CrashCourse — Verification of Finite Automata

CTL Model-Checking

Tibor Schneider

TIK
Formulation of CTL properties

Based on atomic propositions ($\phi$) and quantifiers

- $A\phi \rightarrow \text{«All } \phi\text{»}$, $\phi$ holds on all paths
- $E\phi \rightarrow \text{«Exists } \phi\text{»}$, $\phi$ holds on at least one path
- $X\phi \rightarrow \text{«Ne\text{xt } } \phi\text{»}$, $\phi$ holds on the next state
- $F\phi \rightarrow \text{«Finally } \phi\text{»}$, $\phi$ holds at some state along the path
- $G\phi \rightarrow \text{«Globally } \phi\text{»}$, $\phi$ holds on all states along the path
- $\phi_1 U\phi_2 \rightarrow \text{«}\phi_1 \text{Until } \phi_2\text{»}$, $\phi_1$ holds until $\phi_2$ holds

Quantifiers over paths
Path-specific quantifiers
Formulation of CTL properties

Proper CTL formula: \( \{A,E\} \{X,F,G,U\} \phi \)

→ Quantifiers **go by pairs**, you need one of each.

**Missing Hypothesis**

Interpretation on CTL formula

→ Transition functions are **fully defined**
  (i.e. every state has at least one successor)
Inverting properties is sometimes useful!

\[
\begin{align*}
AG \phi & := \neg EF \neg \phi \\
AF \phi & := \neg EG \neg \phi \\
EF \phi & := \neg AG \neg \phi \\
EG \phi & := \neg AF \neg \phi
\end{align*}
\]
Inverting properties is sometimes useful!

“On all paths, for all states, $\phi$ holds” $\iff$ “There exists no path along which at some state $\phi$ doesn’t hold.”

$\text{AG } \phi := \neg \text{EF } \neg \phi$

$\text{AF } \phi := \neg \text{EG } \neg \phi$

$\text{EF } \phi := \neg \text{AG } \neg \phi$

$\text{EG } \phi := \neg \text{AF } \neg \phi$
Inverting properties is sometimes useful!

\[ \text{AG } \varphi := \neg \text{EF } \neg \varphi \]

“On all paths, for all states, \( \varphi \) holds” ⇔

“There exists no path along which at some state \( \varphi \) doesn’t hold.”

\[ \text{AF } \varphi := \neg \text{EG } \neg \varphi \]

“On all paths, there exists a state where \( \varphi \) holds” ⇔ “There exists no path along which \( \varphi \) doesn’t hold for all states.”

\[ \text{EF } \varphi := \neg \text{AG } \neg \varphi \]

\[ \text{EG } \varphi := \neg \text{AF } \neg \varphi \]

...
Inverting properties is sometimes useful!

“On all paths, for all states, \( \phi \) holds” \( \iff \)
“There exists no path along which at some state \( \phi \) doesn’t hold.”

\[
AG \phi := \neg EF \neg \phi \\
AF \phi := \neg EG \neg \phi \\
EF \phi := \neg AG \neg \phi \\
EG \phi := \neg AF \neg \phi
\]

“On all paths, there exists a state where \( \phi \) holds” \( \iff \) “There exists no path along which \( \phi \) doesn’t hold for all states.”

Remark: There exists other temporal logics.

- LTL (Linear Temporal Logic)
- CTL* = \{CTL, LTL\}
- ...
How to verify CTL properties?

**Convert the property verification into a reachability problem**

1. Start from states in which the property holds;
2. Compute all predecessor states for which the property still holds true; (same as for computing successor, with the inverse the transition function)
3. If initial states set is a subset, the property is satisfied by the model.

*Computation specifics are described in the lecture slides.*
So... what is Model-Checking exactly?

An algorithm

**Input**
- A DES model, $M$
  - Finite automata,
  - Petri nets,
  - Kripke machine, ...
- A logic property, $\phi$
  - CTL,
  - LTL, ...

