CrashCourse — Verification of Finite Automata

CTL Model-Checking

Tibor Schneider

TIK

Formulation of CTL properties

Based on atomic propositions (ϕ) and quantifiers

$A\phi$	$\rightarrow$ «**A**ll ϕ»,	ϕ holds on all paths	Quantifiers over paths
$E\phi$	$\rightarrow$ «**E**xists ϕ»,	ϕ holds on at least one path	
$X\phi$	$\rightarrow$ «Ne**X**t ϕ»,	ϕ holds on the next state	Path-specific quantifiers
$F\phi$	$\rightarrow$ «**F**inally ϕ»,	ϕ holds at some state along the path	
$G\phi$	$\rightarrow$ «**G**lobally ϕ»,	ϕ holds on all states along the path	
$\phi_1 U \phi_2$	$\rightarrow$ «ϕ_1 **U**ntil ϕ_2»,	ϕ_1 holds until ϕ_2 holds	

Formulation of CTL properties

Proper CTL formula: {A,E} {X,F,G,U}ϕ

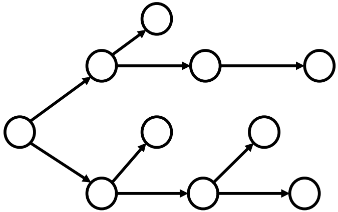
→ Quantifiers **go by pairs**, you need one of each.

Simple "means" that we get rid of leaf nodes…
→ They transition to themselves

Missing Hypothesis

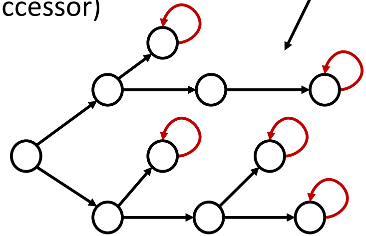
Interpretation on CTL formula

→ Transition functions are **fully defined** (i.e. every state has at least one successor)



Automaton of interest

→



Automaton to work with

Inverting properties is sometimes useful!

$$\mathsf{AG}\,\phi := \neg\mathsf{EF}\,\neg\phi$$

$$\mathsf{AF}\,\phi := \neg\mathsf{EG}\,\neg\phi$$

$$\mathsf{EF}\,\phi := \neg\mathsf{AG}\,\neg\phi$$

$$\mathsf{EG}\,\phi := \neg\mathsf{AF}\,\neg\phi$$

Inverting properties is sometimes useful!

"On all paths, for all states, ϕ holds" $\iff$ "There exists no path along which at some state ϕ doesn't hold."

$$\mathsf{AG}\,\phi := \neg\mathsf{EF}\,\neg\phi$$

$$\mathsf{AF}\,\phi := \neg\mathsf{EG}\,\neg\phi$$

$$\mathsf{EF}\,\phi := \neg\mathsf{AG}\,\neg\phi$$

$$\mathsf{EG}\,\phi := \neg\mathsf{AF}\,\neg\phi$$

Inverting properties is sometimes useful!

$AG\,\phi := \neg EF\,\neg\phi$

$AF\,\phi := \neg EG\,\neg\phi$

$EF\,\phi := \neg AG\,\neg\phi$

$EG\,\phi := \neg AF\,\neg\phi$

"On all paths, for all states, ϕ holds" $\iff$ "There exists no path along which at some state ϕ doesn't hold."

"On all paths, there exists a state where ϕ holds" $\iff$ "There exists no path along which ϕ doesn't hold for all states."

...

Inverting properties is sometimes useful!

$\mathrm{AG}\,\phi := \neg\mathrm{EF}\,\neg\phi$ → "On all paths, for all states, ϕ holds" $\iff$ "There exists no path along which at some state ϕ doesn't hold."

$\mathrm{AF}\,\phi := \neg\mathrm{EG}\,\neg\phi$ → "On all paths, there exists a state where ϕ holds" $\iff$ "There exists no path along which ϕ doesn't hold for all states."

$\mathrm{EF}\,\phi := \neg\mathrm{AG}\,\neg\phi$ → ...

$\mathrm{EG}\,\phi := \neg\mathrm{AF}\,\neg\phi$

Remark: There exists other temporal logics.

- LTL (Linear Temporal Logic)
- $\mathrm{CTL}^* = \{\mathrm{CTL}, \mathrm{LTL}\}$
- ...

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, ϕ
  - CTL,
  - LTL, …

Output

- $\mathbf{M} \models \phi$?
- A trace for which the property does not hold!

Your turn to work!

