Abstraction/Refinement Verification

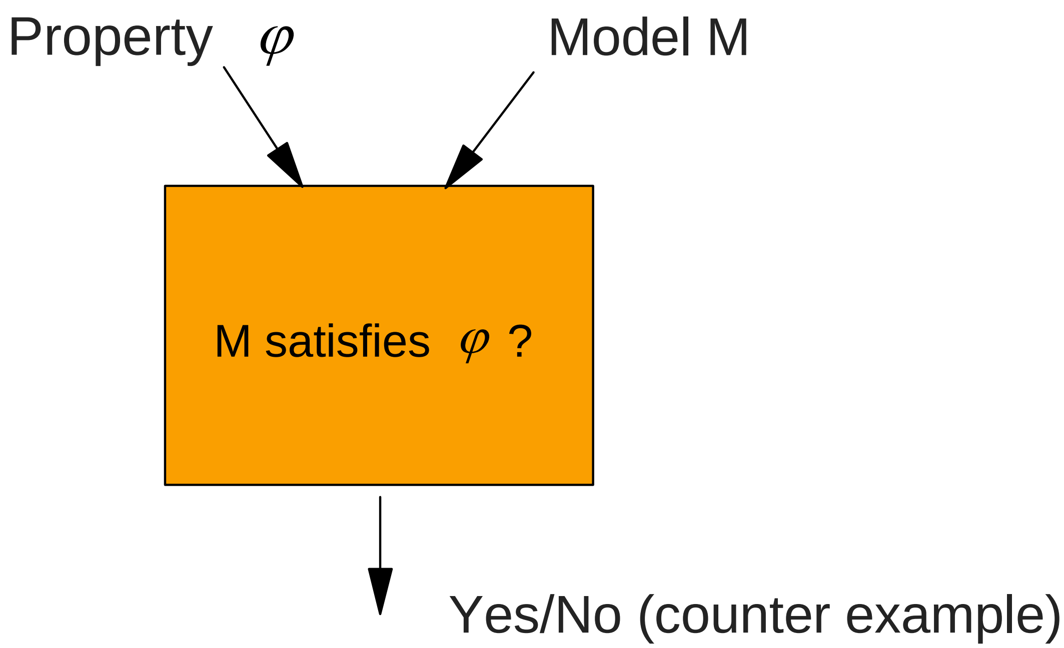
Algorithms with Backtracking and Layering

Sharon Barner

12.9.02

IBM Research Lab in Haifa

Model Checking

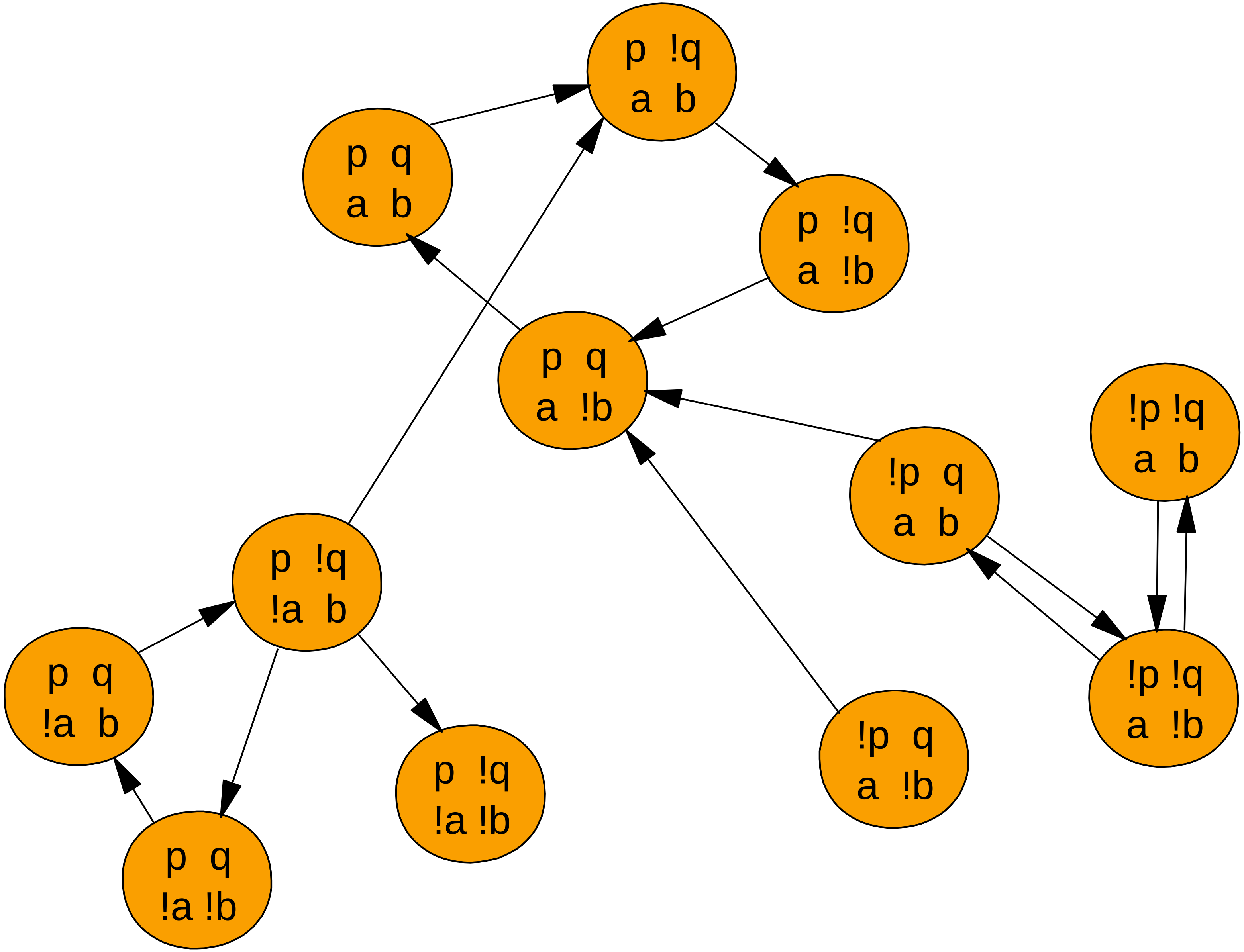


- **The model is given by a Kripke Structure.**
  - Each state is an assignment to all state variables.
- **The property is given in Temporal logic.**
  - In our case Sugar.

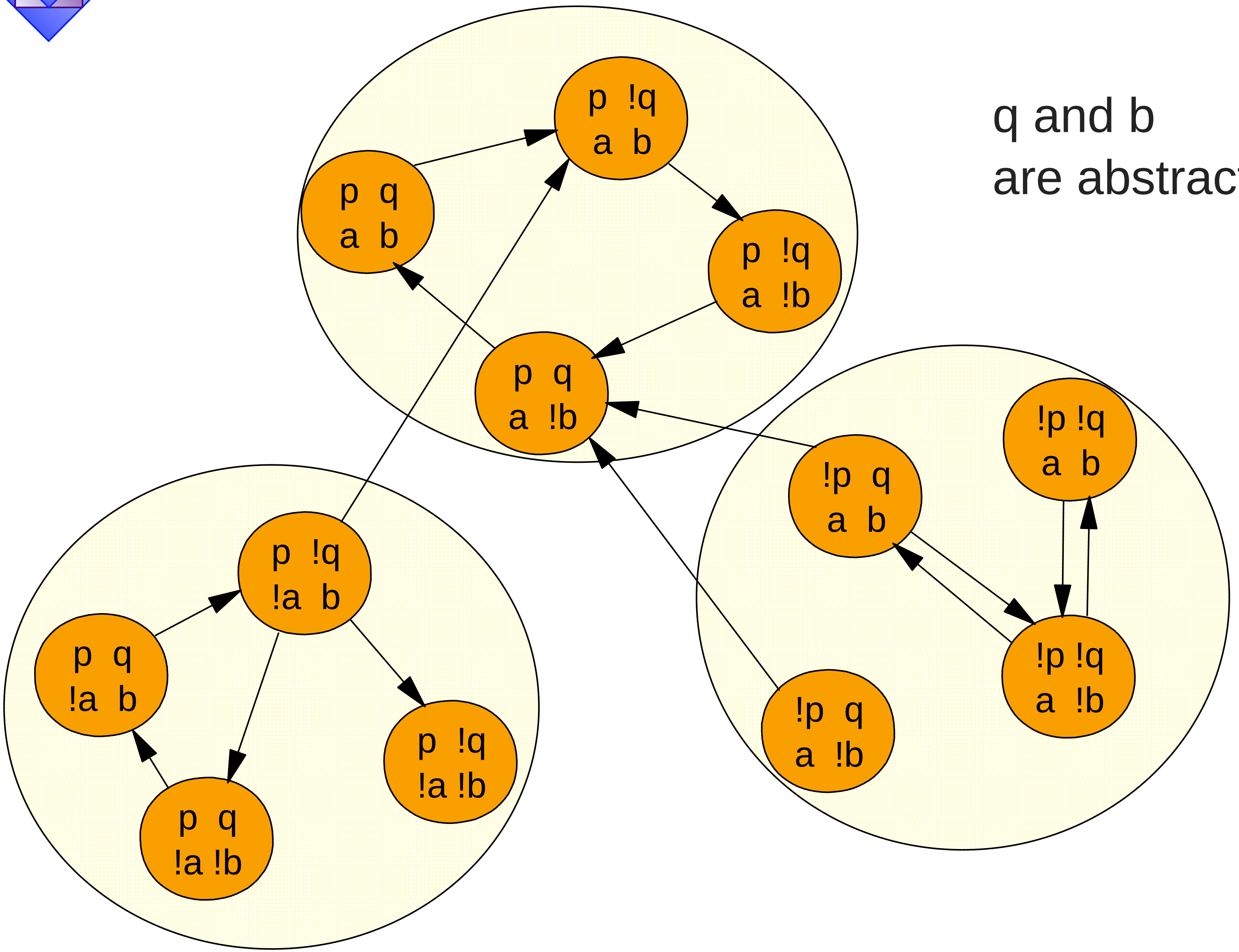
State Explosion Problem

- The main problem of model checking is its high memory requirements. We refer to this problem as "the state explosion problem"
- A possible approach to overcome the state explosion problem is by using abstraction of the model.
- In this work we present an "Abstraction/Refinement" algorithm for always(p) formulas (where p is a boolean expression).

