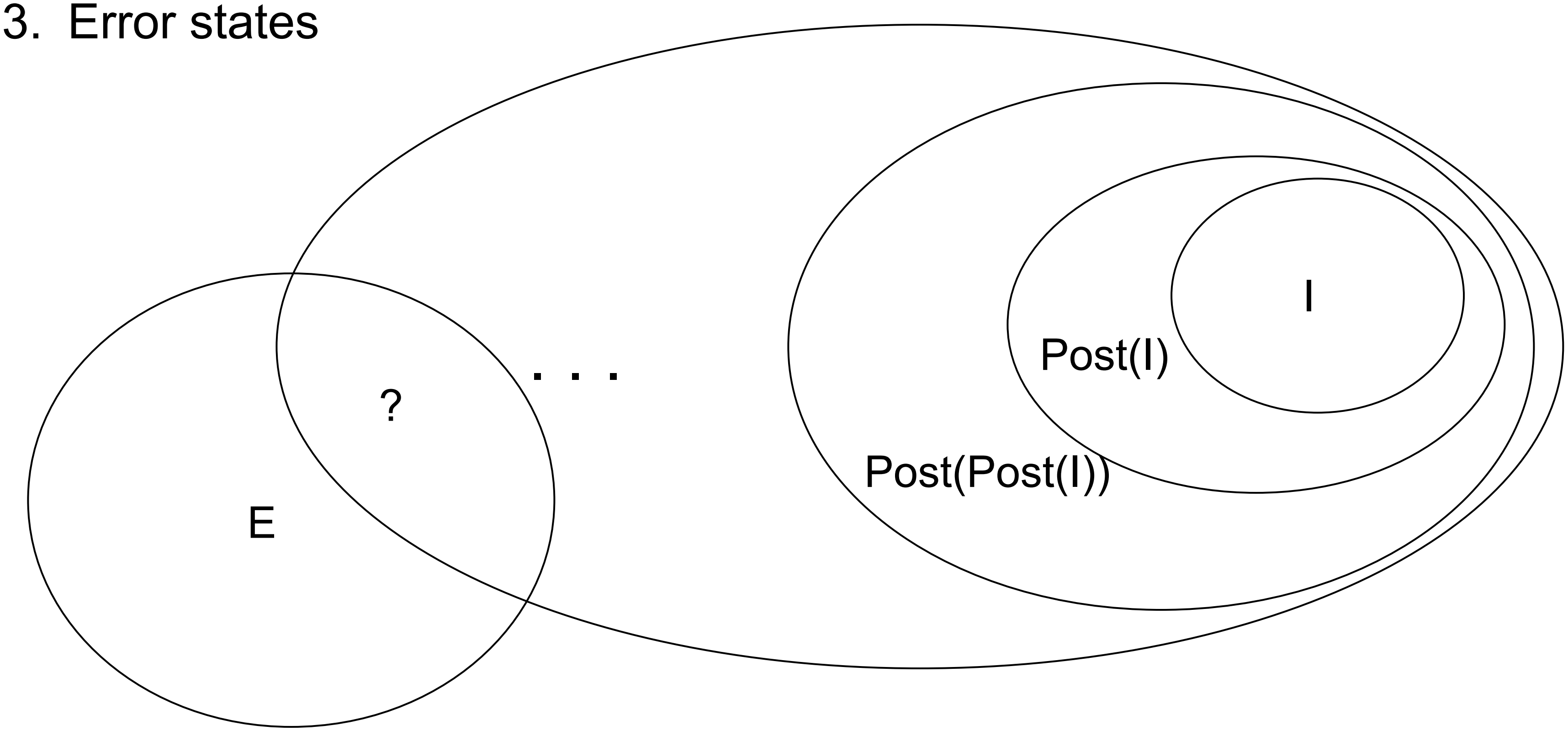
Widening Automata

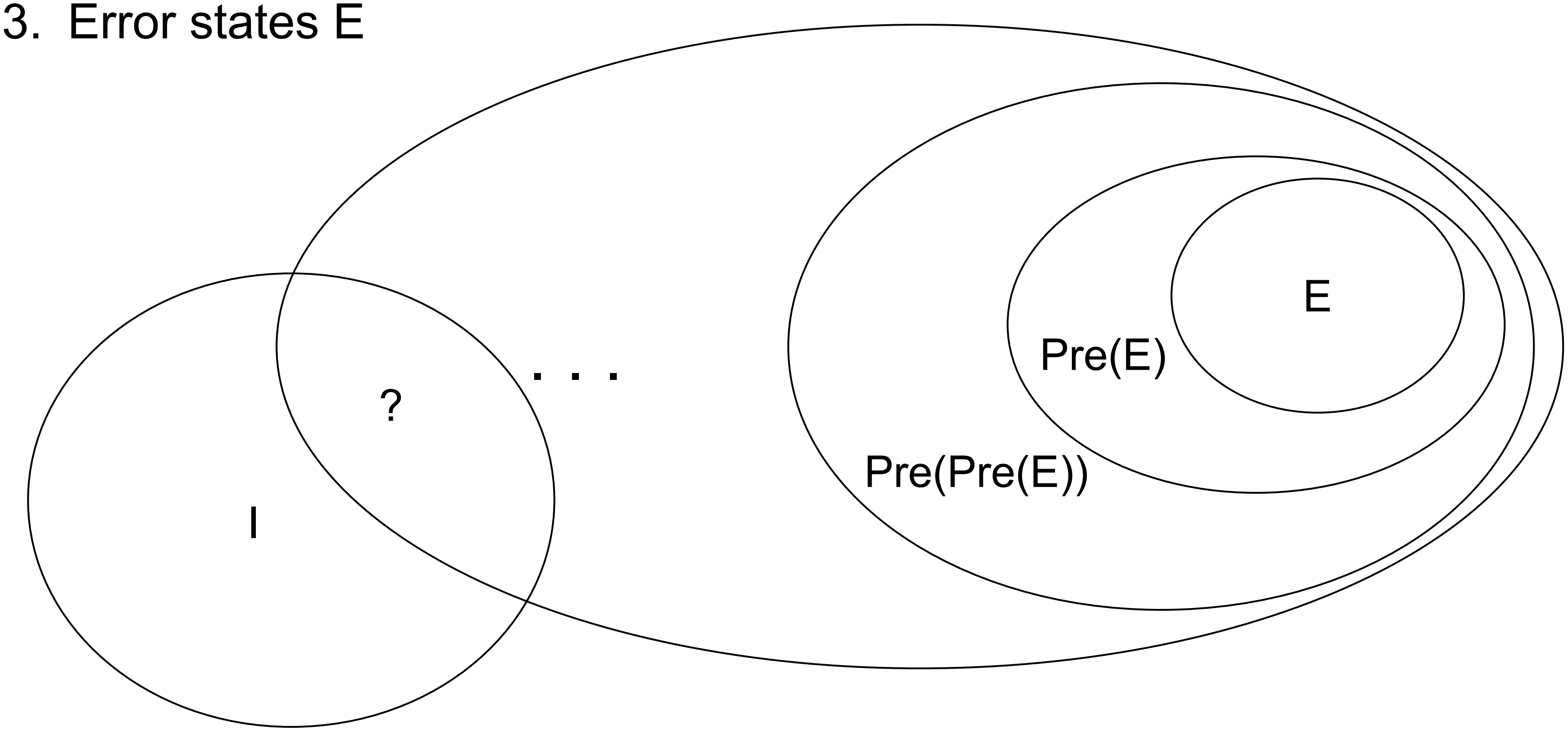
Verification of assertions
Forward fixpoint computation

1. Set of initial states I
2. Postcondition function Post()
3. Error states



Verification of assertions
Backward fixpoint computation

1. Set of initial states I
2. Precondition function Pre()
3. Error states E



Fixpoints may not converge

- For infinite state systems fixpoint computations may
  - not converge at all
  - require a large number of iterations
- Widening is a approximation technique that helps a fixpoint computation converge