Ex 1a) Temporal Logic

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

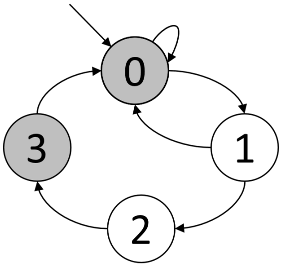
(ii) $\mathsf{EG}\, a : Q = \{0, 3\}$

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

$$\mathsf{AX}\, a : Q_1 = \{2, 3\}$$
$$\mathsf{EX}\,\mathsf{AX}\, a : Q_2 = \{1, 2\}$$

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

$$\neg a : Q_1 = \{1, 2\}$$
$$\mathsf{EX}\, \neg a : Q_2 = \{0, 1\}$$
$$a \wedge \mathsf{EX}\, \neg a : Q_3 = \{0\}$$
$$\mathsf{EF}(a \wedge \mathsf{EX}\, \neg a) : Q_4 = \{0, 1, 2, 3\}$$



Ex 1b) Temporal Logic

(i) $\neg \mathsf{AF}\, Z = \mathsf{EG}\, \neg Z \quad \Longrightarrow \quad \mathsf{AF}\, Z = \neg \mathsf{EG}\, \neg Z$

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

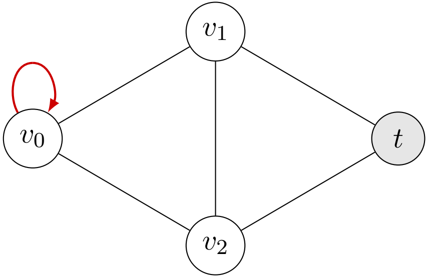
$$
\begin{aligned}
Q_0 &= S \backslash Z \\
Q_{i+1} &= Q_i \cap pred(Q_i, f) \\
k &= \min\{i \mid Q_{i+1} = Q_i\} \\
Q_{\mathsf{AF}\ Z} &= Z \backslash Q_k
\end{aligned}
$$

Require: ψ_Z, ψ_f

$\psi_{cur} \leftarrow \neg \psi_Z$
$\psi_{next} \leftarrow \psi_{cur} \wedge \psi_{pred(\psi_{cur}, f)}$
while $\psi_{cur} \neq \psi_{next}$ **do**
    $\psi_{cur} \leftarrow \psi_{next}$
    $\psi_{next} \leftarrow \psi_{cur} \wedge \psi_{pred(\psi_{cur}, f)}$
end while
return $\psi_{\mathsf{AF}\, Z} = \neg \psi_{cur}$

Ex 2a) Find all possible loops.

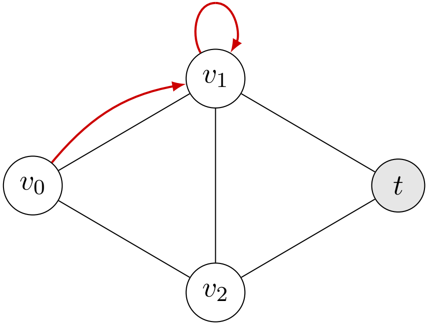
(1) $\rho_1(v_0) = v_0$



Ex 2a) Find all possible loops.

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

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

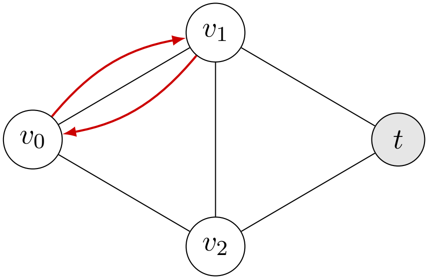


Ex 2a) Find all possible loops.

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

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

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



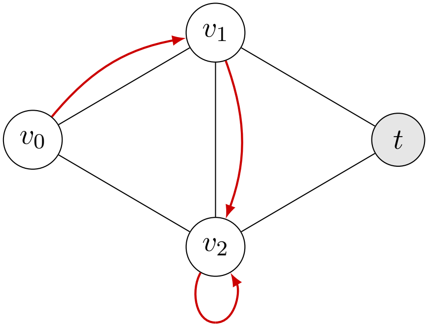
Ex 2a) Find all possible loops.

(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.

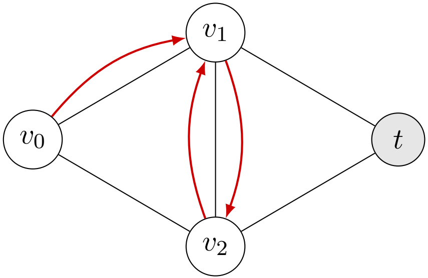
(1) $\rho_1(v_0) = v_0$

(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) $\rho_1(v_0) = v_0$

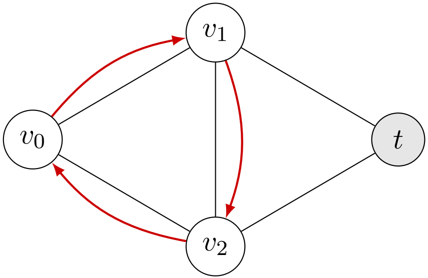
(2) $\rho_2(v_0) = v_1,\ \rho_2(v_1) = v_1$