Model Before Abstraction

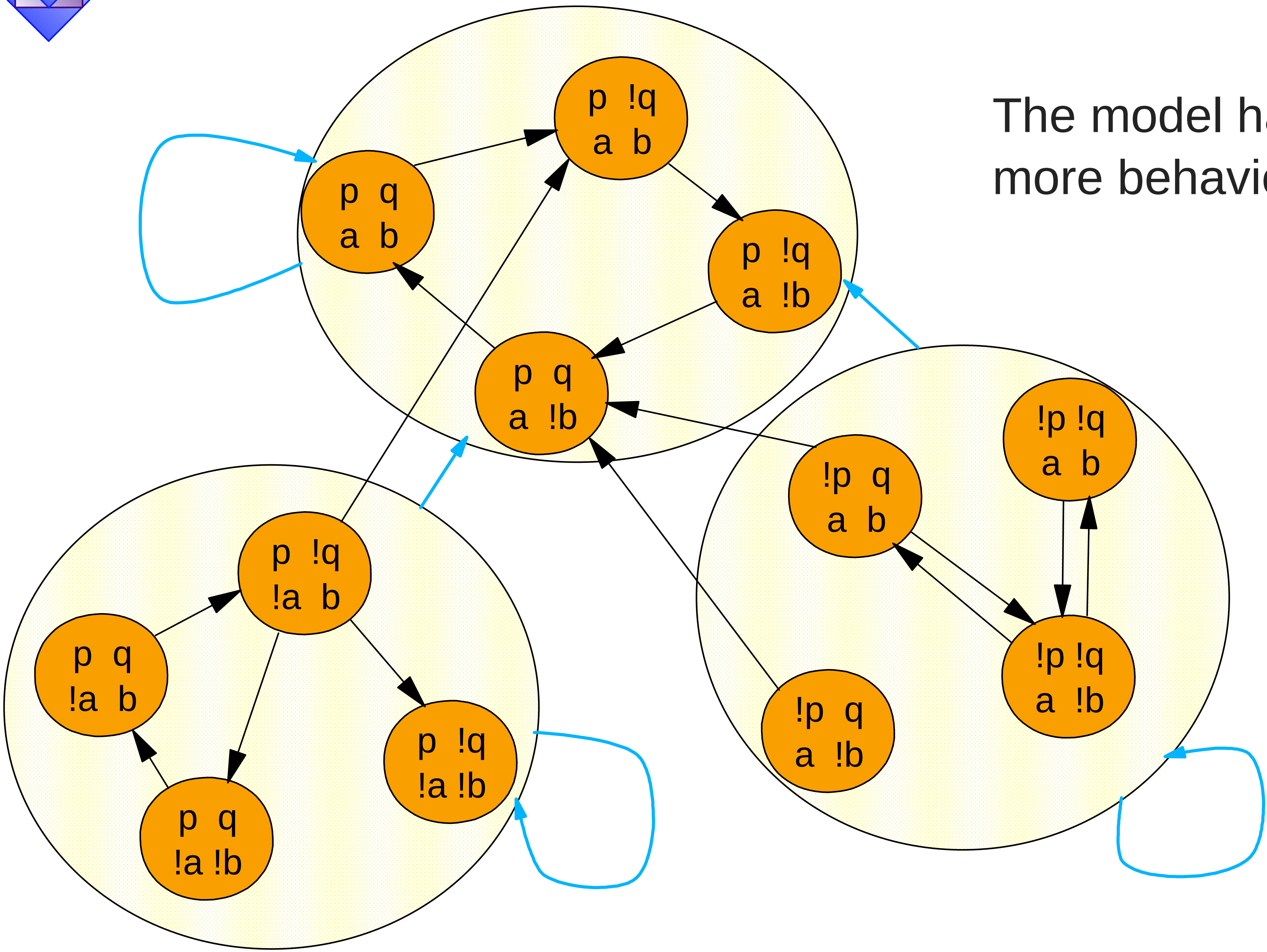


Abstraction



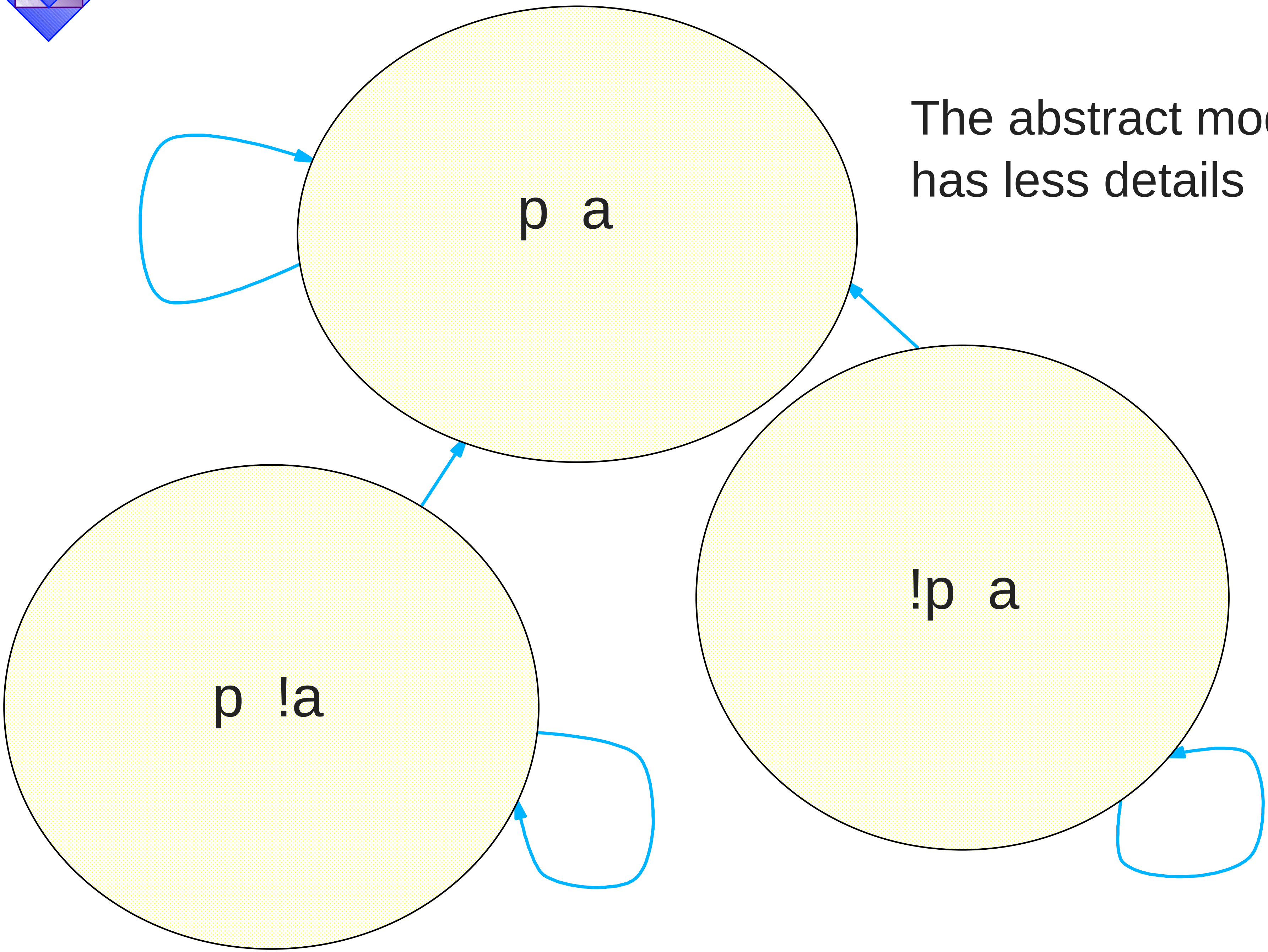
q and b
are abstracted

Abstraction



The model has more behaviors

Model After Abstraction



The abstract model has less details

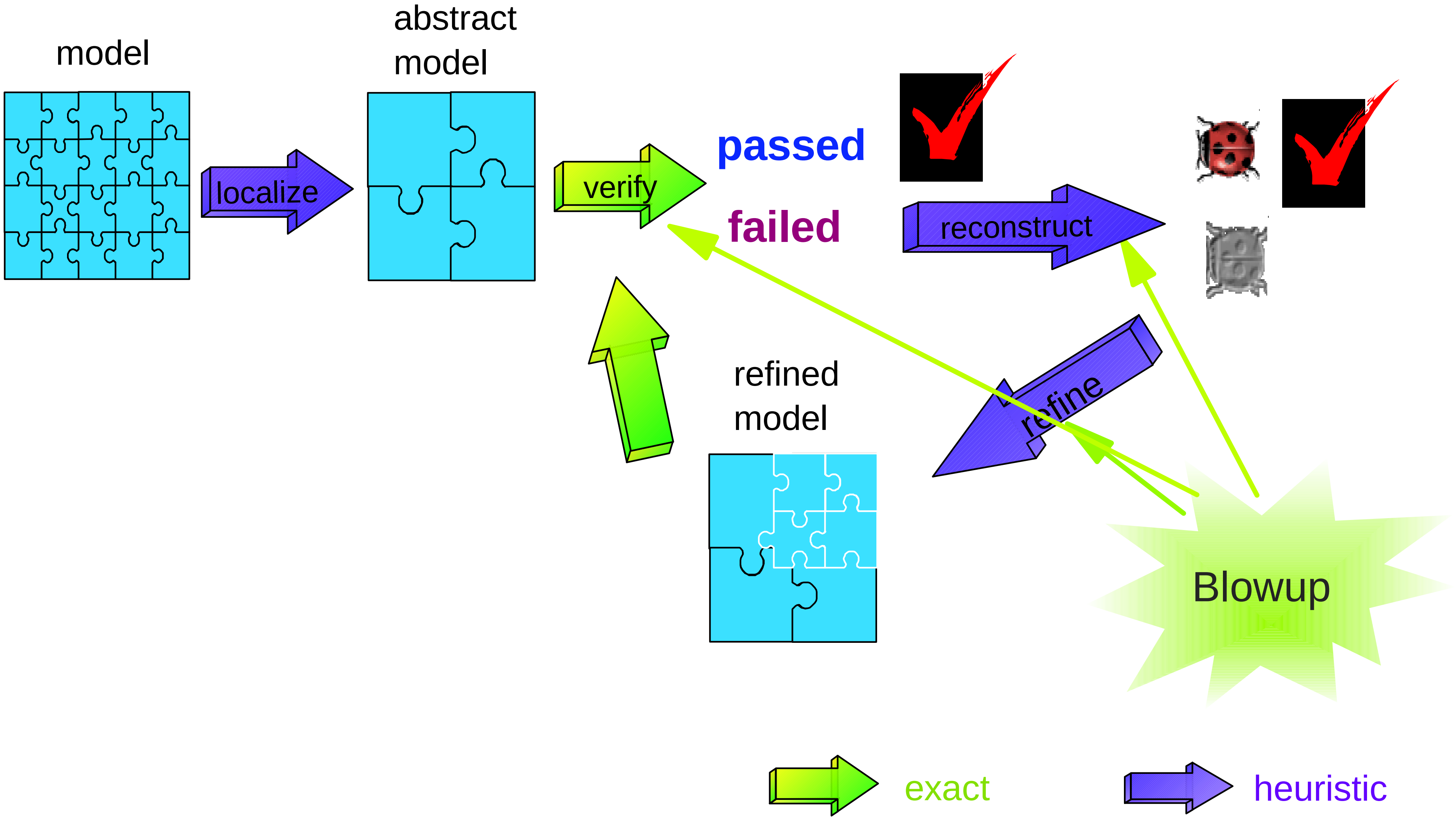
Abstraction

- **A model M' is bigger than a model M if it contains more behaviors than M (M < M').**
- **M' is an abstraction of M if it is bigger than M but it is "less detailed".**
- **Less memory is required in order to represent the abstracted model M'.**
- **If an ACTL formula is true on M' it is also true on M**
  - (Contains all the on-the-fly subset of sugar)

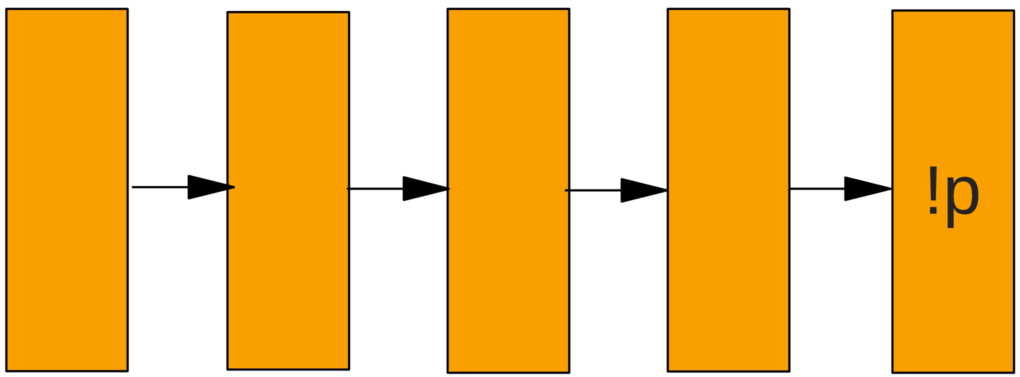
Localization reduction

- **Automatic Abstraction/Refinement algorithm.**
- **Based on the fact that most properties are local**
  - Influenced only by the closed environment of the signals in the property.
- **Good for verification of large designs.**