**Output**
- $M \models \phi$?
- A trace for which the property does not hold!
Your turn to work!
Ex 1a) Temporal Logic

(i) EF \( a : Q = \{0, 1, 2, 3\} \)

(ii) EG \( a : Q = \{0, 3\} \)

(iii) Build the set step-by-step:

\[
\begin{align*}
AX a : Q_1 &= \{2, 3\} \\
EX AX a : Q_2 &= \{1, 2\}
\end{align*}
\]

(iv) Build the set step-by-step:

\[
\begin{align*}
\neg a : Q_1 &= \{1, 2\} \\
EX \neg a : Q_2 &= \{0, 1\} \\
a \land EX \neg a : Q_3 &= \{0\} \\
EF(a \land EX \neg a) : Q_4 &= \{0, 1, 2, 3\}
\end{align*}
\]
Ex 1b) Temporal Logic

(i) \( \neg \text{AF } Z = \text{EG } \neg Z \implies \text{AF } Z = \neg \text{EG } \neg Z \)

(ii) • To get \( \text{EG } \neg Z \) iteratively, we start with \( Q = \{ q : q \notin Z \} \).
    • At each step, require each state \( q \in Q \) to have \( \exists q' \in Q \cup f(q) \).
      \( \quad \text{This will only remove states in } Q \)
      • Stop as soon as nothing changes anymore.

Require: \( \psi_Z, \psi_f \)

\[
Q_0 = S \setminus Z \\
Q_{i+1} = Q_i \cap \text{pred}(Q_i, f) \\
k = \min\{i \mid Q_{i+1} = Q_i\} \\
Q_{\text{AF } Z} = Z \setminus Q_k
\]

\[
\psi_{\text{cur}} \leftarrow \neg \psi_Z \\
\psi_{\text{next}} \leftarrow \psi_{\text{cur}} \land \psi_{\text{pred}(\psi_{\text{cur}}, f)} \\
\text{while } \psi_{\text{cur}} \neq \psi_{\text{next}} \text{ do} \\
\quad \psi_{\text{cur}} \leftarrow \psi_{\text{next}} \\
\quad \psi_{\text{next}} \leftarrow \psi_{\text{cur}} \land \psi_{\text{pred}(\psi_{\text{cur}}, f)} \\
\text{end while} \\
\text{return } \psi_{\text{AF } Z} = \neg \psi_{\text{cur}}
Ex 2a) Find all possible loops.

(1) \( \rho_1(v_0) = v_0 \)
Ex 2a) Find all possible loops.

(1) $\rho_1(v_0) = v_0$
(2) $\rho_2(v_0) = v_1$, $\rho_2(v_1) = v_1$
Ex 2a) Find all possible loops.

(1) $\rho_1(v_0) = v_0$
(2) $\rho_2(v_0) = v_1$, $\rho_2(v_1) = v_1$
(3) $\rho_3(v_0) = v_1$, $\rho_3(v_1) = v_0$
Ex 2a) Find all possible loops.

(1) $\rho_1(v_0) = v_0$

(2) $\rho_2(v_0) = v_1$, $\rho_2(v_1) = v_1$

(3) $\rho_3(v_0) = v_1$, $\rho_3(v_1) = v_0$

(4) $\rho_4(v_0) = v_1$, $\rho_4(v_1) = v_2$, $\rho_4(v_2) = v_2$
Ex 2a) Find all possible loops.

1. \( \rho_1(v_0) = v_0 \)
2. \( \rho_2(v_0) = v_1, \rho_2(v_1) = v_1 \)
3. \( \rho_3(v_0) = v_1, \rho_3(v_1) = v_0 \)
4. \( \rho_4(v_0) = v_1, \rho_4(v_1) = v_2, \rho_4(v_2) = v_2 \)
5. \( \rho_5(v_0) = v_1, \rho_5(v_1) = v_2, \rho_5(v_2) = v_1 \)
Ex 2a) Find all possible loops.

