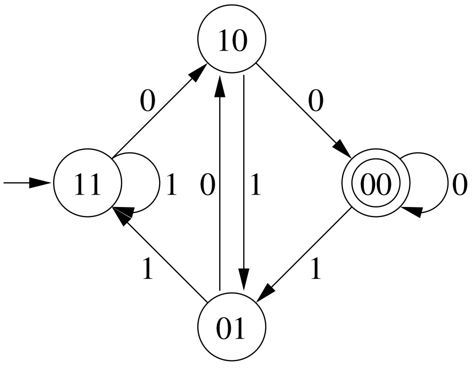
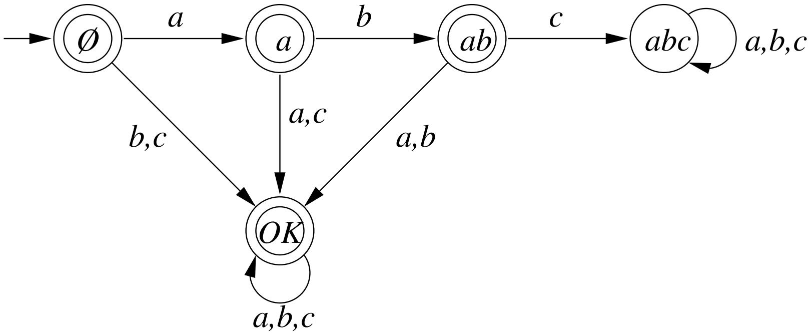
1. Multiples of 4 always end with "00" in binary notation. So our DFA simply accepts every string that ends with "00" and rejects all other strings. Our DFA has states that represent the last two symbols encountered, where "missing" symbols (at the beginning of the string) default to 1.



Formally, we could prove that our DFA is correct because of the following state invariant: for all strings $s \in \{0,1\}^*$,

$$\delta^*(11,s) = \begin{cases} 11 & \text{iff } s \text{ ends with "11" or } s = 1 \text{ or } s = \varepsilon \\ 10 & \text{iff } s \text{ ends with "10" or } s = 0 \\ 01 & \text{iff } s \text{ ends with "01"} \\ 00 & \text{iff } s \text{ ends with "00"} \end{cases}$$

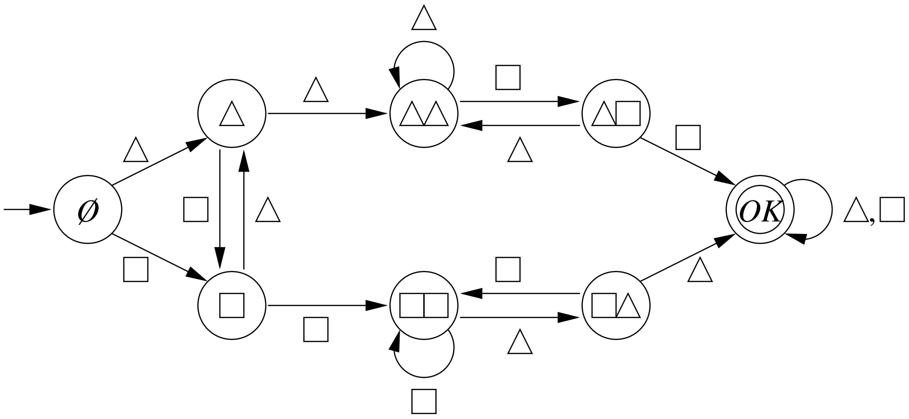
2. Make sure that processing "*abc*" at the beginning leads to a dead state, and all other possibilities lead to an accepting state.



Formally, we could prove that our DFA is correct because of the following state invariant: for all strings $s \in \{a,b\}^*$,

$$\delta^*(\varnothing,s) = \begin{cases} \varnothing & \text{iff } s = \varepsilon \\ a & \text{iff } s = a \\ ab & \text{iff } s = ab \\ abc & \text{iff } s \text{ starts with "}abc\text{"} \\ OK & \text{iff } (|s| = 1 \text{ and } s \neq a) \text{ or } (|s| = 2 \text{ and } s \neq ab \text{ or } s \text{ does \textbf{not} start with "}a\text{"}) \text{ or} \\ & \quad (|s| \geqslant 3 \text{ and } s \text{ does not start with "}abc\text{"}) \end{cases}$$

3. We need to keep track of the last character seen so far, and of whether or not we've seen two △'s or two □'s in a row so far.



Formally, we could prove that our DFA is correct because of the following state invariant: for all strings $s \in \{\triangle, \square\}^*$,

$$\delta^*(\varnothing, s) = \begin{cases} \varnothing & \text{iff } s = \varepsilon \\ \triangle & \text{iff } s \text{ ends with } \triangle \text{ and contains neither } \triangle\triangle \text{ nor } \square\square \\ \square & \text{iff } s \text{ ends with } \square \text{ and contains neither } \triangle\triangle \text{ nor } \square\square \\ \triangle\triangle & \text{iff } s \text{ ends with } \triangle \text{ and contains } \triangle\triangle \text{ but not } \square\square \\ \square\square & \text{iff } s \text{ ends with } \square \text{ and contains } \square\square \text{ but not } \triangle\triangle \\ \triangle\square & \text{iff } s \text{ ends with } \square \text{ and contains } \triangle\triangle \text{ but not } \square\square \\ \square\triangle & \text{iff } s \text{ ends with } \triangle \text{ and contains } \square\square \text{ but not } \triangle\triangle \\ OK & \text{iff } s \text{ contains both } \triangle\triangle \text{ and } \square\square \end{cases}$$