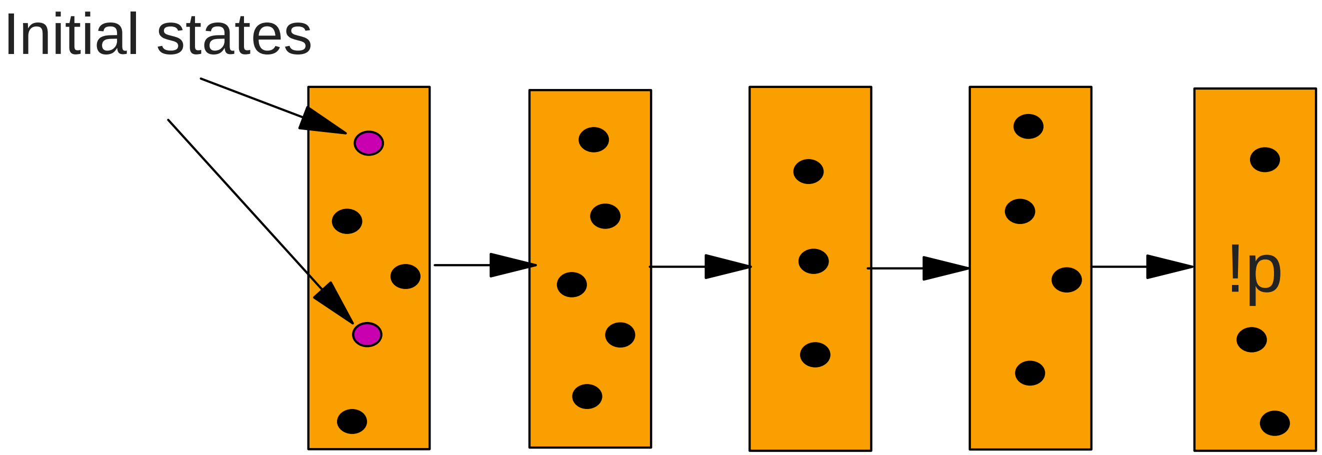
Localization Reduction - Overview



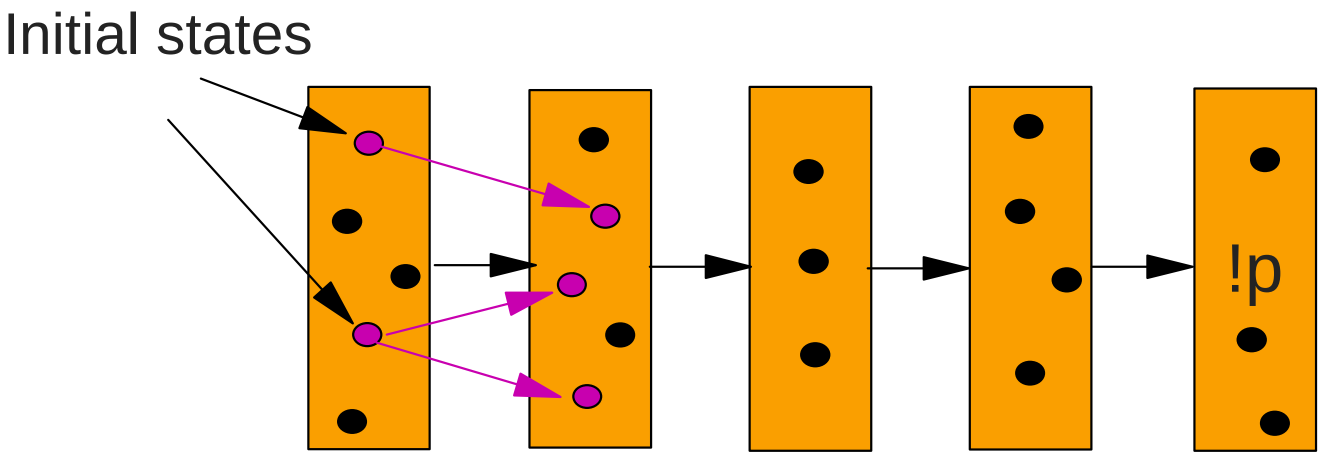
Abstract Counter Example



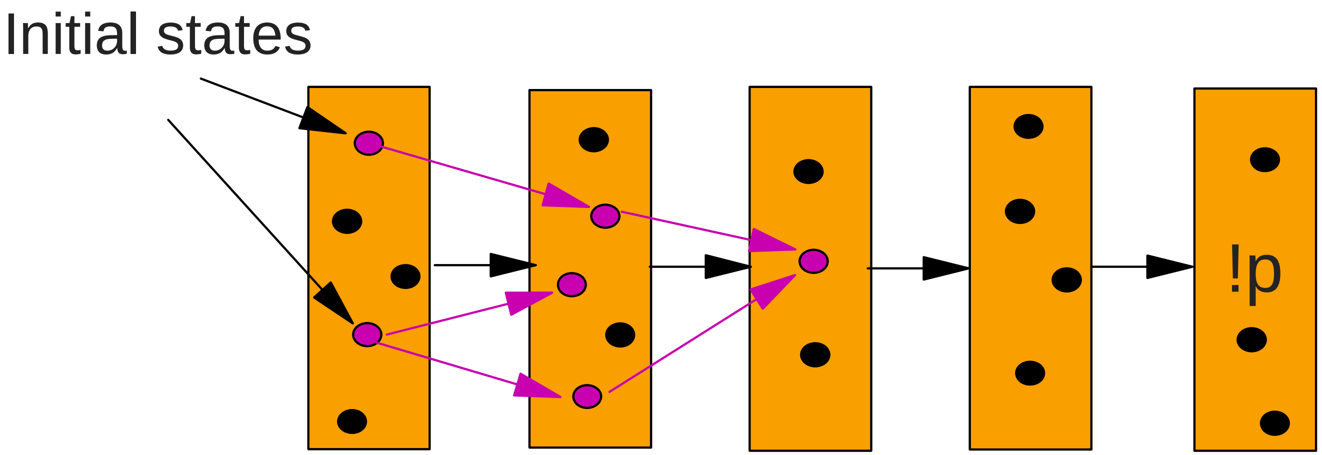
Reconstruction



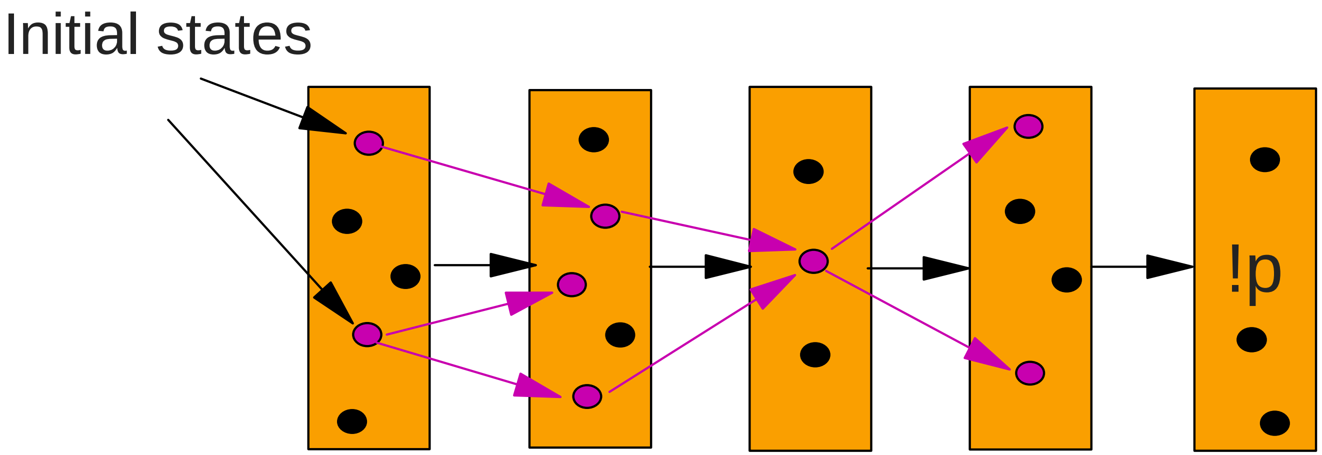
Reconstruction



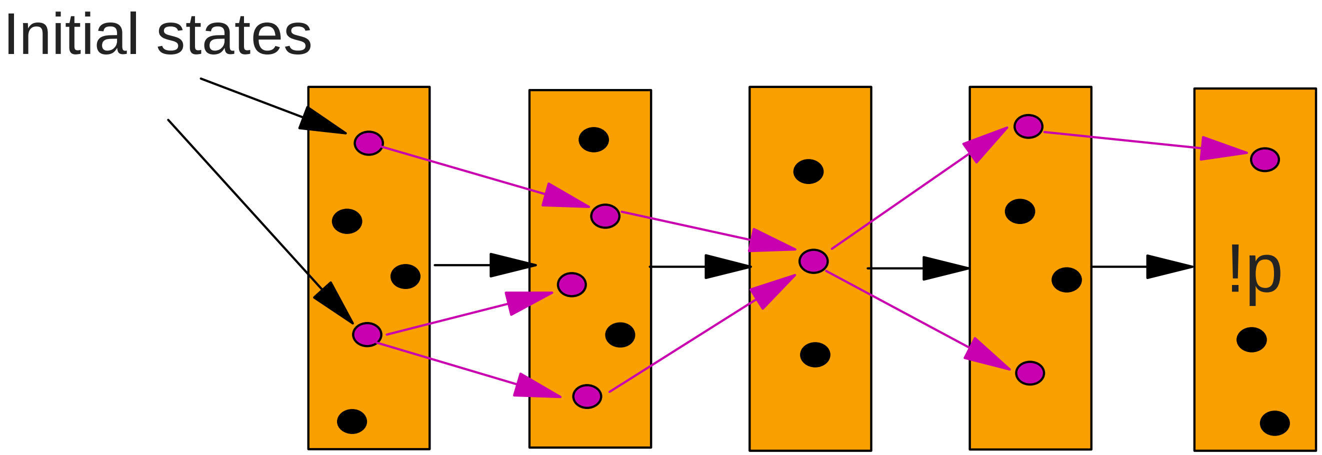
Reconstruction



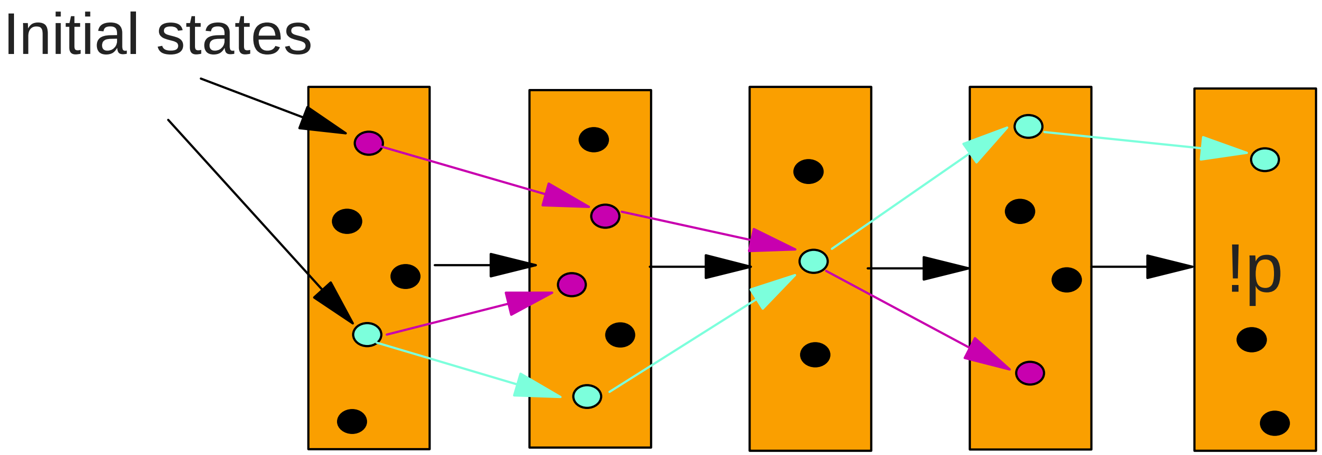
Reconstruction



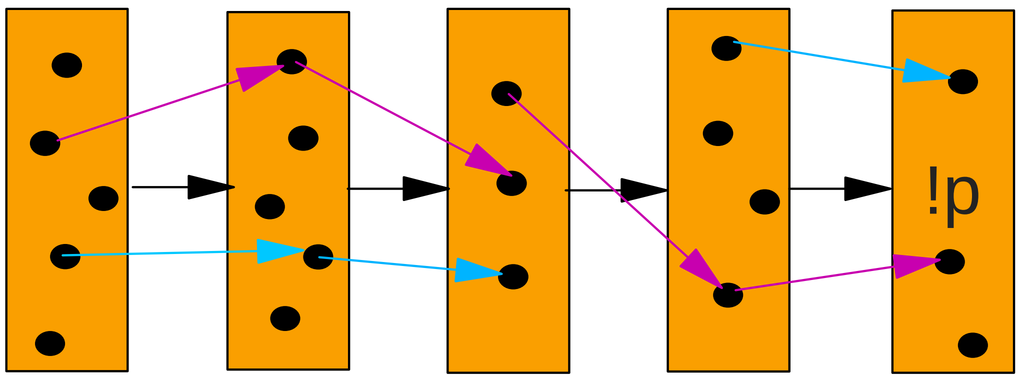
Reconstruction



Concrete Counter Example

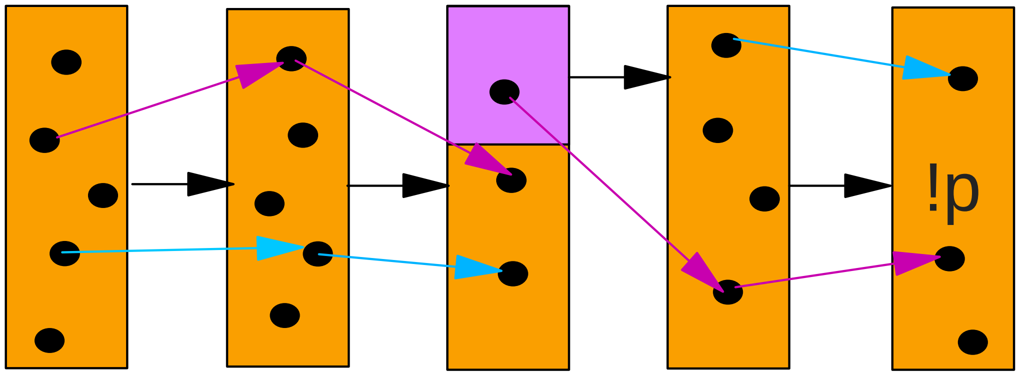


Bogus Counter Example



There are cases where there is no concrete counter example which is consistent with the abstract counter example.

Refinement



