Given two sequential circuits, starting from their respective initial states, do they have the same I-O response? i.e., for all possible input sequences, do they produce the same output sequence?

$C_1$ may not be equivalent to $C_2$, but $S_1 \equiv S_2$.

Examples:
Sequential Equivalence required:

1. Sequential optimizations
   - retiming changes the # of D-FFs.
   - code reassignment

```
# code I
S0 = 00
S1 = 01
S2 = 10
S3 = 11
```

```
# code II
S0 = 11
S1 = 10
S2 = 00
S3 = 01
```

but I-O behaviour is the same.

but combinational logic \( \neq \) same.

2. New problems.

   equivalence of symbolic C++ model
   \( \equiv \) RTL (Clocked Verilog)

This problem slightly different from "testing" (sequential ATPG).
$\textit{Seq. verif} = \textit{state space analysis.}$

$M_1 = $

$M_2 = $

$R \& R' = \text{corresponding initial states.}$

Machine $M_1 \equiv M_2$ if:

1. Both are identical
2. Identical states, but different encodings
3. $M \preceq M_2$ or $M_1 \unlhd M_2$
4. Different reachable states, but same distinguishable states.
5. Different unreachable states, but unreachable states = Don't care.
our problem:–

1) given two FSMs (i.e. given their state tables or graphs)
   \[ M_1 \equiv M_2 \]?

2) Given two circuits, but not their state tables, are they equivalent?

1 \rightarrow "easier"

2 \rightarrow extract the underlying FSMs, & then prove equivalence.

Do this efficiently as a CAD Solution.

Requires:—" Implicit state enumeration."

- Product FSM
- Implicit state enumeration, BFS-traversal
- BDDs.
- equivalence check.
FSM defined as:

\[ M = (\Sigma, \Omega, S, s^0, \Delta, \Lambda) \]

\( \Sigma \) = input label
\( \Omega \) = output label
\( S \) = set of states
\( s^0 \subset S \) = initial (reset) state(s)
\( \Delta : S \times \Sigma \rightarrow S \), next state transition function.
\( \Lambda : S \times \Sigma \rightarrow \Omega \), output function.

For FSM traversal, we will use image computation.

\[ N = \text{Image of } F \text{ under } C = \text{Image}(F, C) \]
First we will understand how to solve \( M_1 = M_2 \) assuming STGs are given.

Then, we will see how to apply those concepts on a circuit.

**Product Machine**

\[
M_1 = \{ \varepsilon, 0, S', S_0^1, \Delta, \lambda^3 \} \\
M_2 = \{ \varepsilon, 0, S^2, S_0^2, \Delta, \lambda^3 \} \\
M_{12} = \{ \varepsilon, 0^{12}, S_0^{12}, \Delta, \lambda^{12} \}
\]
$s' \in S^1, s^2 \in S^2$

Product of states $S^{12} = s' \times s^2 = (s'_1, s^2_2)$ (concatenation).

$\Delta'(s', x) \times \Delta(s^2, x)$

($x \in \Sigma =$ input)

$\Delta^{12} = \Delta^{12}(s^{12}, x) : (s' \times s^2) \times \Sigma \rightarrow (s' \times s^2)$

$\Rightarrow [(s' \times \Sigma) \rightarrow s'] [ (s^2 \times \Sigma) \rightarrow s^2 ]$

$\Rightarrow [\Delta'(s', x), \Delta(s^2, x)]$

$\Rightarrow$ Conjunction of transition relations.

Output $\Lambda^{12} = \Lambda^{12}(s, x) : (s' \times s^2) \times \Sigma \rightarrow \{0, 1\}$

$\exists \Lambda^{12} = 1$ if $\chi(s', x) = \chi(s^2, x)$

$= 0$ otherwise.