A widening operator

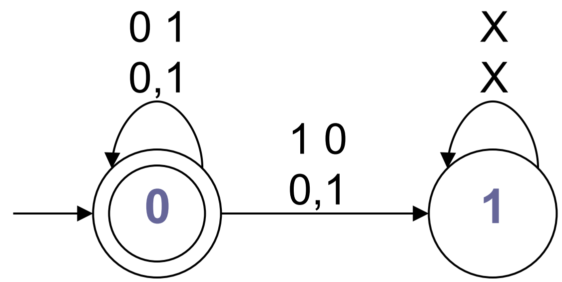
- Idea: Instead of computing a sequence of automata $A_1, A_2, \ldots$ where $A_{i+1}=A_i \cup post(A_i)$, compute $A'_1, A'_2, \ldots$ where
  $A'_{i+1}=A'_i \nabla (A'_i \cup post(A'_i))$
- By definition $A \cup B \subseteq A \nabla B$
- The goal is to find a widening operator ∇ such that:
  1. The sequence $A'_1, A'_2, \ldots$ converges
  2. It converges fast
  3. The computed fixpoint is as close as possible to the exact set of reachable states

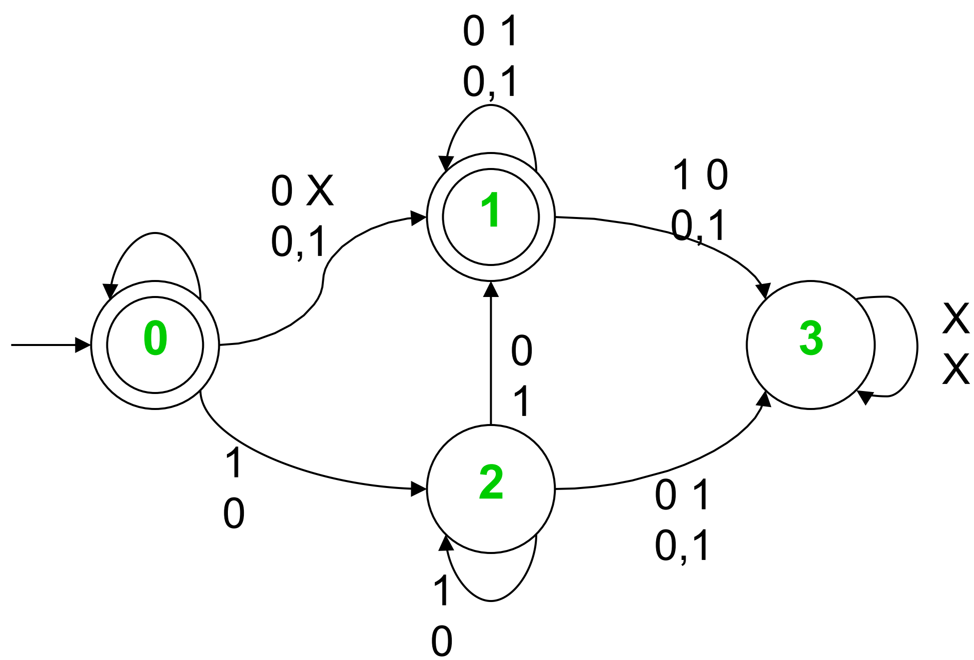
Widening Automata

- Given automata A and A' we want to compute $A \nabla A'$
- We say that states k and k' are equivalent ($k \equiv k'$) if either
  - k and k' can be reached from either initial state with the same string (unless k or k' is a sink state)
  - or, the languages accepted from k and k' are equal
  - or, for some state k'', $k \equiv k''$ and $k' \equiv k''$
- The states of $A \nabla A'$ are the equivalence classes of $\equiv$

