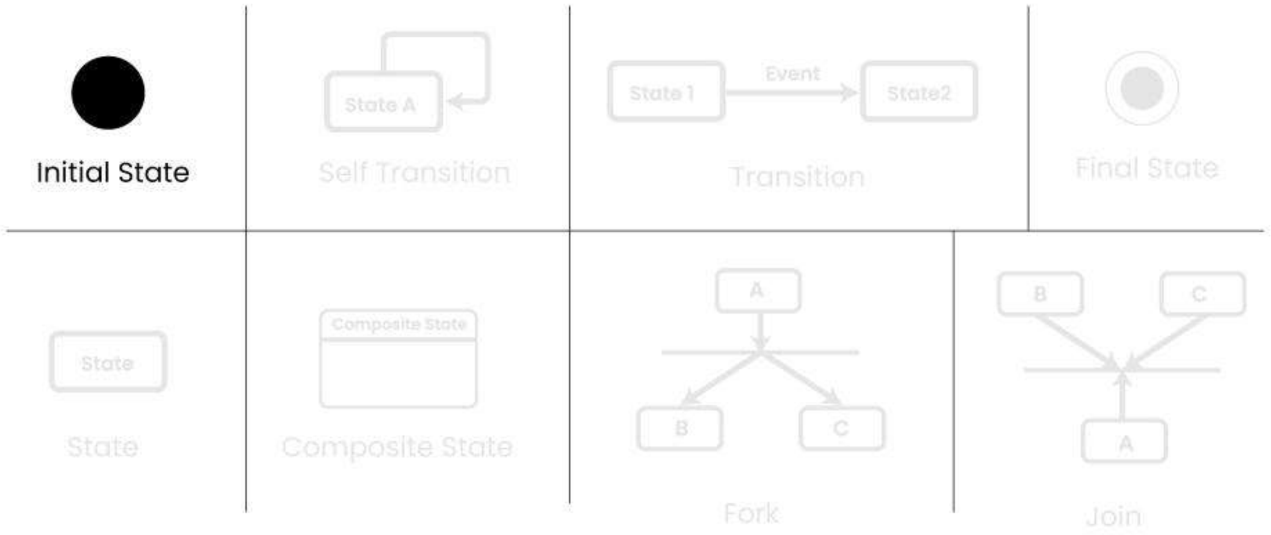
Basic components and notations of a State Machine diagram

Below are the basic components and their notations of a State Machine Diagram:

1. Initial state

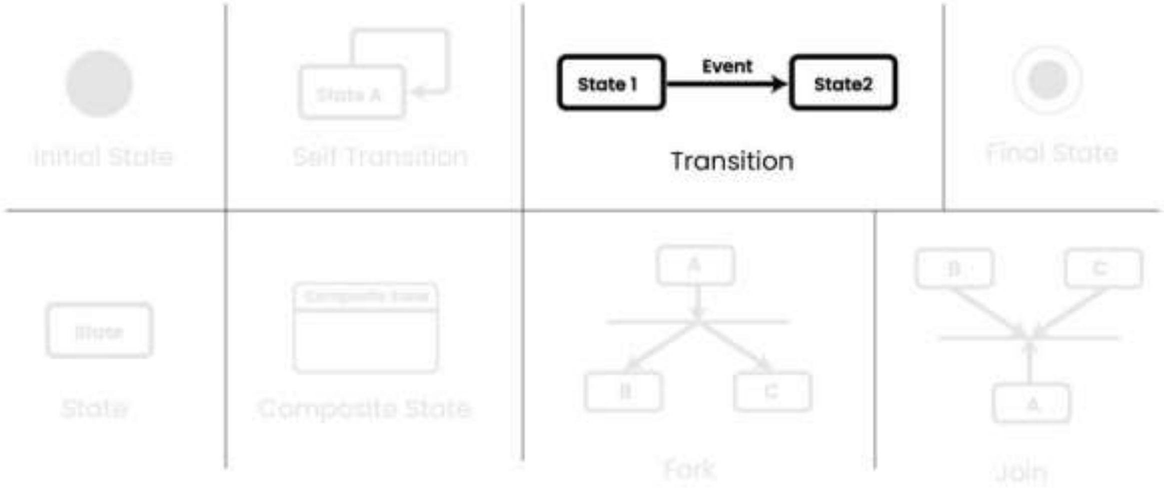
We use a black filled circle represent the initial state of a System or a Class.