(1) $\rho_1(v_0) = v_0$
(2) $\rho_2(v_0) = v_1$, $\rho_2(v_1) = v_1$
(3) $\rho_3(v_0) = v_1$, $\rho_3(v_1) = v_0$
(4) $\rho_4(v_0) = v_1$, $\rho_4(v_1) = v_2$, $\rho_4(v_2) = v_2$
(5) $\rho_5(v_0) = v_1$, $\rho_5(v_1) = v_2$, $\rho_5(v_2) = v_1$
(6) $\rho_6(v_0) = v_1$, $\rho_6(v_1) = v_2$, $\rho_6(v_2) = v_0$
Ex 2a) Find all possible loops.

(1) \[ \rho_1(v_0) = v_0 \]
(2) \[ \rho_2(v_0) = v_1, \rho_2(v_1) = v_1 \]
(3) \[ \rho_3(v_0) = v_1, \rho_3(v_1) = v_0 \]
(4) \[ \rho_4(v_0) = v_1, \rho_4(v_1) = v_2, \rho_4(v_2) = v_2 \]
(5) \[ \rho_5(v_0) = v_1, \rho_5(v_1) = v_2, \rho_5(v_2) = v_1 \]
(6) \[ \rho_6(v_0) = v_1, \rho_6(v_1) = v_2, \rho_6(v_2) = v_0 \]
(7) \[ \rho_7(v_0) = v_2, \rho_7(v_2) = v_2 \]
Ex 2a) Find all possible loops.

(1) \( \rho_1(v_0) = v_0 \)
(2) \( \rho_2(v_0) = v_1, \rho_2(v_1) = v_1 \)
(3) \( \rho_3(v_0) = v_1, \rho_3(v_1) = v_0 \)
(4) \( \rho_4(v_0) = v_1, \rho_4(v_1) = v_2, \rho_4(v_2) = v_2 \)
(5) \( \rho_5(v_0) = v_1, \rho_5(v_1) = v_2, \rho_5(v_2) = v_1 \)
(6) \( \rho_6(v_0) = v_1, \rho_6(v_1) = v_2, \rho_6(v_2) = v_0 \)
(7) \( \rho_7(v_0) = v_2, \rho_7(v_2) = v_2 \)
(8) \( \rho_8(v_0) = v_2, \rho_8(v_2) = v_0 \)
Ex 2a) Find all possible loops.

(1) \( \rho_1(v_0) = v_0 \)
(2) \( \rho_2(v_0) = v_1, \rho_2(v_1) = v_1 \)
(3) \( \rho_3(v_0) = v_1, \rho_3(v_1) = v_0 \)
(4) \( \rho_4(v_0) = v_1, \rho_4(v_1) = v_2, \rho_4(v_2) = v_2 \)
(5) \( \rho_5(v_0) = v_1, \rho_5(v_1) = v_2, \rho_5(v_2) = v_1 \)
(6) \( \rho_6(v_0) = v_1, \rho_6(v_1) = v_2, \rho_6(v_2) = v_0 \)
(7) \( \rho_7(v_0) = v_2, \rho_7(v_2) = v_2 \)
(8) \( \rho_8(v_0) = v_2, \rho_8(v_2) = v_0 \)
(9) \( \rho_9(v_0) = v_2, \rho_9(v_2) = v_1, \rho_9(v_1) = v_1 \)
Ex 2a) Find all possible loops.

