Y, Y', Z, W are conceptually different "operations" we want to perform on the state elements.

Whether or not we want to actually "do" Y, Y', Z, W depends on various other conditions.

All four operations might be enabled at the same time.

Assume Y, W always enabled.

**RTL Modeling**

assign a = (x') ? f0(a, b) : (z') ? h0(a, b);

assign b = (x') ? f1(a, b) : (z') ? h1(a, b);

assign c = g(a, b)

assign e = v(d)

**State Centric**

Must group logic by the state it updates instead of by operation.

Execution semantics: explicitly clocked, must explicitly manage moving between different operations.
Guaranteed atomic access meaning

\texttt{guard}

\begin{align*}
\text{rule } r_1 &: (x_1) : \\
& a = f_0 (a, b) ; \\
& b = +1 (a, b) ; \\
& \text{end rule} \\
\text{rule } r_2 &: (x_2) : \\
& a = h_0 (a, b) ; \\
& b = h_1 (a, b) ; \\
& \text{end rule} \\
\end{align*}

\texttt{operation-centric}

\begin{quote}
\text{group logic by operation, mixing managed by compiler}
\end{quote}

\begin{quote}
\text{no guard, always enabled}
\end{quote}

\texttt{execute semantics: untimed, pick one enabled rule, fire the rule, pick again}

\texttt{rules fire atomically and in isolation}

\texttt{semantics:}

\begin{align*}
| & \rightarrow | \rightarrow | \rightarrow | \rightarrow | \rightarrow | \rightarrow | \rightarrow | \\
Y & \downarrow X & W & Y & W & \uparrow & W & Y & X \\
\end{align*}

\begin{quote}
\text{can always fire w since it is conflict-free}
\end{quote}

\begin{quote}
\text{only } X \text{ or } \uparrow \text{ can fire every cycle because they are mutually exclusive}
\end{quote}

\begin{quote}
\text{Y can always fire in parallel, but it always appears to happen before } X \text{ or } \uparrow \text{ in the serial execution semantics}
\end{quote}
What are possible values for $x, y$ after an execution?

Assume initially $x = 1, y = 2$

```plaintext
rule rA:
  x := y;
end rule

rule rB:
  y := x;
end rule
```

- $rA < rB : \quad x = 2, \; y = 2$
- $rB < rA : \quad x = 1, \; y = 1$
- $rA : \quad x = 2, \; y = 2$
- $rB : \quad x = 1, \; y = 1$
- $\phi : \quad x = 1, \; y = 2$
- $rA < rA < rB : \quad x = 2, \; y = 2$
```
For each set of two rules explain if these rules can be enforced at the same time, and if so can they be executed concurrently?

1) rule $ra (z > 10);$
   \[ x \leftarrow x + 1; \]
   end rule
   rule $rb (z > 20);$
   \[ y \leftarrow y + 1; \]
   end rule
   enabled? yes
   concurrently? yes
   $ra < rb$
   $rb < ra$

2) rule $ra (z > 10);$
   \[ x \leftarrow y + 1; \]
   end rule
   rule $rb (z > 20);$
   \[ y \leftarrow x + 1; \]
   end rule
   enabled? yes
   concurrently? no

3) rule $ra (z > 10);$
   \[ x \leftarrow y + z; \]
   end rule
   rule $rb (z > 20);$
   \[ y \leftarrow y + z; \]
   end rule
   enabled? yes
   concurrently? yes
   $ra < rb$

4) rule $ra (z > 10);$
   \[ x \leftarrow y + 1; \]
   end rule
   rule $rb (z > 20);$
   \[ x \leftarrow y + 2; \]
   end rule
   enabled? yes
   concurrently? yes

Describing an update logic:
it will appear as if
$ra < rb$ or $rb < ra$