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

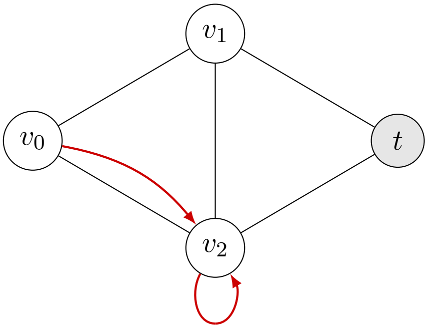
(3) $\rho_3(v_0) = v_1$, $\rho_3(v_1) = v_0$

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

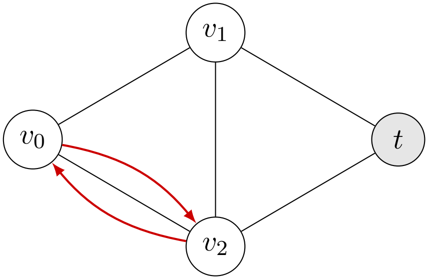
(4) $\rho_4(v_0) = v_1,\ \rho_4(v_1) = v_2,\ \rho_4(v_2) = v_2$

(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) $\rho_4(v_0) = v_1,\ \rho_4(v_1) = v_2,\ \rho_4(v_2) = v_2$

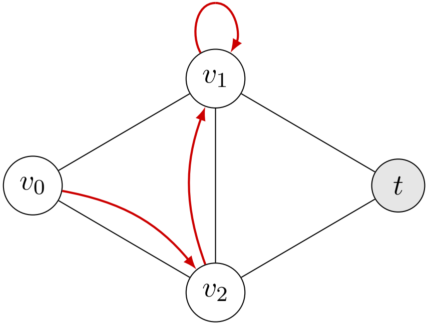
(5) $\rho_5(v_0) = v_1,\ \rho_5(v_1) = v_2,\ \rho_5(v_2) = v_1$

(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) $\rho_5(v_0) = v_1,\ \rho_5(v_1) = v_2,\ \rho_5(v_2) = v_1$

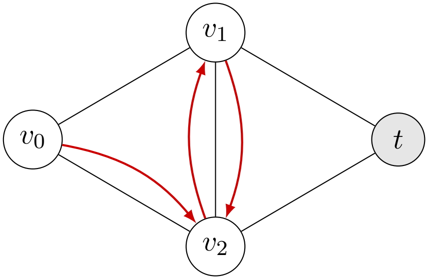
(6) $\rho_6(v_0) = v_1,\ \rho_6(v_1) = v_2,\ \rho_6(v_2) = v_0$

(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) $\rho_6(v_0) = v_1,\ \rho_6(v_1) = v_2,\ \rho_6(v_2) = v_0$

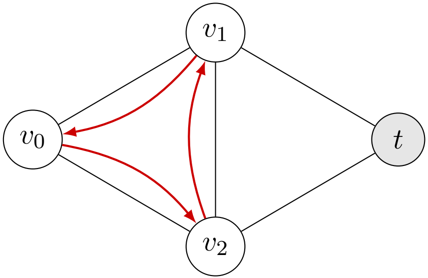
(7) $\rho_7(v_0) = v_2,\ \rho_7(v_2) = v_2$

(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) $\rho_{11}(v_0) = v_2,\ \rho_{11}(v_2) = v_1,\ \rho_{11}(v_1) = v_0$



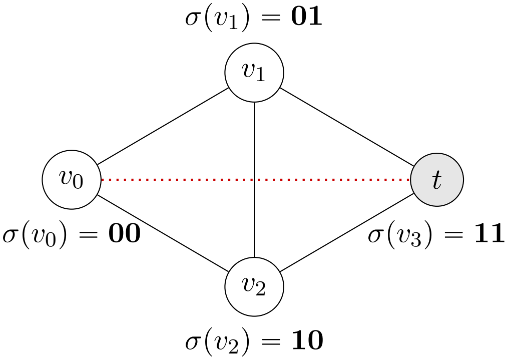
Ex 2b) Encode the physical topology

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

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

$$\rho(v_0) \neq t$$

$$\psi_{topo}(\mathbf{Z}) = \neg\left(z_0^1 z_0^0\right)$$

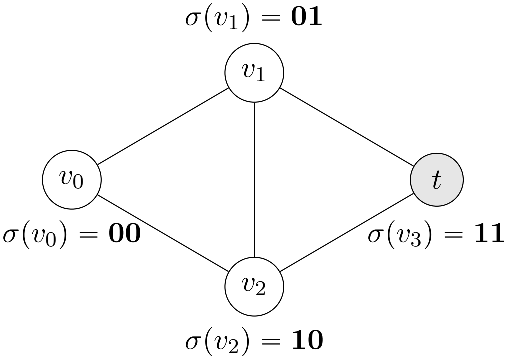


Ex 2c) Packets eventually reach t.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

Enumerate all possible paths to reach t.