(1) \( \rho_1(v_0) = v_0 \)
(2) \( \rho_2(v_0) = v_1, \rho_2(v_1) = v_1 \)
(3) \( \rho_3(v_0) = v_1, \rho_3(v_1) = v_0 \)
(4) \( \rho_4(v_0) = v_1, \rho_4(v_1) = v_2, \rho_4(v_2) = v_2 \)
(5) \( \rho_5(v_0) = v_1, \rho_5(v_1) = v_2, \rho_5(v_2) = v_1 \)
(6) \( \rho_6(v_0) = v_1, \rho_6(v_1) = v_2, \rho_6(v_2) = v_0 \)
(7) \( \rho_7(v_0) = v_2, \rho_7(v_2) = v_2 \)
(8) \( \rho_8(v_0) = v_2, \rho_8(v_2) = v_0 \)
(9) \( \rho_9(v_0) = v_2, \rho_9(v_2) = v_1, \rho_9(v_1) = v_1 \)
(10) \( \rho_{10}(v_0) = v_2, \rho_{10}(v_2) = v_1, \rho_{10}(v_1) = v_2 \)
Ex 2a) Find all possible loops.

(1) $\rho_1(v_0) = v_0$
(2) $\rho_2(v_0) = v_1$, $\rho_2(v_1) = v_1$
(3) $\rho_3(v_0) = v_1$, $\rho_3(v_1) = v_0$
(4) $\rho_4(v_0) = v_1$, $\rho_4(v_1) = v_2$, $\rho_4(v_2) = v_2$
(5) $\rho_5(v_0) = v_1$, $\rho_5(v_1) = v_2$, $\rho_5(v_2) = v_1$
(6) $\rho_6(v_0) = v_1$, $\rho_6(v_1) = v_2$, $\rho_6(v_2) = v_0$
(7) $\rho_7(v_0) = v_2$, $\rho_7(v_2) = v_2$
(8) $\rho_8(v_0) = v_2$, $\rho_8(v_2) = v_0$
(9) $\rho_9(v_0) = v_2$, $\rho_9(v_2) = v_1$, $\rho_9(v_1) = v_1$
(10) $\rho_{10}(v_0) = v_2$, $\rho_{10}(v_2) = v_1$, $\rho_{10}(v_1) = v_2$
(11) $\rho_{11}(v_0) = v_2$, $\rho_{11}(v_2) = v_1$, $\rho_{11}(v_1) = v_0$
Ex 2b) Encode the physical topology

$G$ is a complete graph, except for the edge between $v_0$ and $t$.

$\rho(v_0) \neq t$

$\psi_{\text{topo}}(Z) = \neg(z^1_0z^0_0)$
Ex 2c) Packets eventually reach $t$.

Enumerate all possible paths to reach $t$.

$$\psi_t(Z) = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$
Ex 2c) Packets eventually reach $t$.

Enumerate all possible paths to reach $t$.

$$\psi_t(Z) = z_0^1 z_0^0$$
Ex 2c) Packets eventually reach $t$. 

Enumerate all possible paths to reach $t$.

$$\psi_t(Z) = z_0^1z_0^0 + z_0^1z_0^0$$
Ex 2c) Packets eventually reach $t$.

Enumerate all possible paths to reach $t$.

$$\psi_t(Z) = z_0^1 z_0^0 + z_0^1 z_0^0 z_1^1 z_1^0$$

$$Z = \begin{cases} 
    z_0^1 z_0^0 & \rho(v_0) \\
    z_1^1 z_1^0 & \rho(v_1) \\
    z_2^1 z_2^0 & \rho(v_2) 
\end{cases}$$

$$\sigma(v_0) = 00 \quad \sigma(v_1) = 01 \quad \sigma(v_2) = 10 \quad \sigma(v_3) = 11$$
Ex 2c) Packets eventually reach $t$.

Enumerate all possible paths to reach $t$.

$$\psi_t(Z) = z_0^1 z_0^0 + z_0^1 z_0^0 (z_1^1 z_1^0 + z_1^1 z_1^0 z_2^1 z_2^0)$$
Ex 2c) Packets eventually reach \( t \).