initial State

2. Transition

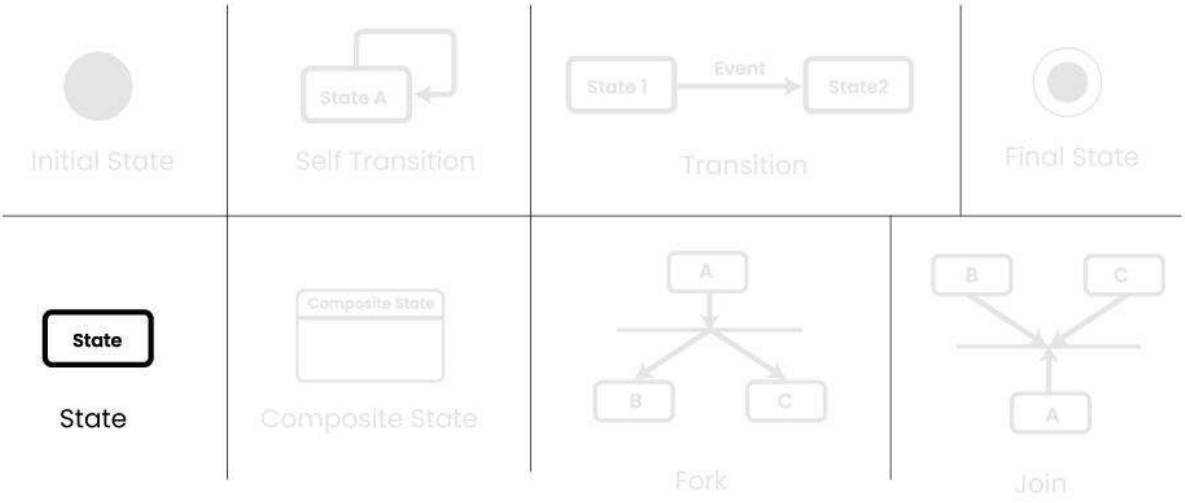
We use a solid arrow to represent the transition or change of control from one state to another. The arrow is labelled with the event which causes the change in state.



Transition

3. State

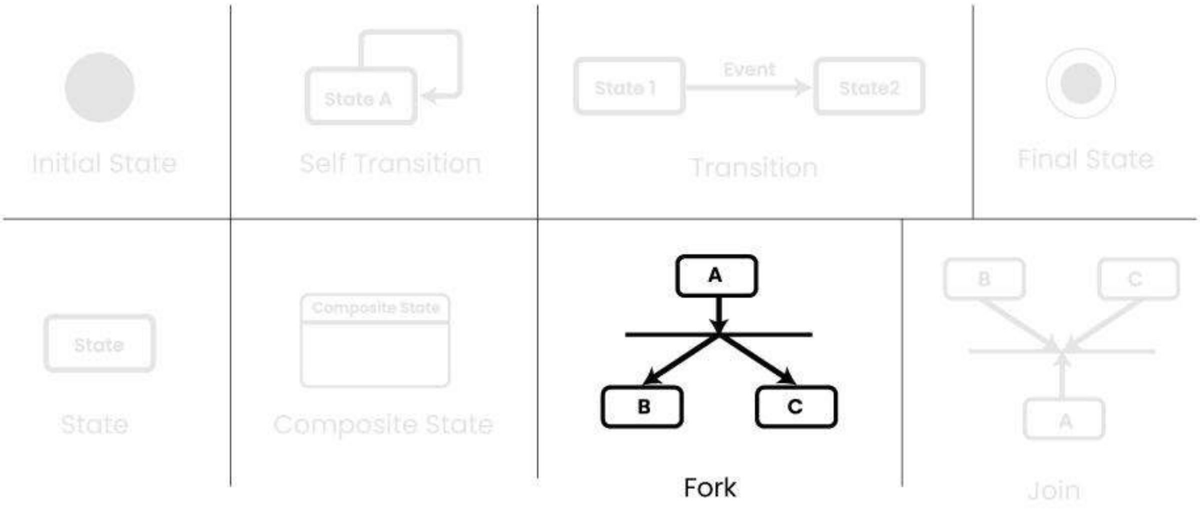
We use a rounded rectangle to represent a state. A state represents the conditions or circumstances of an object of a class at an instant of time.