Split an abstract state in a way that removes the bogus counter example.

Definitions - Projection

- **Model M has a finite set of variables $V = \{v_1, \ldots v_n\}$ called state variables**
  - **a vector of values $(v_1, \ldots, v_n)$ is a state of M.**
- **<u>Projection</u> of $V' \subseteq V$ of a set A of states, denoted *project*(A, V'),**
  - means the removal of all coordinates not in V' **from elements in A.**

$$V = \{v_1, v_2, v_3\} \qquad V' = \{v_1, v_2\}$$

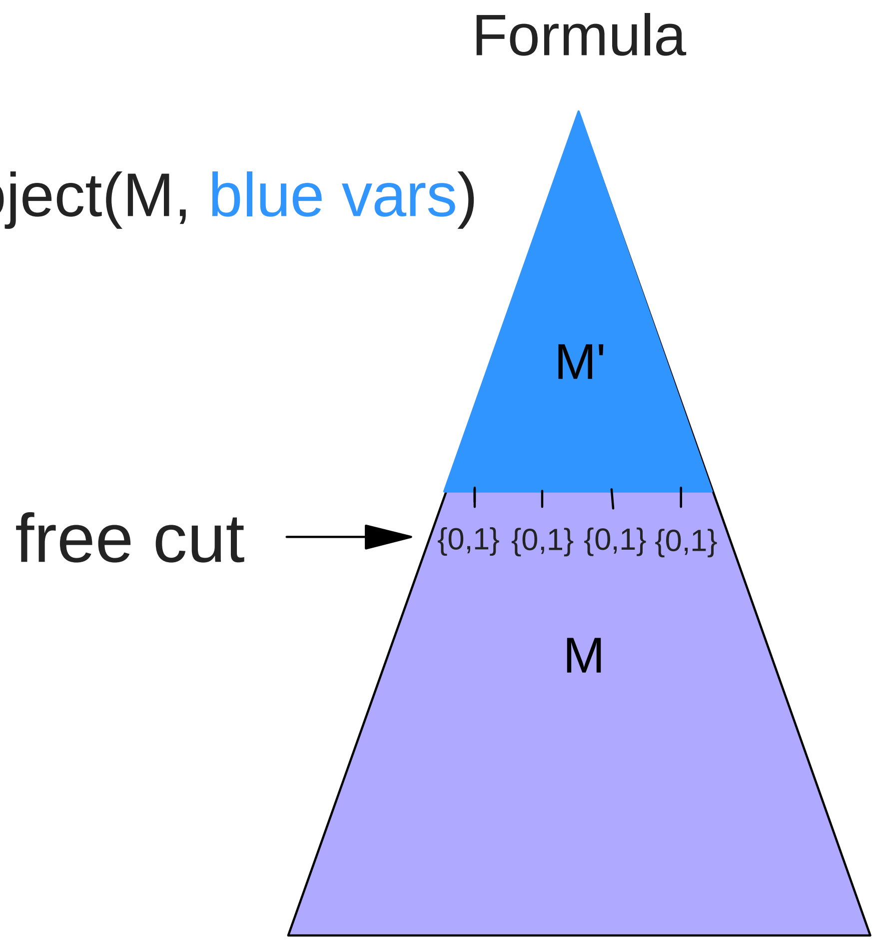
$$A = \left\{ \begin{matrix} (0,1,0) \\ (0,1,1) \\ (1,1,1) \end{matrix} \right\} \quad \textit{project}(A, V') = \left\{ \begin{matrix} (0,1) \\ (1,1) \end{matrix} \right\}$$