$\psi_t(\mathbf{Z}) =$

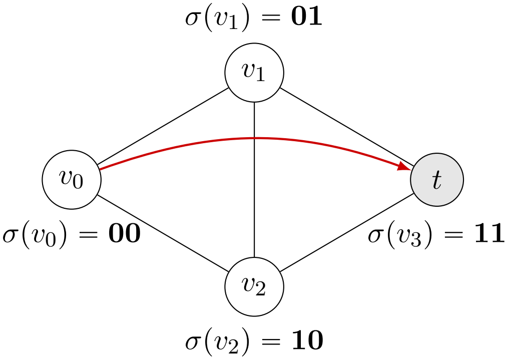


Ex 2c) Packets eventually reach t.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

Enumerate all possible paths to reach t.

$$\psi_t(\mathbf{Z}) = z_0^1 z_0^0$$

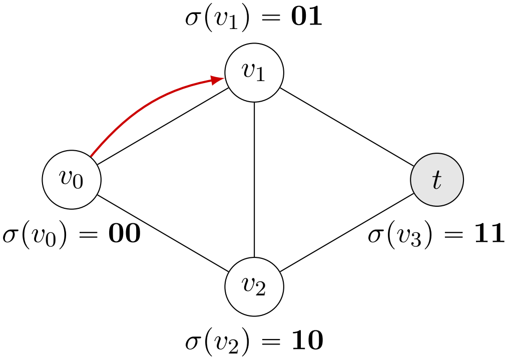


Ex 2c) Packets eventually reach t.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

Enumerate all possible paths to reach t.

$$\psi_t(\mathbf{Z}) = z_0^1 z_0^0 \\ + \bar{z}_0^1 z_0^0$$

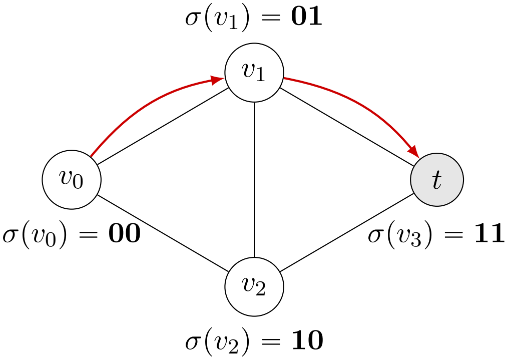


Ex 2c) Packets eventually reach t.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

Enumerate all possible paths to reach t.

$$\psi_t(\mathbf{Z}) = z_0^1 z_0^0$$
$$+ \bar{z}_0^1 z_0^0 \Big(z_1^1 z_1^0$$

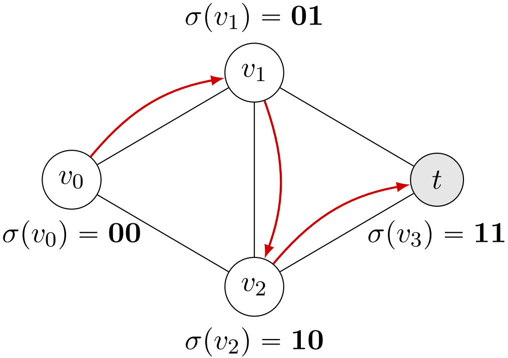


Ex 2c) Packets eventually reach t.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

Enumerate all possible paths to reach t.

$$\begin{aligned}\psi_t(\mathbf{Z}) = {} & z_0^1 z_0^0 \\ & + \overline{z_0^1} z_0^0 \left(z_1^1 z_1^0 + z_1^1 \overline{z_1^0} \; z_2^1 z_2^0 \right)\end{aligned}$$

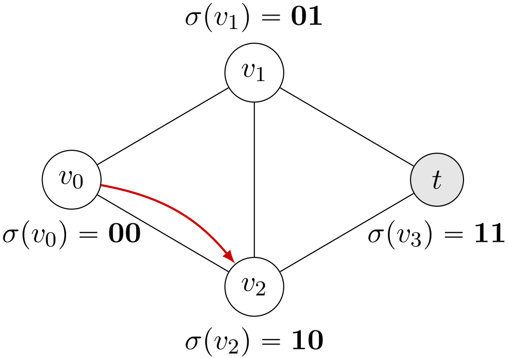


Ex 2c) Packets eventually reach t.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

Enumerate all possible paths to reach t.

$$\begin{aligned}\psi_t(\mathbf{Z}) = \; & z_0^1 z_0^0 \\ & + \overline{z_0^1} z_0^0 \left(z_1^1 z_1^0 + z_1^1 \overline{z_1^0} \; z_2^1 z_2^0 \right) \\ & + z_0^1 \overline{z_0^0}\end{aligned}$$