Enumerate all possible paths to reach \( t \).

\[
\psi_t(Z) = z^1_0 z^0_0 \\
+ \bar{z}^1_0 z^0_0 \left( z^1_1 z^0_1 + z^1_1 \bar{z}^0_1 z^1_2 z^0_2 \right) \\
+ z^1_0 \bar{z}^0_0
\]

\[
Z = \begin{pmatrix}
    z^1_0 z^0_0 \\
    z^1_1 z^0_1 \\
    z^1_2 z^0_2
\end{pmatrix}
\]

\[
\sigma(v_0) = 00 \\
\sigma(v_1) = 01 \\
\sigma(v_2) = 10 \\
\sigma(v_3) = 11
\]
Ex 2c) Packets eventually reach \( t \).

Enumerate all possible paths to reach \( t \).

\[
\psi_t(Z) = z_0^1 z_0^0 \\
+ \bar{z}_0^1 z_0^0 \left( z_1^1 z_1^0 + z_1^1 \bar{z}_1^0 \ z_2^1 z_2^0 \right) \\
+ z_0^1 \bar{z}_0^0 \left( z_2^1 \bar{z}_2^0 \right)
\]

\[
Z = \begin{bmatrix}
    z_0^1 & z_0^0 & z_1^1 & z_1^0 & z_2^1 & z_2^0
\end{bmatrix}^{\rho(v_0)} \begin{bmatrix}
    z_1^1 & z_1^0 \end{bmatrix}^{\rho(v_1)} \begin{bmatrix}
    z_2^1 & z_2^0 \end{bmatrix}^{\rho(v_2)}
\]

\[
\sigma(v_1) = 01 \\
\sigma(v_0) = 00 \\
\sigma(v_2) = 10 \\
\sigma(v_3) = 11
\]
Ex 2c) Packets eventually reach $t$.

Enumerate all possible paths to reach $t$.

$$
\psi_t(Z) = z_0^1 z_0^0 \\
+ \bar{z}_0^1 z_0^0 (z_1^1 z_1^0 + z_1^0 \bar{z}_1^1 z_2^1 z_2^0) \\
+ z_0^1 \bar{z}_0^0 (z_2^1 z_2^0 + \bar{z}_2^1 z_2^1 z_1^0)
$$

$$
Z = \begin{array}{c}
\rho(v_0) \\
\rho(v_1) \\
\rho(v_2) \\
\end{array}
\begin{array}{c}
z_0^1 z_0^0 \\
z_1^1 z_1^0 \\
z_2^1 z_2^0 \\
\end{array}
$$

$$
\sigma(v_1) = 01 \\
\sigma(v_0) = 00 \\
\sigma(v_2) = 10 \\
\sigma(v_3) = 11
$$
Ex 2d) Packets must traverse $v_2$.

Enumerate all possible paths to reach $v_0$.

$$\psi_{v_2}(Z) =$$

$$Z = \begin{cases} z^1_0 z^0_0, & \rho(v_0) \\ z^1_1 z^0_1, & \rho(v_1) \\ z^1_2 z^0_2, & \rho(v_2) \end{cases}$$

$$\sigma(v_0) = 00 \quad \sigma(v_1) = 01 \quad \sigma(v_2) = 10 \quad \sigma(v_3) = 11$$
Ex 2d) Packets must traverse $v_2$.

Enumerate all possible paths to reach $v_0$.

$$\psi_{v_2}(Z) = z^1_0 z^0_0 +$$
Ex 2d) Packets must traverse $v_2$.

