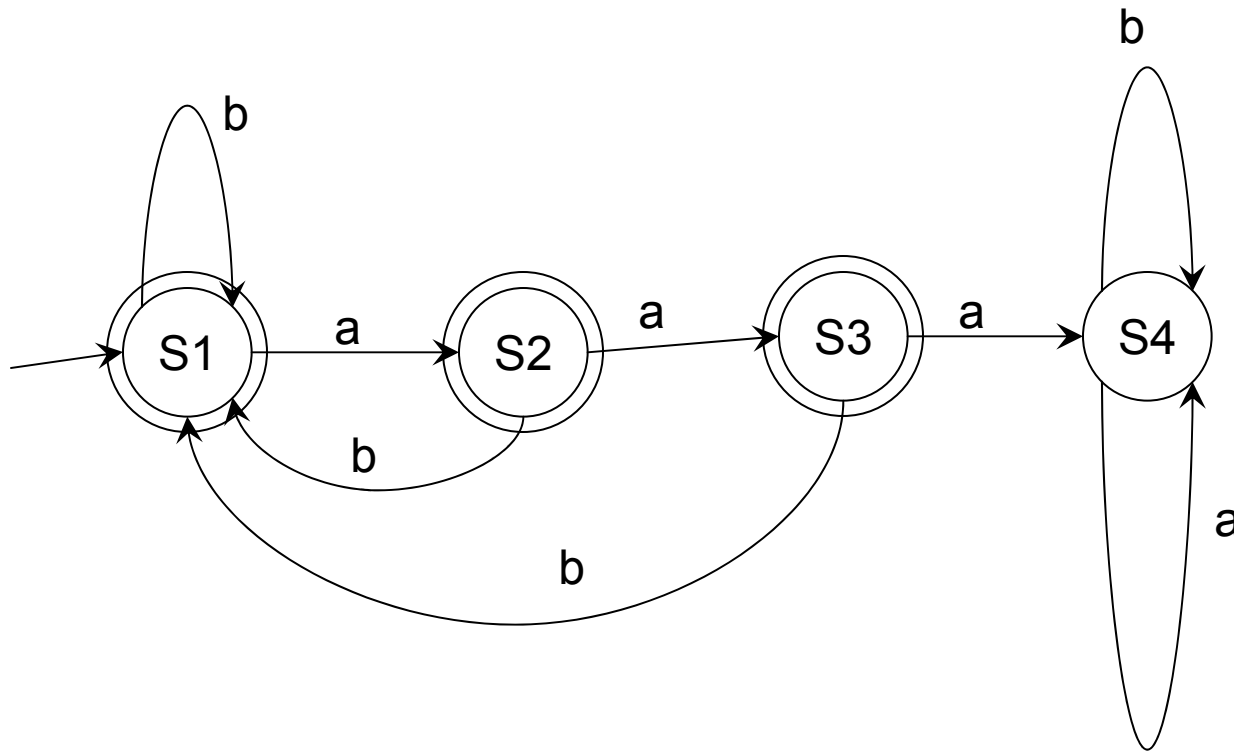


DFA Construction

- Intuition behind the construction
 - Every time you get a problem “strings such that **condition x** is satisfied”, create a DFA with a “core” which checks for **condition x**.
 - In the previous example, $S1 \rightarrow S2 \rightarrow S3 \rightarrow S4$ is the “core”
 - If the condition is satisfied, “trap” it in an accepting state
 - In the previous example, once you have checked it for 3 consecutive a’s, you are “trapped” in the accepting state S4. That is, subsequent a’s and b’s will cause you to remain in the accepting state
 - If during the check for the condition, the condition is violated start over
 - In the previous example, assume we had a sub-sequence “aab”. Upon encountering the “b”, we start over by going back to state S1 from state S3