State

4. Fork

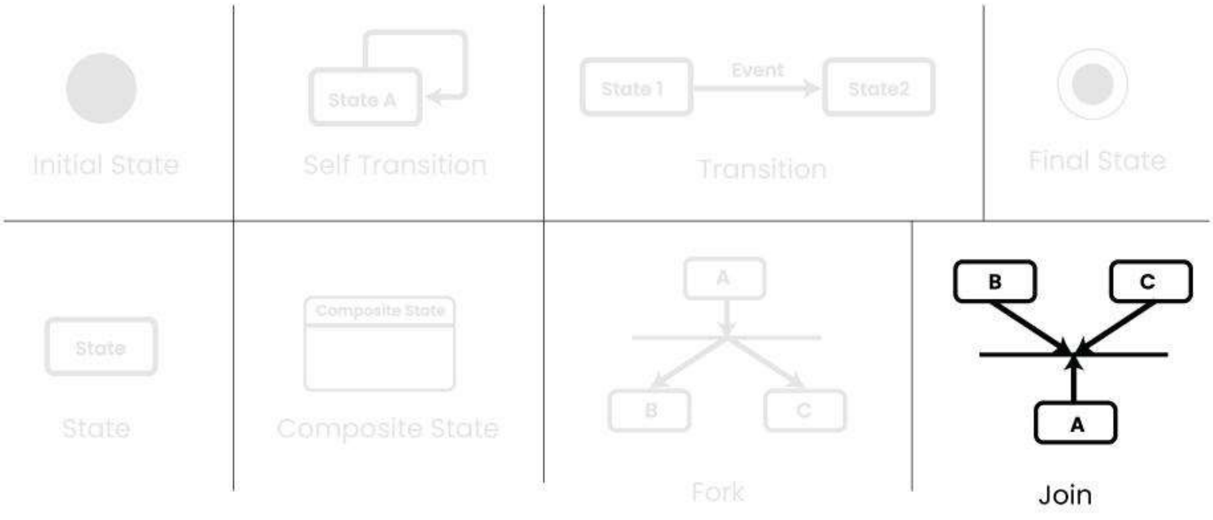
We use a rounded solid rectangular bar to represent a Fork notation with incoming arrow from the parent state and outgoing arrows towards the newly created states. We use the fork notation to represent a state splitting into two or more concurrent states.



Fork

5. Join

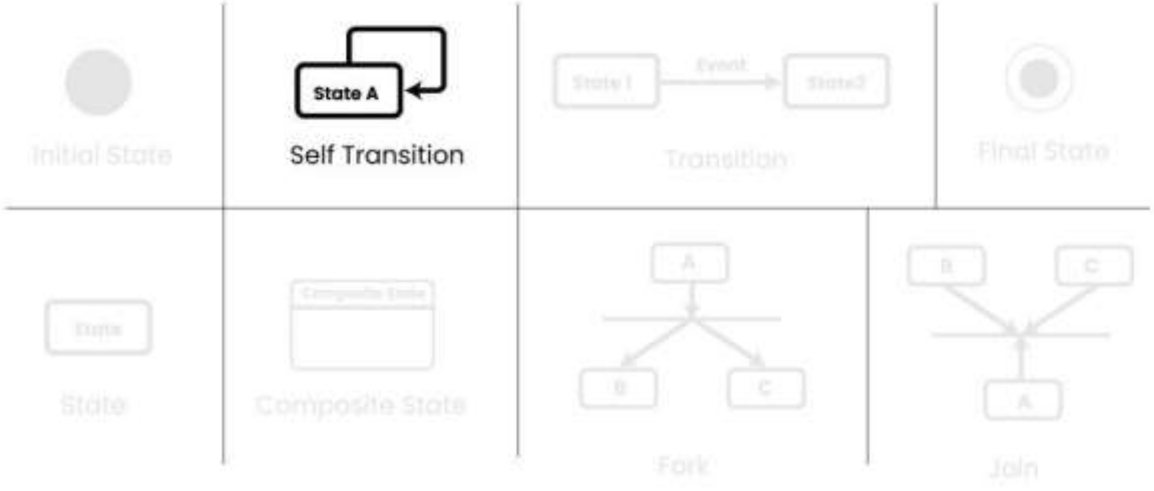
We use a rounded solid rectangular bar to represent a Join notation with incoming arrows from the joining states and outgoing arrow towards the common goal state. We use the join notation when two or more states concurrently converge into one on the occurrence of an event or events.