The abstract model

- **Our model abstraction is limited to projection on a subset V' of the state variables.**
  - the abstract model state variables becomes V' and the variables which influence V' directly.
  - the initial state set is projected onto this set, the transition relation is projected, etc.
  - The next state functions of the variables that influence directly the signals in V' are left nondeterministic.

Example - Abstraction by projection

M' = project(M, blue vars)

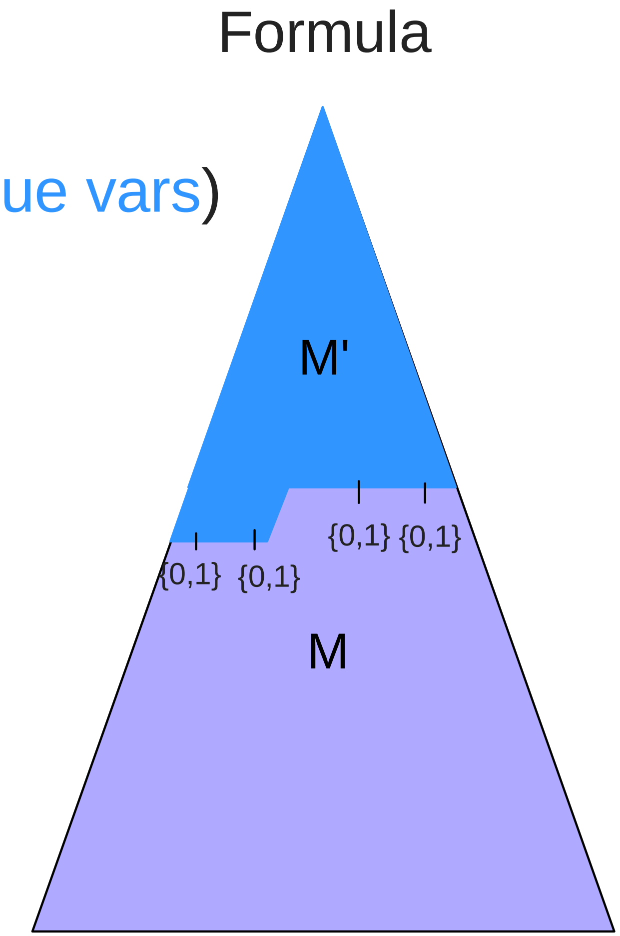


Reconstruction / Refinement

- We choose one abstract counter example.
- If we can extend it to counter example in the original model the property failed.
- Otherwise we refine the abstraction with the variables which have wrong behavior in the abstract counter example (their behavior is not allowed in the original model).

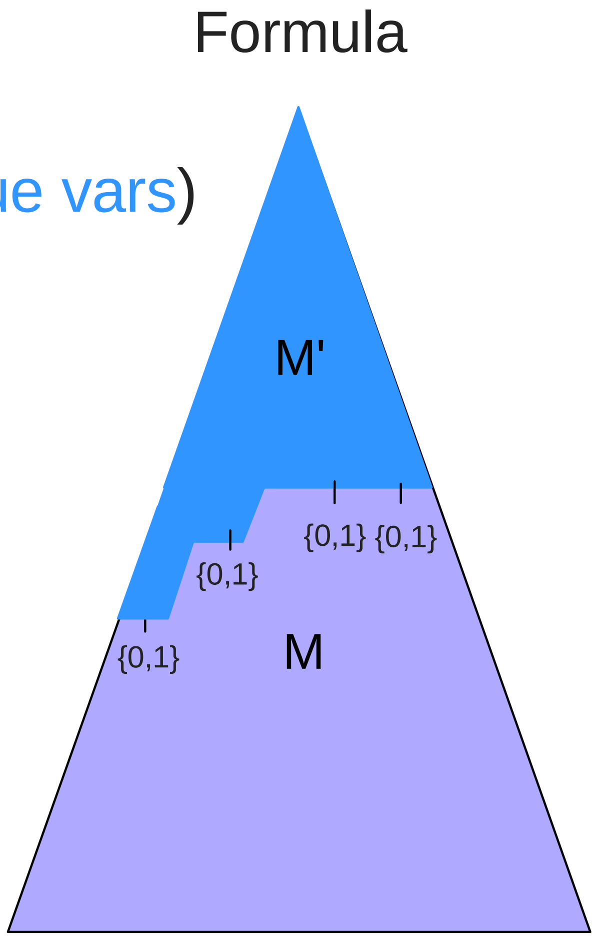
Example - Abstraction by projection

M' = project(M, blue vars)



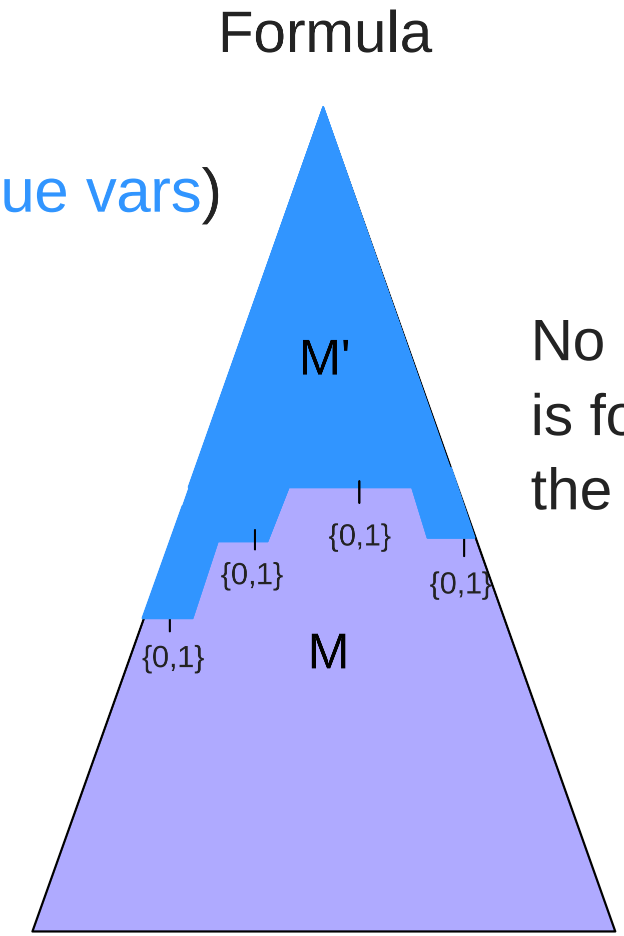
Example - Abstraction by projection

M' = project(M, blue vars)



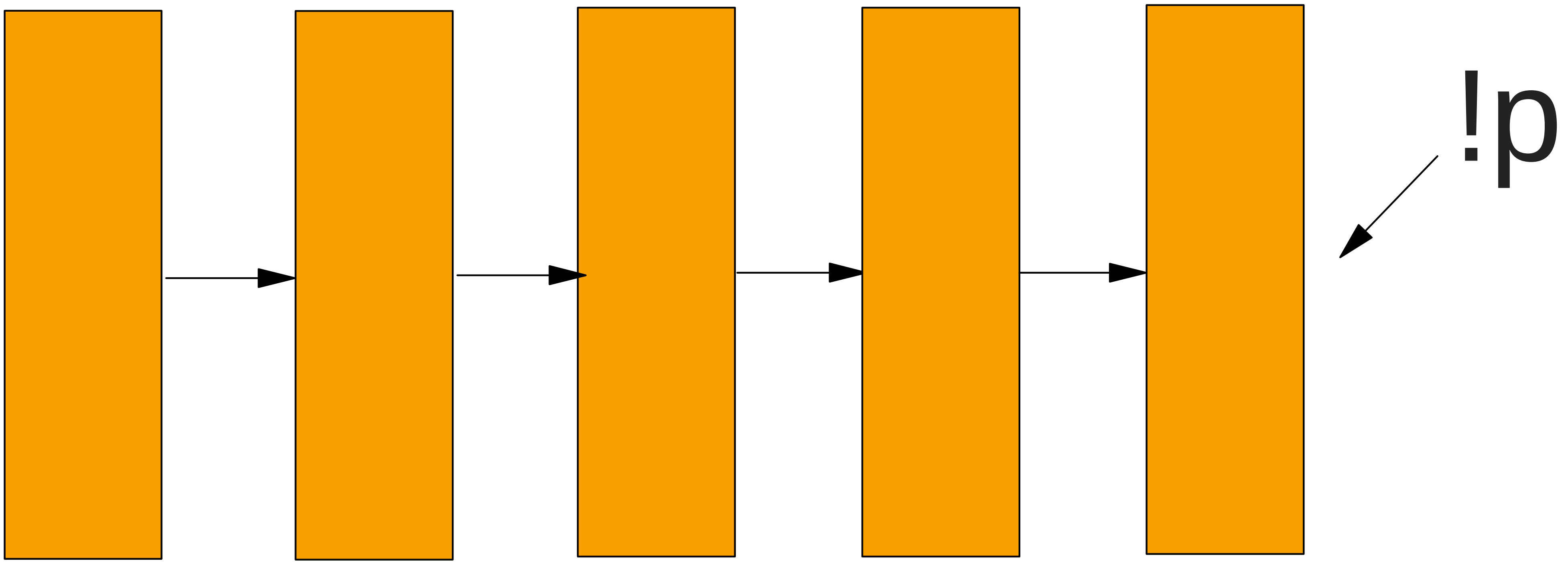
Example - Final abstraction

M' = project(M, blue vars)