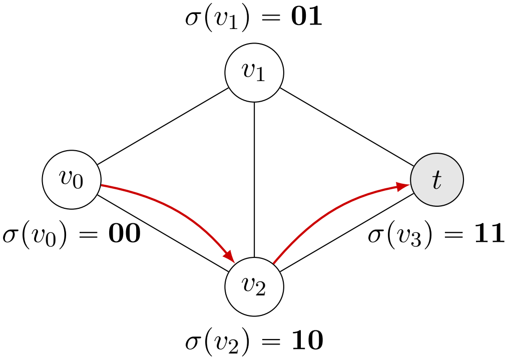


Ex 2c) Packets eventually reach t.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

Enumerate all possible paths to reach t.

$$\begin{aligned} \psi_t(\mathbf{Z}) = \; & z_0^1 z_0^0 \\ & + \bar{z}_0^1 z_0^0 \left(z_1^1 z_1^0 + z_1^1 \bar{z}_1^0 \; z_2^1 z_2^0 \right) \\ & + z_0^1 \bar{z}_0^0 \left(z_2^1 z_2^0 \right. \end{aligned}$$

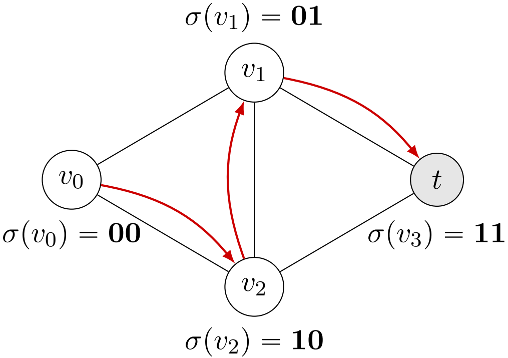


Ex 2c) Packets eventually reach t.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

Enumerate all possible paths to reach t.

$$\begin{aligned}\psi_t(\mathbf{Z}) = \; & z_0^1 z_0^0 \\ & + \bar{z}_0^1 z_0^0 \left(z_1^1 z_1^0 + z_1^1 \bar{z}_1^0 \; z_2^1 z_2^0 \right) \\ & + z_0^1 \bar{z}_0^0 \left(z_2^1 z_2^0 + \bar{z}_2^1 z_2^0 \; z_1^1 z_1^0 \right)\end{aligned}$$

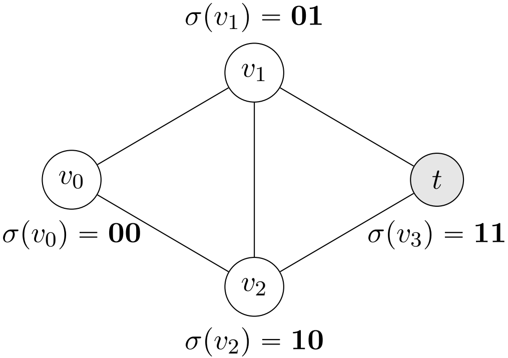


Ex 2d) Packets must traverse v_2.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

Enumerate all possible paths to reach v_0*.*

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

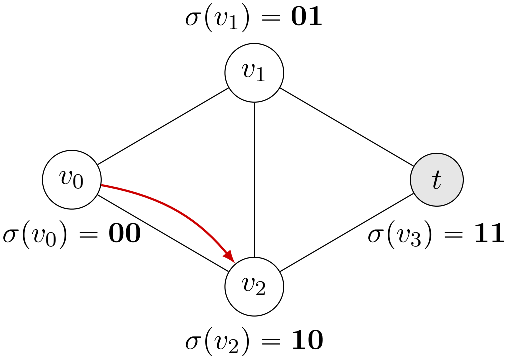


Ex 2d) Packets must traverse v_2.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

Enumerate all possible paths to reach v_0.

$$\psi_{v_2}(\mathbf{Z}) = z_0^1 \overline{z_0^0} +$$

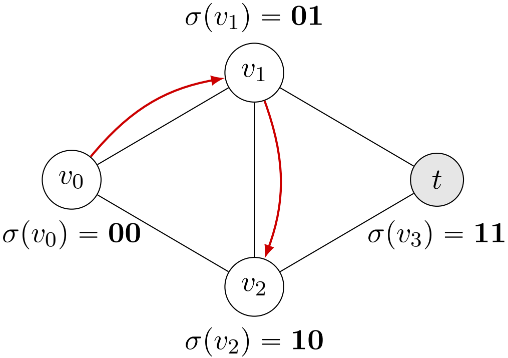


Ex 2d) Packets must traverse v_2.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

Enumerate all possible paths to reach v_0.

$$\psi_{v_2}(\mathbf{Z}) = z_0^1 \overline{z_0^0} + \overline{z_0^1} z_0^0 \; z_1^1 \overline{z_1^0}$$