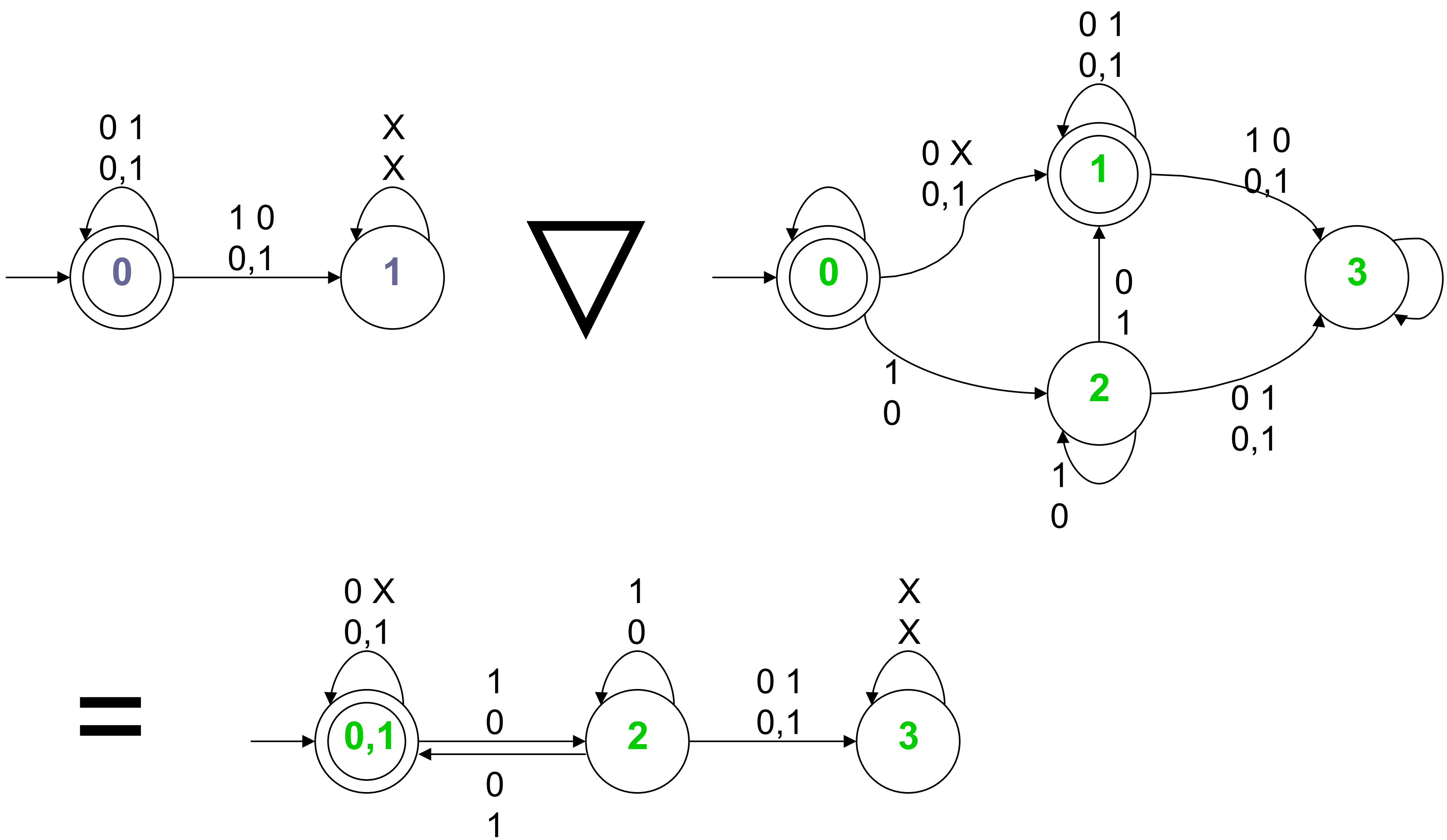
Example

0	0	1	
1	3		
0			
1			
2			
3			





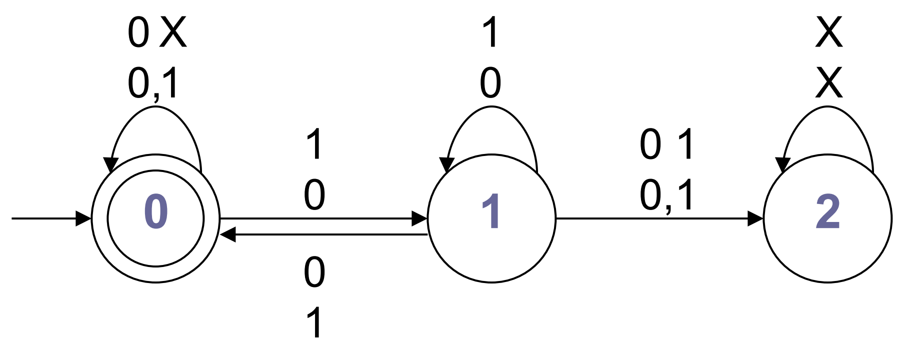
Example