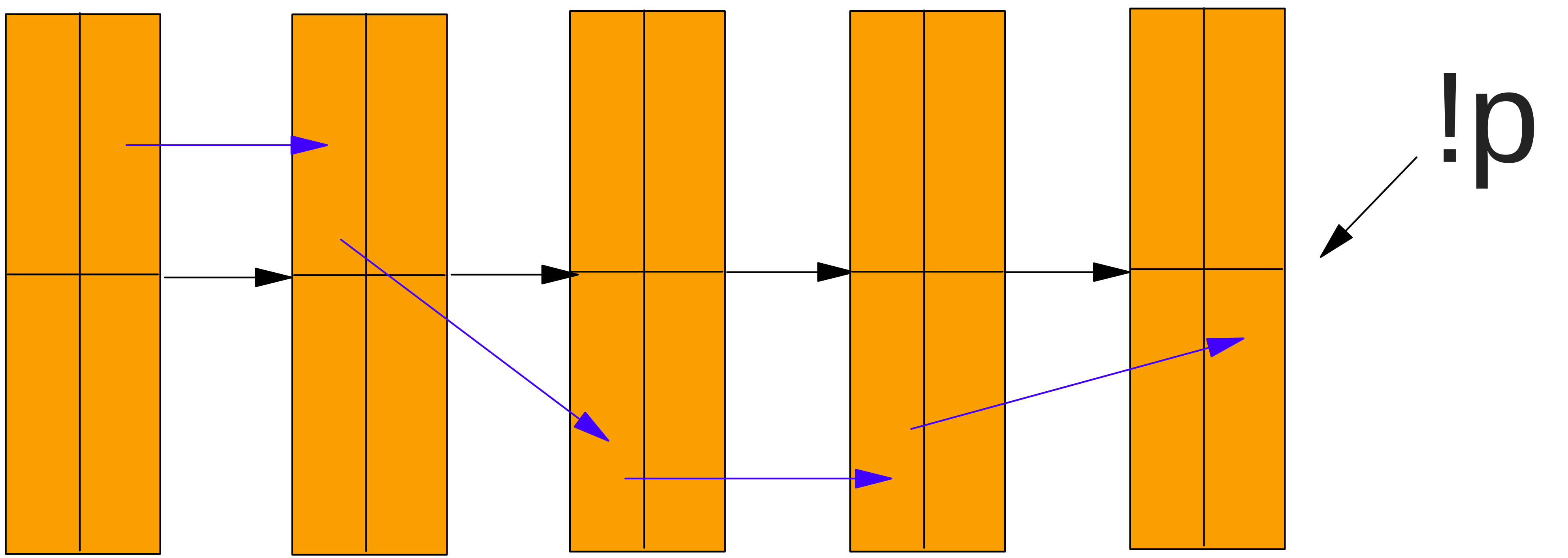


No bogus counter example is found - the formula passes

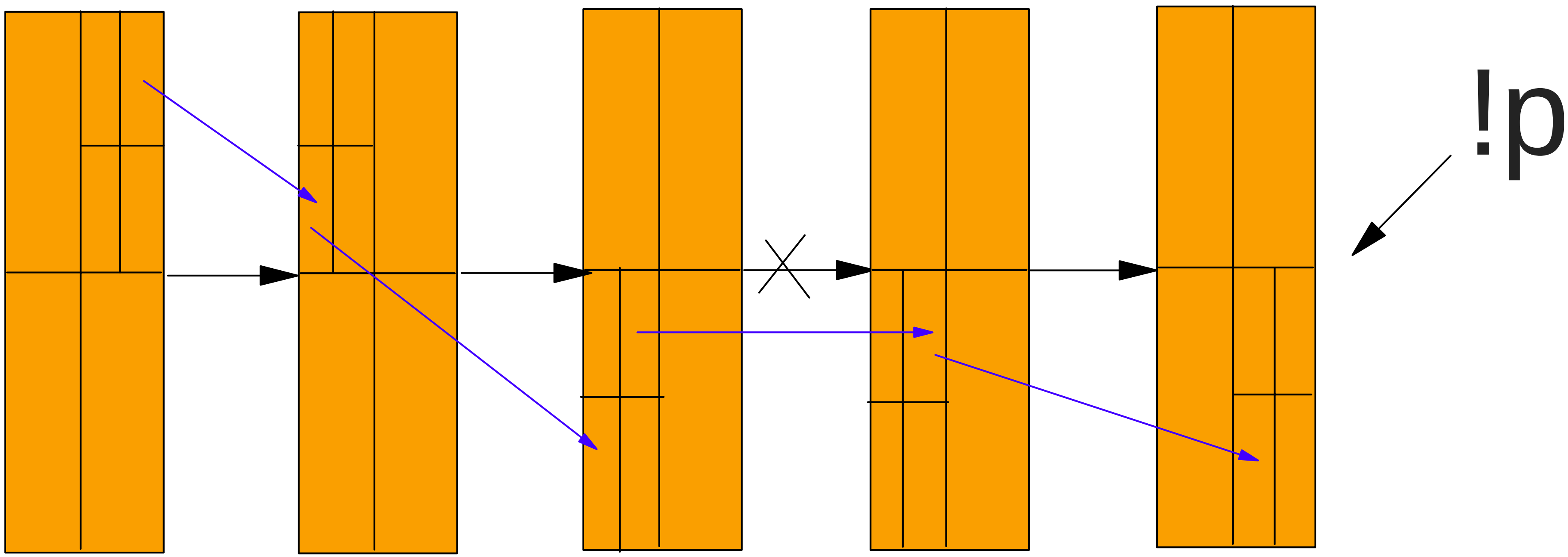
Gradual reconstruction



Gradual reconstruction - layer 1



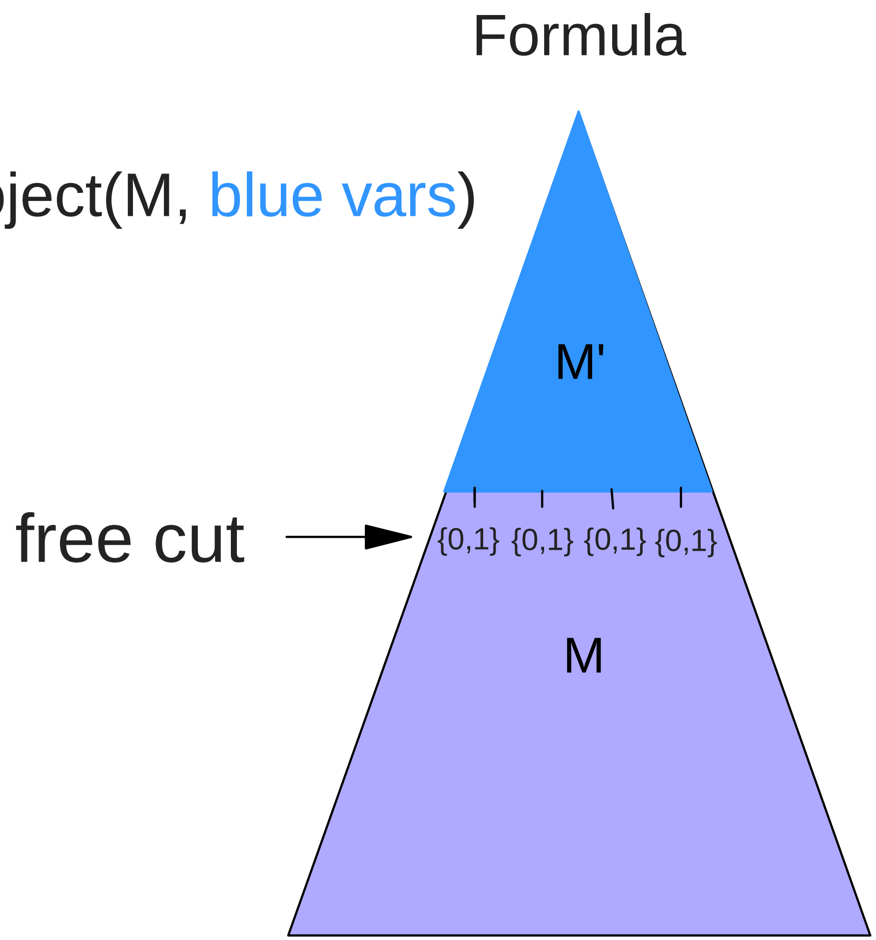
Gradual reconstruction - layer 2



We found that the counter example is bogus without using the original model

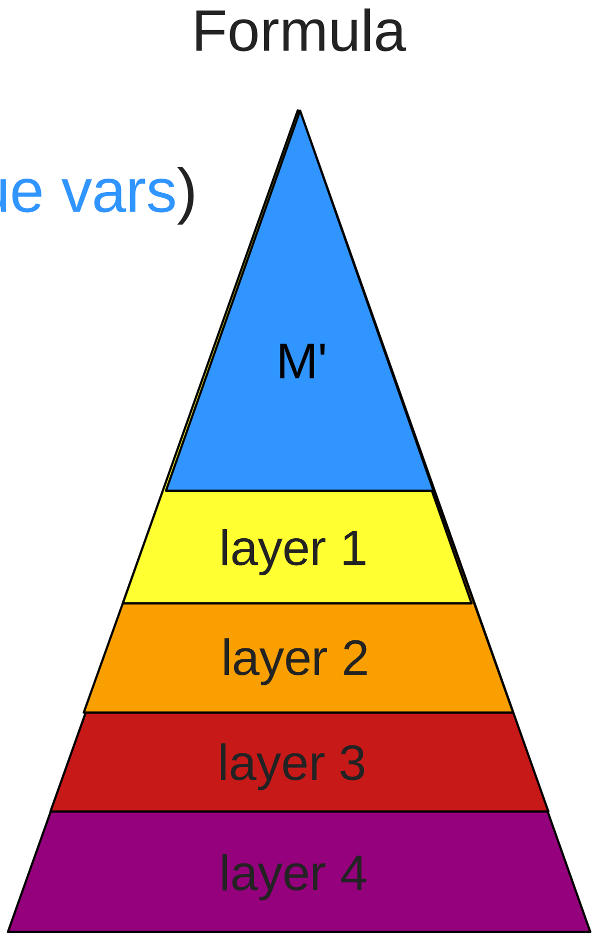
Example - Abstraction by projection

M' = project(M, blue vars)