Enumerate all possible paths to reach $v_0$.

$$\psi_{v_2}(Z) = z_0^1 z_0^0 + \bar{z}_0^1 z_0^0 \ z_1^1 \bar{z}_1^0$$
Ex 2e) Packets must traverse $v_2$ and eventually reach $t$.

\[ \psi_\phi(Z) = \psi_t(Z) \cdot \psi_{v_2}(Z) = \]

\[ Z = \begin{pmatrix} z_0^1 z_0^0 \\ z_1^1 z_1^0 \\ z_2^1 z_2^0 \end{pmatrix} \begin{pmatrix} \rho(v_1) \\ \rho(v_0) \\ \rho(v_2) \end{pmatrix} \]

\[ \sigma(v_1) = 01 \]
\[ \sigma(v_0) = 00 \]
\[ \sigma(v_2) = 10 \]
\[ \sigma(v_3) = 11 \]
Ex 2e) Packets must traverse $v_2$ and eventually reach $t$.

$$\psi_\phi(Z) = \psi_t(Z) \cdot \psi_{v_2}(Z) = \overline{z_0^1 z_0^0}$$

$$Z = \begin{pmatrix}
z_0^1 z_0^0 \\
z_1^1 z_1^0 \\
z_2^1 z_2^0
\end{pmatrix}
\begin{cases}
\rho(v_0) \\
\rho(v_1) \\
\rho(v_2)
\end{cases}$$

$$\sigma(v_0) = 00$$
$$\sigma(v_2) = 10$$
$$\sigma(v_1) = 01$$
$$\sigma(v_3) = 11$$
Ex 2e) Packets must traverse $v_2$ and eventually reach $t$.

$$
\psi_\phi(Z) = \psi_t(Z) \cdot \psi_{v_2}(Z)
= z_0^1 z_0^0 \cdot z_1^1 z_1^0
$$

$$
\sigma(v_1) = 01
$$

$$
\sigma(v_0) = 00
$$

$$
\sigma(v_2) = 10
$$

$$
\sigma(v_3) = 11
$$

$$
Z = \begin{bmatrix}
    z_0^1 z_0^0 & z_1^1 z_1^0 & z_2^1 z_2^0 \\
    \rho(v_0) & \rho(v_1) & \rho(v_2)
\end{bmatrix}
$$
Ex 2e) Packets must traverse $v_2$ and eventually reach $t$.

$$
\psi_\phi(Z) = \psi_t(Z) \cdot \psi_{v_2}(Z)
= \bar{z}_0 z_0 \bar{z}_1 z_1 \bar{z}_2 z_2
$$

$$
Z = \left\{ \begin{array}{c}
\rho(v_0) \\
\rho(v_1) \\
\rho(v_2)
\end{array} \right\}
= \left\{ \begin{array}{c}
z_0^1 z_0^0 \\
z_1^1 z_1^0 \\
z_2^1 z_2^0
\end{array} \right\}
$$

\[\sigma(v_0) = 00, \quad \sigma(v_1) = 01, \quad \sigma(v_2) = 10, \quad \sigma(v_3) = 11\]
Ex 2e) Packets must traverse $v_2$ and eventually reach $t$.

$$\psi_\phi(Z) = \psi_t(Z) \cdot \psi_{v_2}(Z)$$

$$= \bar{z}_0^1 z_0^0 \bar{z}_1^1 z_1^0 \bar{z}_2^1 z_2^0 + z_0^1 \bar{z}_0^0$$

$$Z = \begin{pmatrix} z_0^1 z_0^0 \\ z_1^1 z_1^0 \\ z_2^1 z_2^0 \end{pmatrix}$$

$$= \begin{pmatrix} \rho(v_0) \\ \rho(v_1) \\ \rho(v_2) \end{pmatrix}$$

$$\sigma(v_0) = 00$$
$$\sigma(v_2) = 10$$
$$\sigma(v_1) = 01$$
$$\sigma(v_3) = 11$$
Ex 2e) Packets must traverse \( v_2 \) and eventually reach \( t \).