Join

6. Self-transition

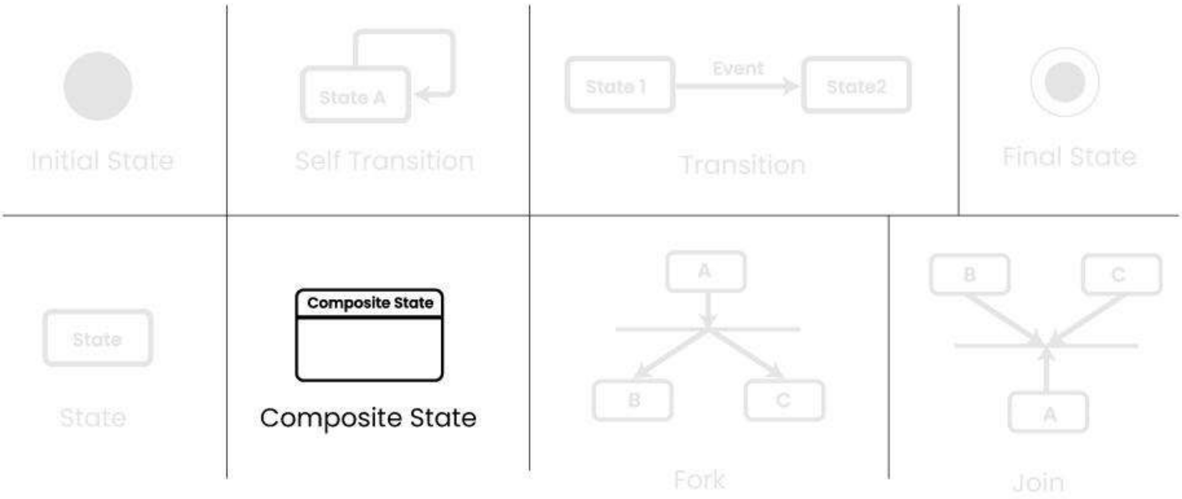
We use a solid arrow pointing back to the state itself to represent a self-transition. There might be scenarios when the state of the object does not change upon the occurrence of an event. We use self-transitions to represent such cases.



Self transition

7. Composite state

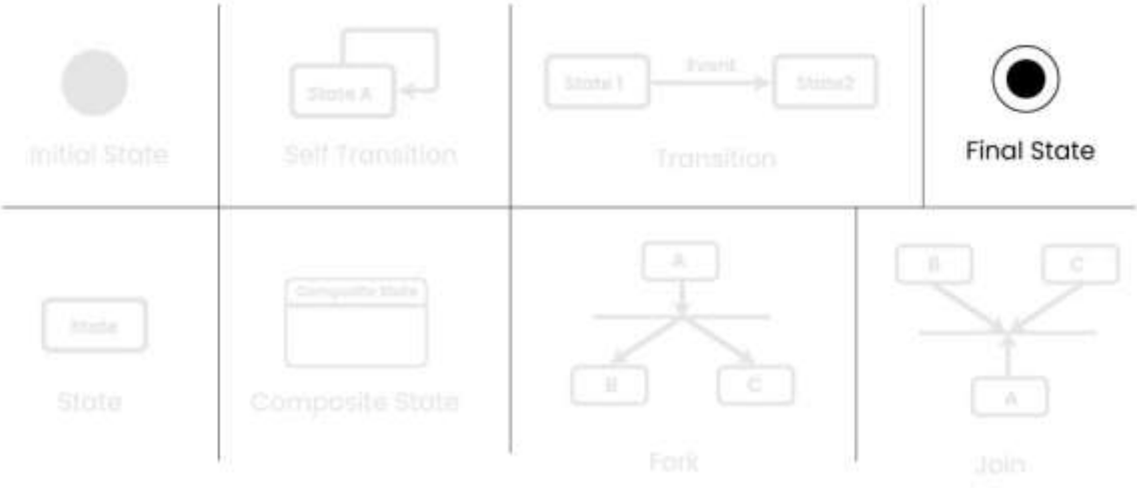
We use a rounded rectangle to represent a composite state also. We represent a state with internal activities using a composite state.



Composite State

8. Final State

We use a filled circle within a circle notation to represent the final state in a state machine diagram.



Final State