DFA Construction

- Strings of a's and b's such that there 3 a's **do not** appear consecutively

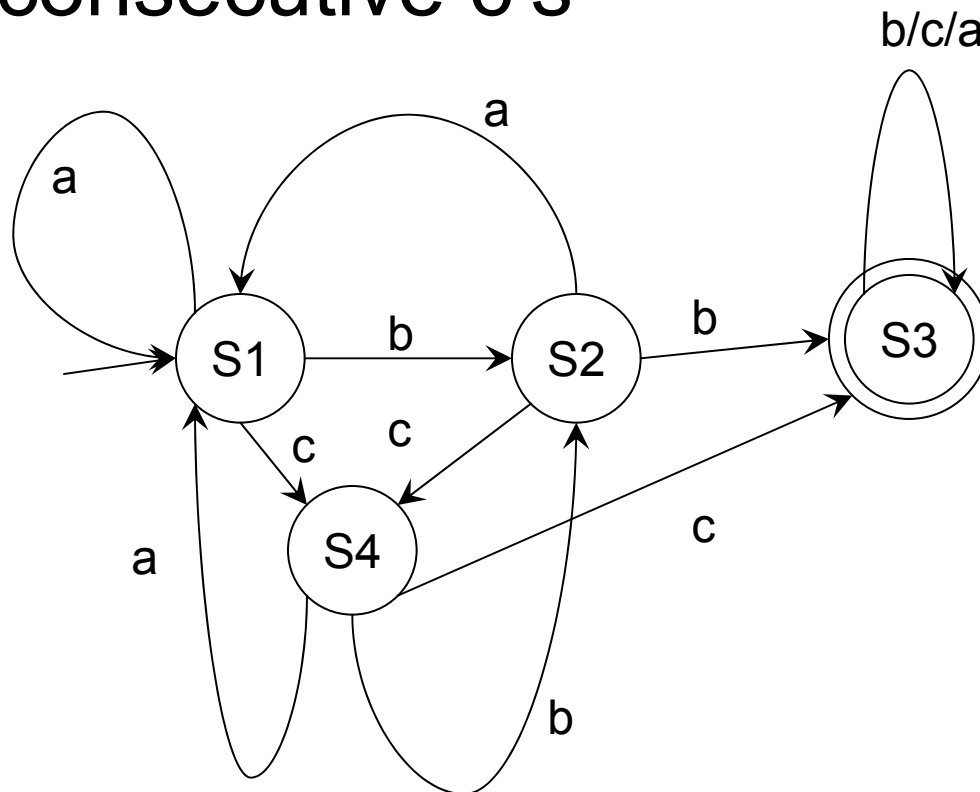


DFA Construction

- Intuition behind the construction
 - Every time you get a problem “strings such that **condition x** is not satisfied”, create a DFA with a “core” which checks for **condition x**.
 - In the previous example, $S1 \rightarrow S2 \rightarrow S3 \rightarrow S4$ is the “core”
 - If the condition is satisfied, “trap” it in an *rejecting* state
 - In the previous example, once you have checked it for 3 consecutive a’s, you are “trapped” in the *rejecting* state $S4$. That is subsequent a’s and b’s will cause you to remain in $S4$ and you cannot ever go to the accepting state from $S4$
 - Mark all the other states as accepting state
 - If during the check for the condition, the condition is violated start over
 - In the previous example, assume we had a sub-sequence “aab”. Upon encountering the “b”, we start over by going back to state $S1$
- Note
 - If you construct a DFA “D” for checking “**condition x**”, a new DFA such that all accepting states of “D” are marked as normal states and all normal states of “D” are marked as accepting states, then the new DFA checks for the inverse of condition x. Compare the solution of problem 1 with the solution for problem 2, and convince yourself that this is true

DFA Construction

- Strings of a's b's and c's such that there are at least two consecutive b's or two consecutive c's



DFA Construction

- Intuition behind the construction
 - Every time you get a problem “strings such that **condition x** or **condition x** satisfied”, create a DFA with a “core” which checks for **condition x**.
 - In the previous example, $S1 \rightarrow S2 \rightarrow S3$ is the “core”
 - If the condition is satisfied, “trap” it in an accepting state
 - In the previous example, once you have checked it for two consecutive b’s, you are “trapped” in the accepting state S3. That is, subsequent a’s, b’s and c’s will cause you to remain in the accepting state
 - If during the check for the condition, the condition is violated start over
 - In the previous example, assume we had a sub-sequence “ba”. Upon encountering the “b”, we start over by going back to state S1
 - Cont on next slide....

DFA Construction

- Similarly create a “core” for **condition y**
- We now have to “compose” these two DFA’s together
 - Merge the “start” states
 - Merge the “accept” states (if possible)
 - Take the DFA that checks for **condition x** and add transitions to states of the DFA that checks for **condition y** and vice-versa
 - $S_2 \rightarrow S_4$ and $S_4 \rightarrow S_2$ respectively

DFA Construction

- Strings of a's and b's such that there are at least two consecutive b's and at least two consecutive c's (not necessarily in that order)
 - Answer: Try it yourself
- Intuition
 - Everytime you get a problem “check for **condition x** and **condition y**”
 - Create a “core” that checks for **condition x** and then **condition y**
 - Create a “core” that checks for **condition y** and then **condition x**
 - Compose the two DFA's by merging the start states and merging the accept states (if possible)

DFA Construction

- You now know how to
 - Check for condition x
 - Check for **not** condition x
 - Check for condition x **or** condition y
 - Check for condition x **and** condition y
- Using this, you can construct a general DFA