\[
\psi_\phi(Z) = \psi_t(Z) \cdot \psi_{v_2}(Z) \\
= \overline{z}_0 z_0 \overline{z}_1 z_1 \overline{z}_2 z_2 \\
+ z_0 \overline{z}_0 \left( z_1 \overline{z}_1 z_2 \overline{z}_2 \right)
\]

\[
Z = \begin{cases}
    z_0^1 \overline{z}_0^0 & \rho(v_1) \\
    z_1^1 \overline{z}_1^0 & \rho(v_0) \\
    z_2^1 \overline{z}_2^0 & \rho(v_2)
\end{cases}
\]

\[
\sigma(v_0) = 00 \\
\sigma(v_1) = 01 \\
\sigma(v_2) = 10 \\
\sigma(v_3) = 11
\]
Ex 2e) Packets must traverse $v_2$ and eventually reach $t$.

$$
\psi(\mathbf{Z}) = \psi_t(\mathbf{Z}) \cdot \psi_{v_2}(\mathbf{Z})
= \bar{z}^1 z_0 \bar{z}^0 z^1_0 z_0 z^0_1 z^0_2
+ z_0^1 \bar{z}^0_0 \left( z^1_2 z_2 + \bar{z}^1_2 z_2 z^1_1 z_1 \right)
$$
Ex 2f) Verify the initial and final state.

\[ Z = \begin{cases} 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{cases} \]

\[ \rho(v_0) \quad \rho(v_1) \quad \rho(v_2) \]

\[ \sigma(v_0) = 00 \quad \sigma(v_1) = 01 \quad \sigma(v_2) = 10 \quad \sigma(v_3) = 11 \]
Ex 2f) Verify the initial and final state.

- Initial state $\rho_0$
  - $\sigma(\rho_0) = 01\ 10\ 11$
  - $\psi_\phi(\psi_{\rho_0}(Z)) = true$

\[
Z = \begin{bmatrix}
  z_0^1 & z_0^0 \\
  z_1^1 & z_1^0 \\
  z_2^1 & z_2^0
\end{bmatrix}
\]

- Final state $\rho_f$
  - $\sigma(v_1) = 01$
  - $\sigma(v_0) = 00$
  - $\sigma(v_2) = 10$
  - $\sigma(v_3) = 11$
Ex 2f) Verify the initial and final state.

- Initial state $\rho_0$
  $\sigma(\rho_0) = 01\ 10\ 11$
  $\psi_\phi(\psi_{\rho_0}(Z)) = true$

- Final state $\rho_f$
  $\sigma(\rho_f) = 10\ 01\ 11$
  $\psi_\phi(\psi_{\rho_f}(Z)) = true$

\[
Z = \begin{pmatrix}
  \rho(v_0) & z_1^1\ z_0^0 \\
  z_0^1\ z_0^0 & \rho(v_1) \\
  z_2^1\ z_2^0 & \rho(v_2)
\end{pmatrix}
\]

$\sigma(v_1) = 01$

$\sigma(v_0) = 00$

$\sigma(v_2) = 10$

$\sigma(v_3) = 11$
Ex 2g) Describing State Transitions.

**Requirement**: We can change only one node at a time!
Ex 2g) Describing State Transitions.

**Requirement:** We can change only one node at a time!

\[ \psi_{trans}(Z, Z') = \exists i \in \{0, 1, 2\} : \forall k \in \{0, 1, 2\} : \begin{cases} 
  z_k = z'_k & \text{if } k \neq i \\
  z_k \neq z'_k & \text{if } k = i 
\end{cases} \]
Ex 2g) Describing State Transitions.

**Requirement:** We can change only one node at a time!

\[
\psi_{\text{trans}}(Z, Z') = \exists i \in \{0, 1, 2\} : \forall k \in \{0, 1, 2\} : \left\{ \begin{array}{ll}
z_k = z'_k & \text{if } k \neq i \\
z_k \neq z'_k & \text{if } k = i
\end{array} \right.
\]

\[
= \left[ (z_0 \neq z'_0) \cdot (z_1 = z'_1) \cdot (z_2 = z'_2) \right] + \\
\left[ (z_0 = z'_0) \cdot (z_1 \neq z'_1) \cdot (z_2 = z'_2) \right] + \\
\left[ (z_0 = z'_0) \cdot (z_1 = z'_1) \cdot (z_2 \neq z'_2) \right]
\]
Ex 2h) Find a safe migration with a Model Checker.