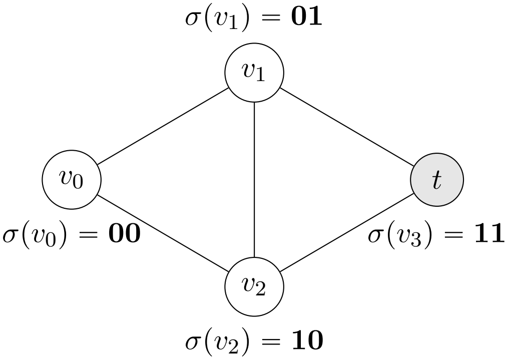


Ex 2e) Packets must traverse v_2 and eventually reach t.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

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

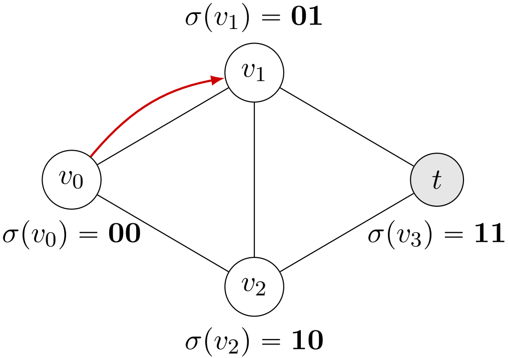
$$=$$



Ex 2e) Packets must traverse v_2 and eventually reach t.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

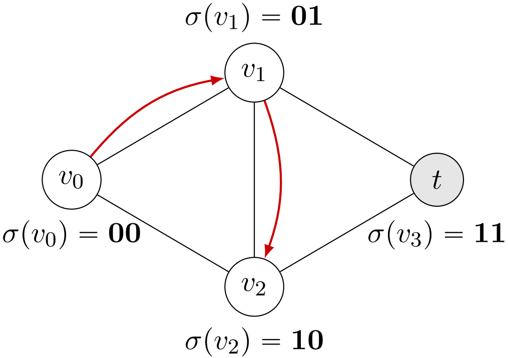
$$\begin{aligned} \psi_\phi(\mathbf{Z}) &= \psi_t(\mathbf{Z}) \cdot \psi_{v_2}(\mathbf{Z}) \\ &= \bar{z}_0^1 z_0^0 \end{aligned}$$



Ex 2e) Packets must traverse v_2 and eventually reach t.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

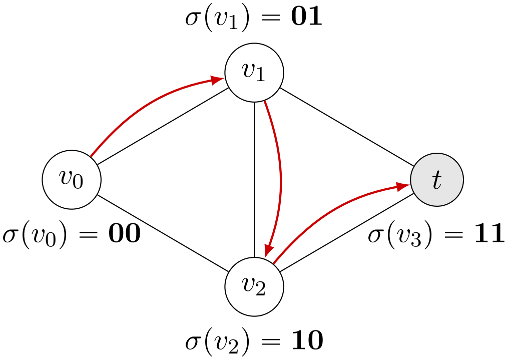
$$\begin{aligned} \psi_\phi(\mathbf{Z}) &= \psi_t(\mathbf{Z}) \cdot \psi_{v_2}(\mathbf{Z}) \\ &= \overline{z_0^1} z_0^0 \; z_1^1 \overline{z_1^0} \end{aligned}$$



Ex 2e) Packets must traverse v_2 and eventually reach t.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

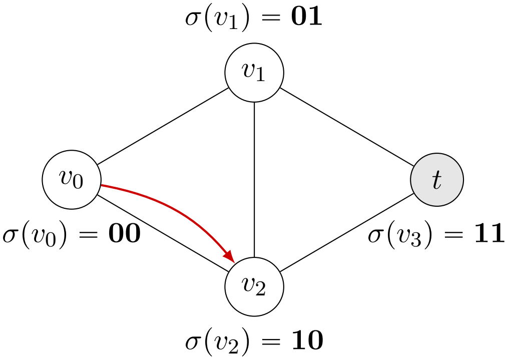
$$\begin{aligned} \psi_\phi(\mathbf{Z}) &= \psi_t(\mathbf{Z}) \cdot \psi_{v_2}(\mathbf{Z}) \\ &= \overline{z_0^1} z_0^0 \; z_1^1 \overline{z_1^0} \; z_2^1 z_2^0 \end{aligned}$$



Ex 2e) Packets must traverse v_2 and eventually reach t.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \; \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \; \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