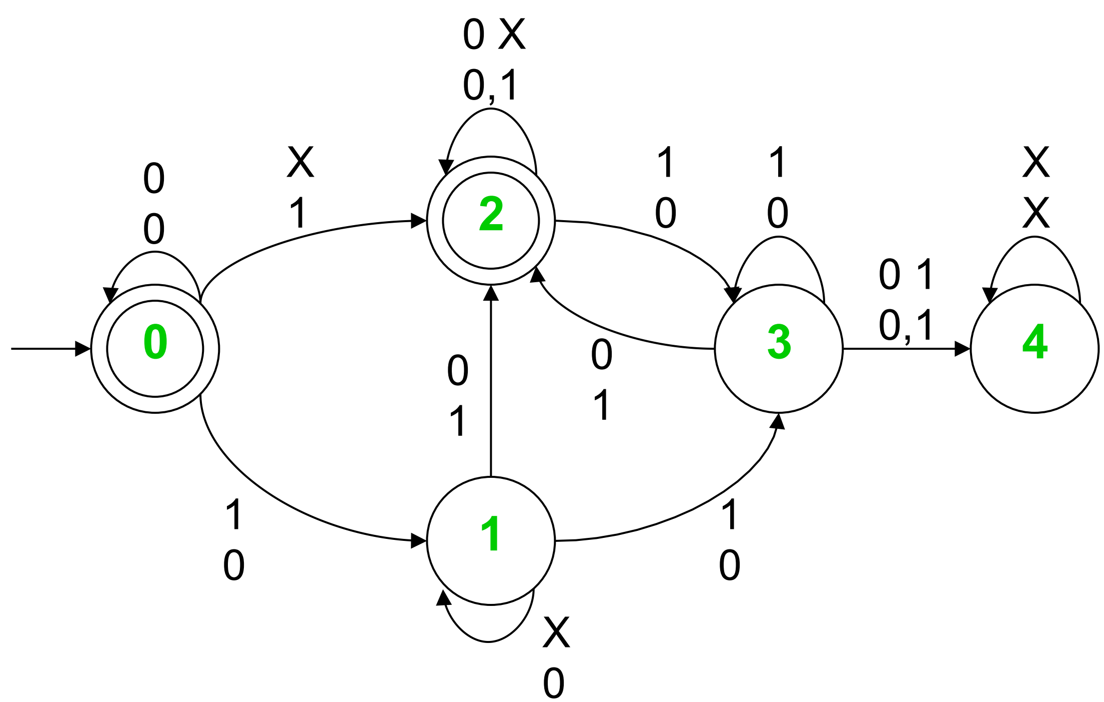


0 1
0,1
1 0
0,1
X
X
0
1
0 X
0,1
0 1
0,1
1 0
0,1
X
X
0
1
2
3
0
1
1
0
1
0
0 1
0,1
=
0 X
0,1
1
0
1
0
0
1
0 1
0,1
X
X
0,1
2
3