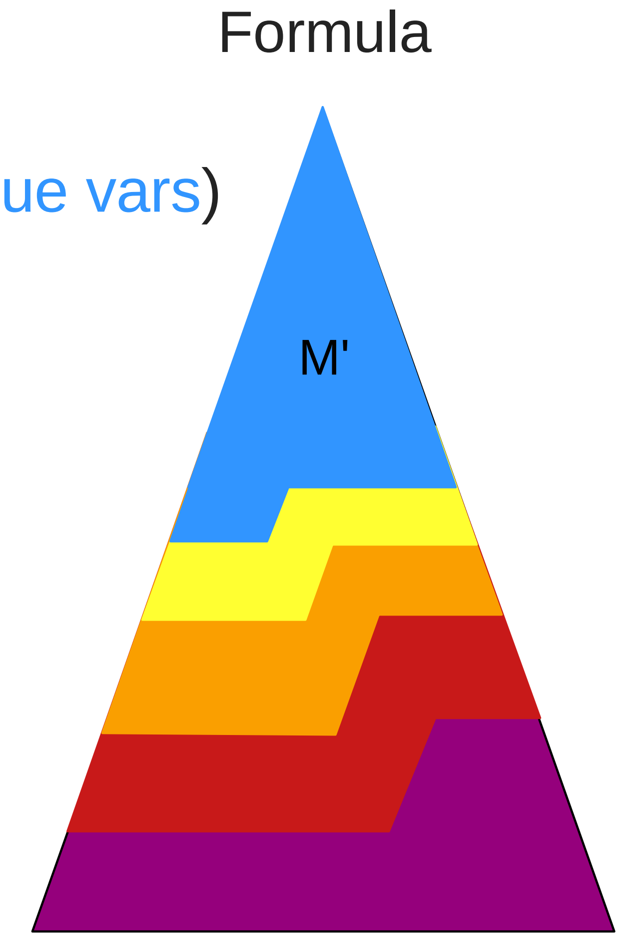
Layers

M' = project(M, blue vars)



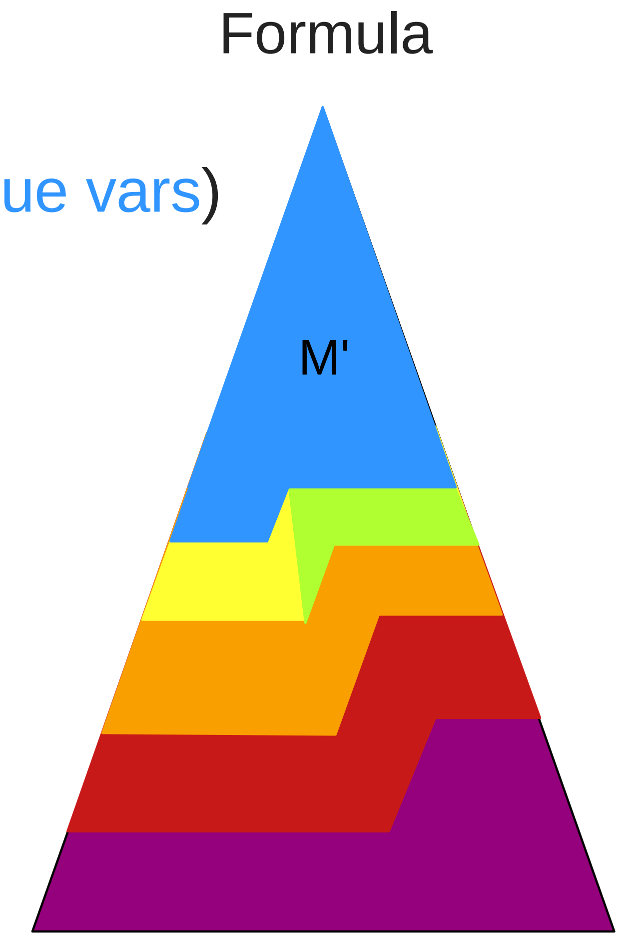
Layers after refinement

M' = project(M, blue vars)



DFS layers

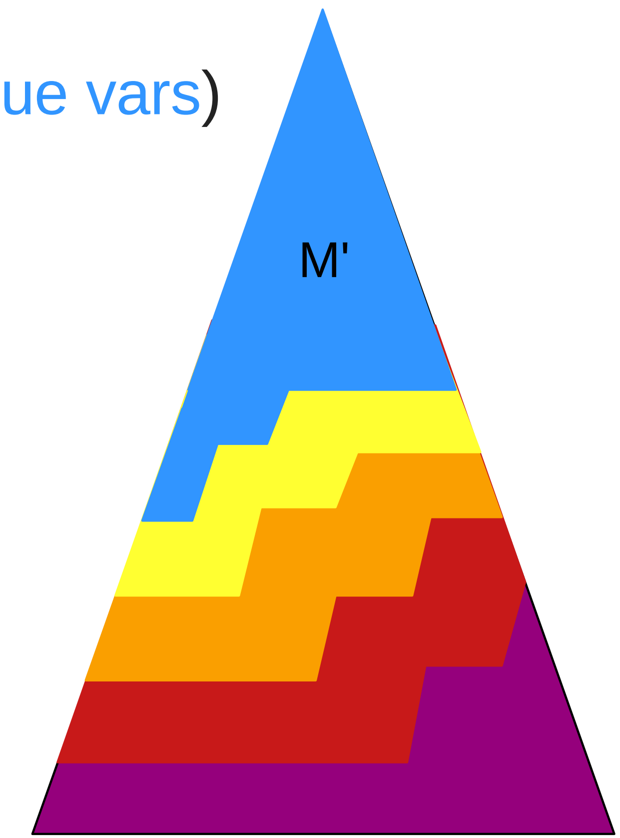
M' = project(M, blue vars)



Layers after refinement

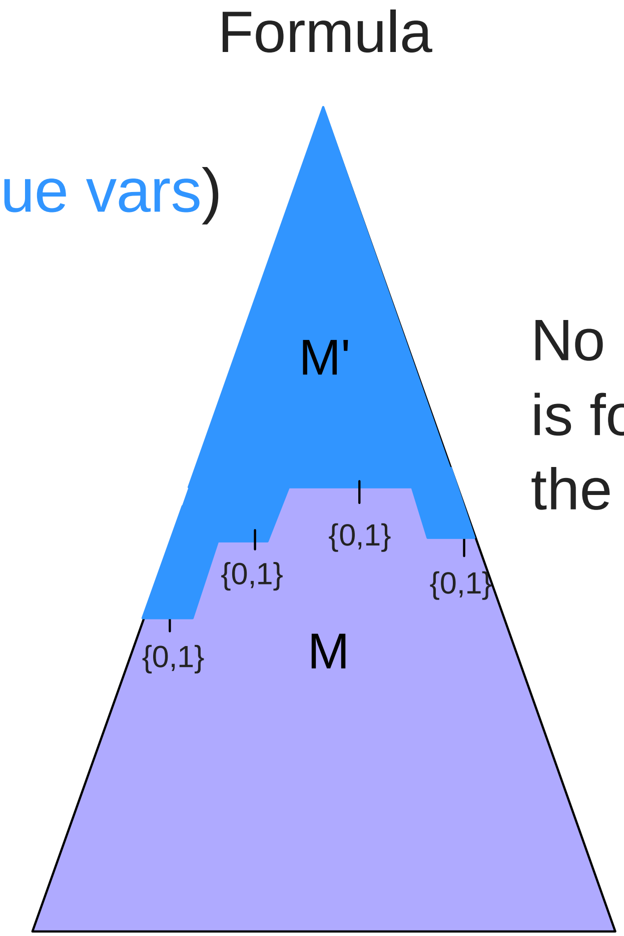
Formula

M' = project(M, blue vars)



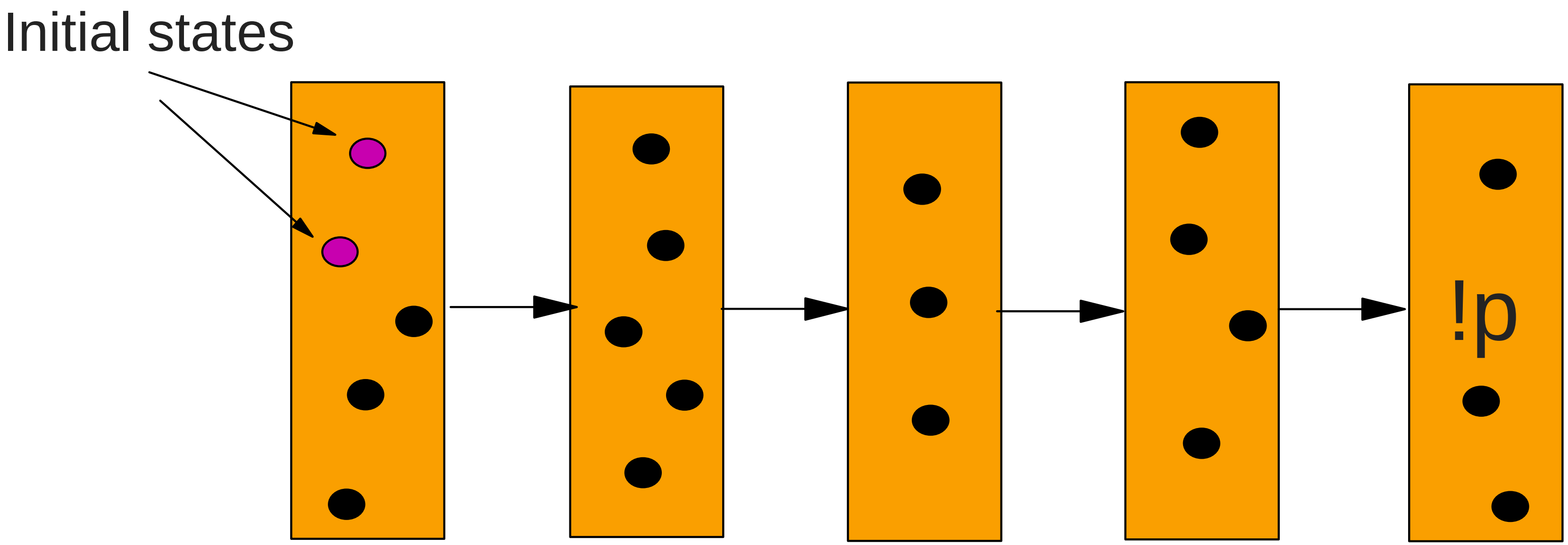
Final abstraction

M' = project(M, blue vars)