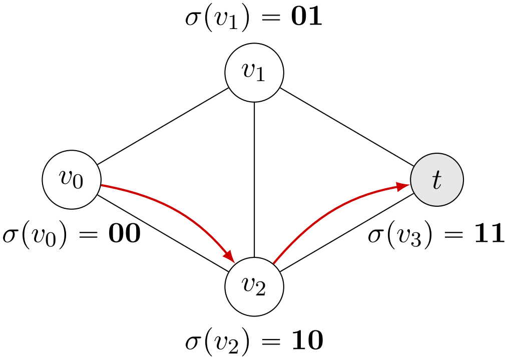
$$\begin{aligned}\psi_\phi(\mathbf{Z}) &= \psi_t(\mathbf{Z}) \cdot \psi_{v_2}(\mathbf{Z}) \\ &= \bar{z}_0^1 z_0^0 \; z_1^1 \bar{z}_1^0 \; z_2^1 z_2^0 \\ &\quad + z_0^1 \bar{z}_0^0\end{aligned}$$



Ex 2e) Packets must traverse v_2 and eventually reach t.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

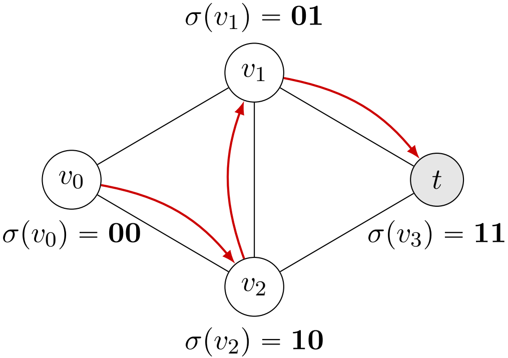
$$\begin{aligned}\psi_\phi(\mathbf{Z}) &= \psi_t(\mathbf{Z}) \cdot \psi_{v_2}(\mathbf{Z}) \\ &= \bar{z}_0^1 z_0^0 \; z_1^1 \bar{z}_1^0 \; z_2^1 z_2^0 \\ &\quad + z_0^1 \bar{z}_0^0 \Big(z_2^1 z_2^0\end{aligned}$$



Ex 2e) Packets must traverse v_2 and eventually reach t.

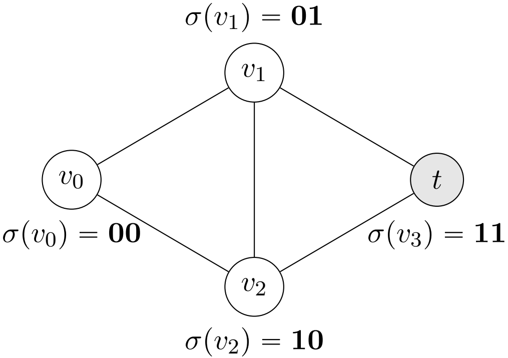
$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

$$\begin{aligned}
\psi_\phi(\mathbf{Z}) &= \psi_t(\mathbf{Z}) \cdot \psi_{v_2}(\mathbf{Z}) \\
&= \overline{z_0^1} z_0^0 \; z_1^1 \overline{z_1^0} \; z_2^1 z_2^0 \\
&\quad + z_0^1 \overline{z_0^0} \left(z_2^1 z_2^0 + \overline{z_2^1} z_2^0 \; z_1^1 z_1^0 \right)
\end{aligned}$$



Ex 2f) Verify the initial and final state.

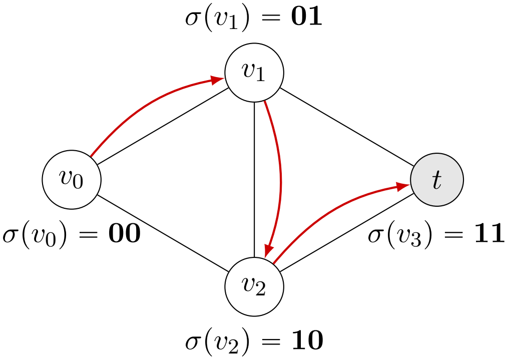
$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$



Ex 2f) Verify the initial and final state.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

- Initial state ρ_0
  $\sigma(\rho_0) = 01\,10\,11$
  $\psi_\phi(\psi_{\rho_0}(\mathbf{Z})) = true$

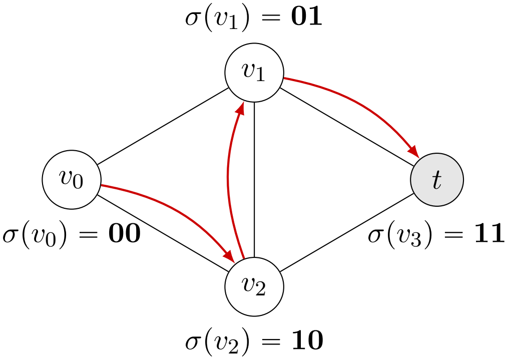


Ex 2f) Verify the initial and final state.

$$\mathbf{Z} = \underbrace{z_0^1 z_0^0}_{\rho(v_0)} \overbrace{z_1^1 z_1^0}^{\rho(v_1)} \underbrace{z_2^1 z_2^0}_{\rho(v_2)}$$

- Initial state ρ_0
  $\sigma(\rho_0) = 01\,10\,11$
  $\psi_\phi(\psi_{\rho_0}(\mathbf{Z})) = \mathit{true}$

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



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!*

$$\psi_{trans}(\mathbf{Z}, \mathbf{Z}') = \exists i \in \{0,1,2\} : \forall k \in \{0,1,2\} : \begin{cases} \mathbf{z}_k = \mathbf{z}'_k & \text{if } k \neq i \\ \mathbf{z}_k \neq \mathbf{z}'_k & \text{if } k = i \end{cases}$$

Ex 2g) Describing State Transitions.