(1) Build the state machine with $2^6 = 64$ states and the transitions as described in $\psi_{trans}$.
Ex 2h) Find a safe migration with a Model Checker.

(1) Build the state machine with $2^6 = 64$ states and the transitions as described in $\psi_{trans}$.

(2) Remove all transitions from the final state, but add a transition to itself for the transition function to be fully defined.
Ex 2h) Find a safe migration with a Model Checker.

(1) Build the state machine with $2^6 = 64$ states and the transitions as described in $\psi_{trans}$.
(2) Remove all transitions from the final state, but add a transition to itself for the transition function to be **fully defined**.
(3) Describe the migration in CTL ($\phi'$: states that satisfy $\psi_{trans}$ and $\psi_{\phi}$, $\phi_f$: the final state):

$$\text{EG} (\phi' \land \text{EF} \phi_f)$$
Ex 2h) Find a safe migration with a Model Checker.

(1) Build the state machine with $2^6 = 64$ states and the transitions as described in $\psi_{trans}$.

(2) Remove all transitions from the final state, but add a transition to itself for the transition function to be fully defined.

(3) Describe the migration in CTL ($\phi'$: states that satisfy $\psi_{trans}$ and $\psi_\phi$, $\phi_f$: the final state):

$$\text{EG}(\phi' \land \text{EF } \phi_f)$$

(4) Invert the CTL to find a counter-example.

$$\neg\text{EG}(\phi' \land \text{EF } \phi_f) = \text{AF } \neg(\phi' \land \text{EF } \phi_f)$$

$$= \text{AF}(\neg\phi' \lor \neg\text{EF } \phi_f)$$

$$= \text{AF}(\neg\phi' \lor \text{AG } \neg\phi_f)$$
Ex 2i) Find all valid migrations using BDD.

- 3 Transitions $\Rightarrow$ 4 States $Z_0$, $Z_1$, $Z_2$ and $Z_3$. 

\[
\begin{align*}
\psi_{\text{topo}}(p & \to Z_1 q) \\
\psi_{\text{topo}}(p & \to Z_2 q)
\end{align*}
\]
Ex 2i) Find all valid migrations using BDD.

- 3 Transitions $\implies$ 4 States $Z_0$, $Z_1$, $Z_2$ and $Z_3$.
- Characteristic function for the initial state: $\psi_0(Z) = \overline{z}_0^1 z_1^0 \overline{z}_1^1 z_2^0 z_1^1 z_2^0$
- Characteristic function for the final state: $\psi_f(Z) = z_0^1 \overline{z}_0^0 \overline{z}_1^1 z_1^0 z_2^0 z_1^1 z_2^0$
Ex 2i) Find all valid migrations using BDD.

- 3 Transitions $\Rightarrow$ 4 States $Z_0$, $Z_1$, $Z_2$ and $Z_3$.
- Characteristic function for the initial state: $\psi_0(Z) = z_0^1z_0^0 z_1^1z_1^0 z_2^1z_2^0$
- Characteristic function for the final state: $\psi_f(Z) = z_0^1z_0^0 z_1^0z_1^0 z_2^1z_2^0$
- The complete equation:

$$\psi^* = \psi_0(Z_0) \cdot \psi_f(Z_3)$$

$$\cdot \psi_{\text{trans}}(Z_0, Z_1)$$

$$\cdot \psi_{\text{trans}}(Z_1, Z_2)$$

$$\cdot \psi_{\text{trans}}(Z_2, Z_3)$$

$$\cdot \psi_{\text{topo}}(Z_1) \cdot \psi_{\phi}(Z_1)$$

$$\cdot \psi_{\text{topo}}(Z_2) \cdot \psi_{\phi}(Z_2)$$
Ex 2i) Find all valid migrations using BDD.