

In a class diagram they both look similar to each other but have some differences.

Similarities

1. It is easy to add new state or strategy without affecting Object itself, which uses them.
2. State pattern wraps different behaviors in form of different State objects, while Strategy pattern wraps different behavior in form of different Strategy objects.
3. Both Strategy and State Patterns relies on subclasses to implement behavior.

Differences

They are similar in structure, but their intents are different.

1. Strategy pattern encapsulate a set of algorithms, and allow user to use interchangeable behaviors through composition. On the other hand, State pattern help a class to exhibit different behaviors in different state.
2. State encapsulate state of an Object, while Strategy encapsulate an algorithm or strategy. Because each state is associated with an object, it cannot be reused, but we can reuse strategy by separating them from its context.
3. In State pattern, individual state can contain reference of Context, but Strategy does not contain reference of Context, which they are used.
4. Order of State transition is well defined in State Pattern, which means they will change if meet valid condition. There is no such requirement for strategy pattern, the user can choose any Strategy of his choice.
5. If you see your code needs to use different kind of algorithms, then using Strategy Pattern. If you need to manage state and state transition, without condition statements, state pattern is to use.
6. Change in Strategy is done by User, but Change in State can be done by Context or State object itself.

Có gì mấy bạn tham khảo thêm web này để hiểu hơn về điểm giống và khác nhau về strategy pattern mình chỉ ghi gì những gì mình hiểu nên có thể thiếu sót.

<http://javarevisited.blogspot.com/2014/04/difference-between-state-and-strategy-design-pattern-java.html>