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

$$\psi_{trans}(\mathbf{Z}, \mathbf{Z}') = \exists i \in \{0,1,2\} : \forall k \in \{0,1,2\} : \begin{cases} \mathbf{z}_k = \mathbf{z}'_k & \text{if } k \neq i \\ \mathbf{z}_k \neq \mathbf{z}'_k & \text{if } k = i \end{cases}$$

$$\begin{aligned} = & \left[(\mathbf{z}_0 \neq \mathbf{z}'_0) \cdot (\mathbf{z}_1 = \mathbf{z}'_1) \cdot (\mathbf{z}_2 = \mathbf{z}'_2)\right] + \\ & \left[(\mathbf{z}_0 = \mathbf{z}'_0) \cdot (\mathbf{z}_1 \neq \mathbf{z}'_1) \cdot (\mathbf{z}_2 = \mathbf{z}'_2)\right] + \\ & \left[(\mathbf{z}_0 = \mathbf{z}'_0) \cdot (\mathbf{z}_1 = \mathbf{z}'_1) \cdot (\mathbf{z}_2 \neq \mathbf{z}'_2)\right] \end{aligned}$$

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 ψ_{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 ψ_{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 ψ_{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 (ϕ': states that satisfy ψ_{trans} and ψ_{ϕ}, ϕ_f: the final state):

$$\mathsf{EG}\,(\phi' \wedge \mathsf{EF}\ \phi_f)$$

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 ψ_{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 (ϕ': states that satisfy ψ_{trans} and ψ_ϕ, ϕ_f: the final state):

$$\mathsf{EG}\,(\phi' \wedge \mathsf{EF}\ \phi_f)$$

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

$$\begin{aligned}\neg\mathsf{EG}\big(\phi' \wedge \mathsf{EF}\,\phi_f\big) &= \mathsf{AF}\,\neg\big(\phi' \wedge \mathsf{EF}\,\phi_f\big)\\ &= \mathsf{AF}\big(\neg\phi' \vee \neg\mathsf{EF}\,\phi_f\big)\\ &= \mathsf{AF}\big(\neg\phi' \vee \mathsf{AG}\,\neg\phi_f\big)\end{aligned}$$

Ex 2i) Find all valid migrations using BDD.

- 3 Transitions $\Longrightarrow$ 4 States $\mathbf{Z}_0$, $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$.

Ex 2i) Find all valid migrations using BDD.

- 3 Transitions $\Longrightarrow$ 4 States $\mathbf{Z}_0$, $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$.
- Characteristic function for the initial state: $\psi_0(\mathbf{Z}) = \overline{z_0^1} z_0^0 \; z_1^1 \overline{z_1^0} \; z_2^1 z_2^0$
- Characteristic function for the final state: $\psi_f(\mathbf{Z}) = z_0^1 \overline{z_0^0} \; \overline{z_1^1} z_1^0 \; z_2^1 z_2^0$

Ex 2i) Find all valid migrations using BDD.

- 3 Transitions $\Longrightarrow$ 4 States $\mathbf{Z}_0$, $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$.
- Characteristic function for the initial state: $\psi_0(\mathbf{Z}) = \overline{z}_0^1 z_0^0 \; z_1^1 \overline{z}_1^0 \; z_2^1 z_2^0$
- Characteristic function for the final state: $\psi_f(\mathbf{Z}) = z_0^1 \overline{z}_0^0 \; \overline{z}_1^1 z_1^0 \; z_2^1 z_2^0$
- The complete equation:

$$
\begin{aligned}
\psi^* = \psi_0(\mathbf{Z_0}) \; & \cdot \psi_f(\mathbf{Z}_3) \\
& \cdot \psi_{trans}(\mathbf{Z}_0, \mathbf{Z}_1) \\
& \cdot \psi_{trans}(\mathbf{Z}_1, \mathbf{Z}_2) \\
& \cdot \psi_{trans}(\mathbf{Z}_2, \mathbf{Z}_3) \\
& \cdot \psi_{topo}(\mathbf{Z}_1) \cdot \psi_{\phi}(\mathbf{Z}_1) \\
& \cdot \psi_{topo}(\mathbf{Z}_2) \cdot \psi_{\phi}(\mathbf{Z}_2)
\end{aligned}
$$

Ex 2i) Find all valid migrations using BDD.