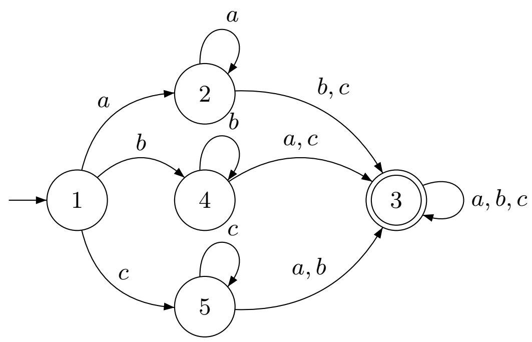
Ex. 1

Given the alphabet $X = \{a, b, c\}$, propose a deterministic finite-state automaton (not necessarily complete) which recognizes all the words of X^* which contain at least two different letters.

.... Answer ....

In adition to the initial state, cases that have to be distingued are the case where having found an a, we expect either a b or a c, the case where having found a b, we expect either an a or a c, *etc.*
Here is the resulting automaton:



Ex. 2

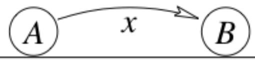
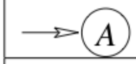
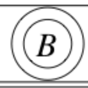
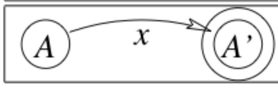
Consider the following regular grammar:

$$\begin{array}{lcl} S & \rightarrow & aA \mid bB \\ B & \rightarrow & aA \mid bC \mid b \\ A & \rightarrow & bB \mid aC \mid a \\ C & \rightarrow & aC \mid a \mid b \mid bC \end{array}$$

1. Build the finite-state automaton corresponding to this grammar (hint: the states of the automaton correspond closely to the non-terminal symbols of the grammar).
2. Show the sequences of states corresponding to the recognition path of the words *aaa*, *babba* and *babaaaa*.
3. Is this automaton deterministic? If not, propose a deterministic finite-state automaton recognizing the same language.

. . . . Answer . . . .

1. It's possible to build an automaton by applying a systematic correspondence between rules of a regular grammar and transitions in an automaton:

	$A \to xB$
	Axiome $= A$
	$B \to \varepsilon$
 (A' nouvel état)	$A \to x$

   In our specific case, this will lead to the automaton on the left (the states are labelled with letters to show the correspondence).

2. $S^a A^a C^a Z$

   $S^b B^a A^b B^b C^a Z$

   $S^b B^a A^b B^a A^a C^a C^a Z$

3. The automaton on the left is easy to make deterministic by essentially merging the two states C and Z. The result is given here on the right hand side.