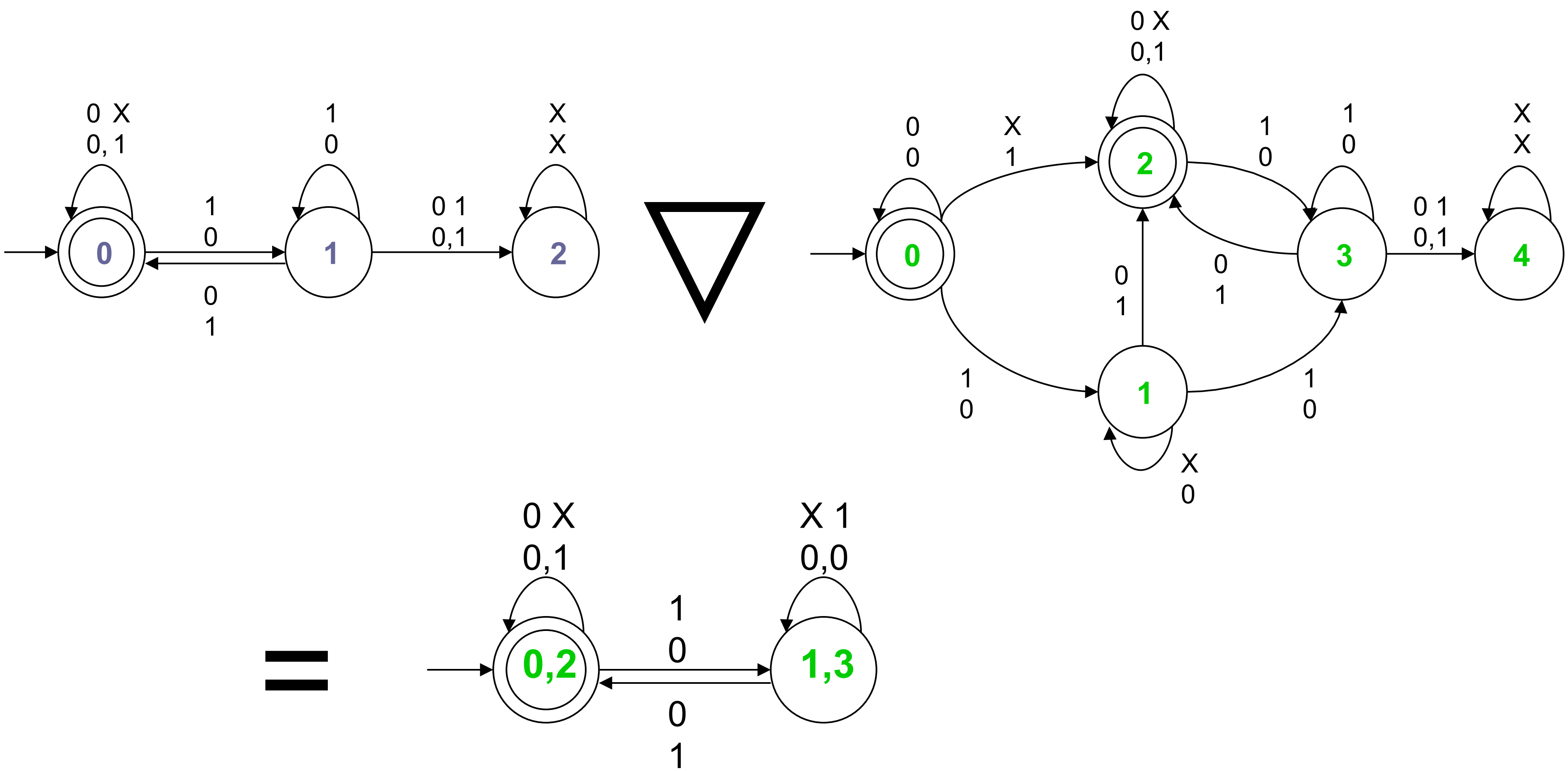
Example

0	0	2	
1	1	3	
2	4		
0			
1			
2			
3			
4			





Example



0 X
0, 1
1
0
X
X
0
1
0
0 1
0,1
0
1
2
∇
0 X
0,1
0
0
X
1
1
0
1
0
X
X
0 1
0,1
0
2
3
4
0
1
0
1
1
0
1
0
X
0
1
=
0 X
0,1
X 1
0,0
1
0
0
1
0,2
1,3

An exactness result (some definitions)

- An automaton (Q,Σ,δ,q_0,F) is called state-disjoint if for all $q_i \neq q_j \in Q$, $L(q_i) \cap L(q_j) = \emptyset$
- An automaton $(Q_1,\Sigma,\delta_1,q_{01},F_1)$ is called weakly equivalent to $(Q_2,\Sigma,\delta_2,q_{02},F_2)$ iff there exists

  $f: Q_1 \rightarrow Q_2$, such that:
  - $f(q_{01})=q_{02}$
  - $f(\delta_1(q,\sigma))=\delta_2(f(q), \sigma)$ for all $q \in Q_1$ and $\sigma \in \Sigma$
  - $f(q) \in F_2$ for all $q \in F1$

An exactness result

- If
  - a least fixpoint is represented by a state-disjoint automaton A_∞
  - and, the first automaton A_s in the approximate sequence is weakly equivalent to A_∞
- then
  - the approximate sequence converges to the exact least fixpoint