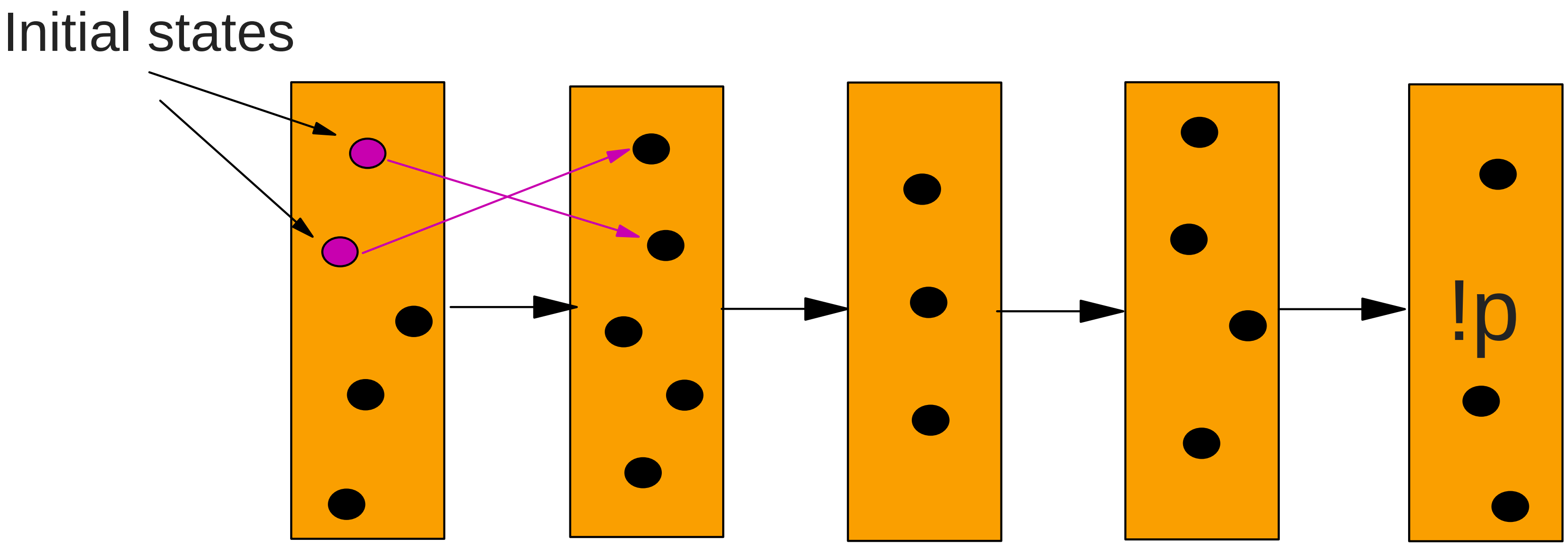


No bogus counter example
is found -
the formula passes

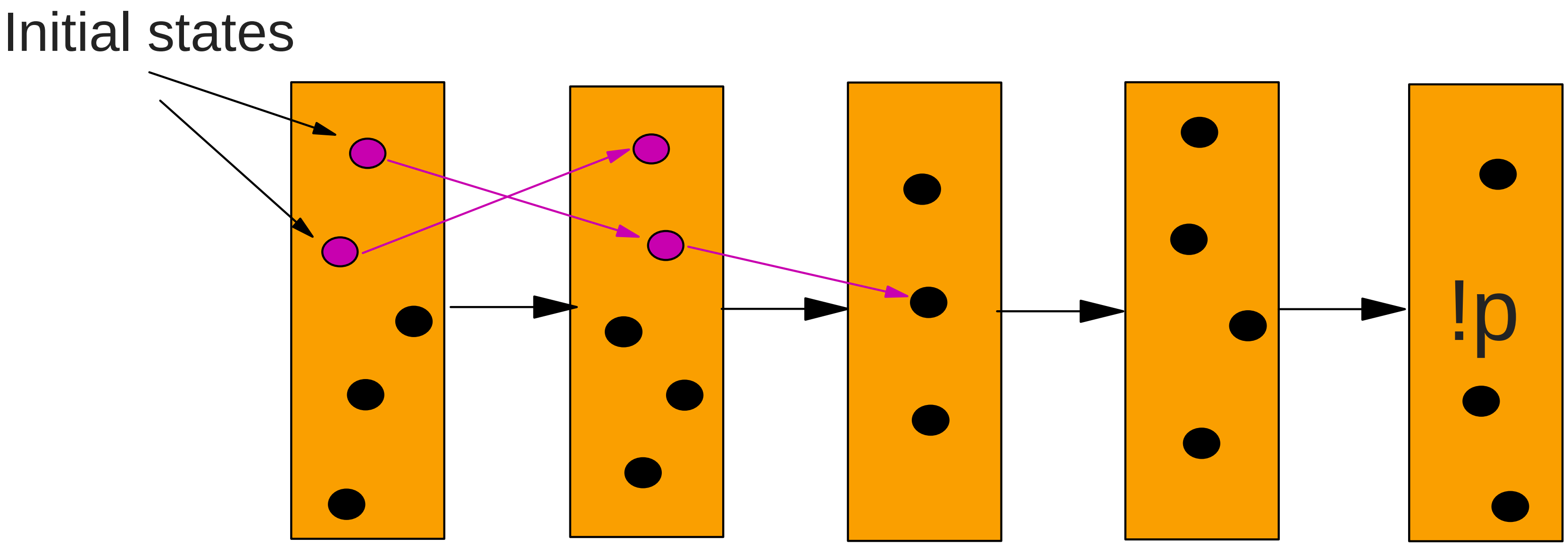
Reconstruction with Abstract State Replacement



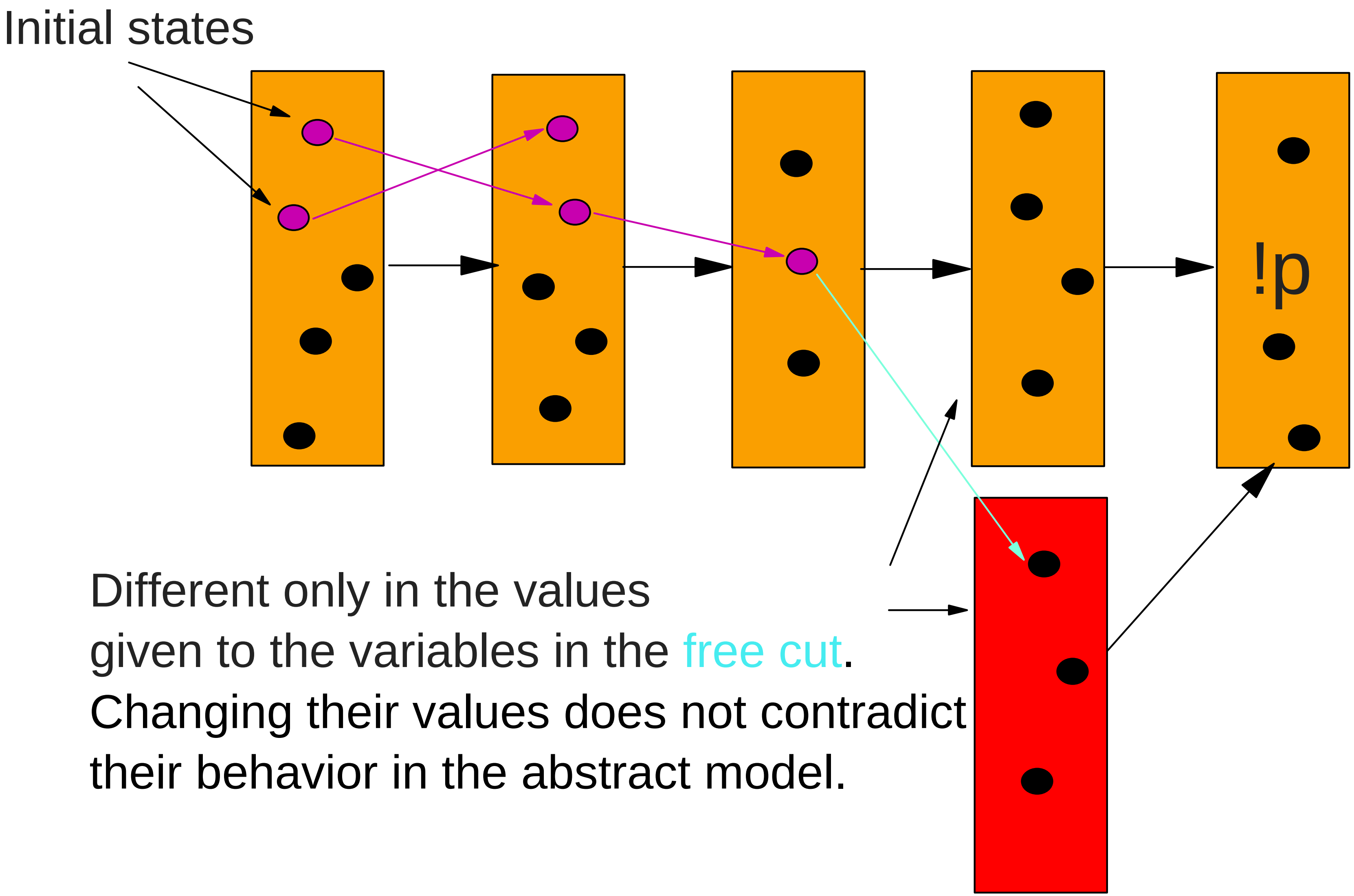
Reconstruction with Abstract State Replacement



Reconstruction with Abstract State Replacement

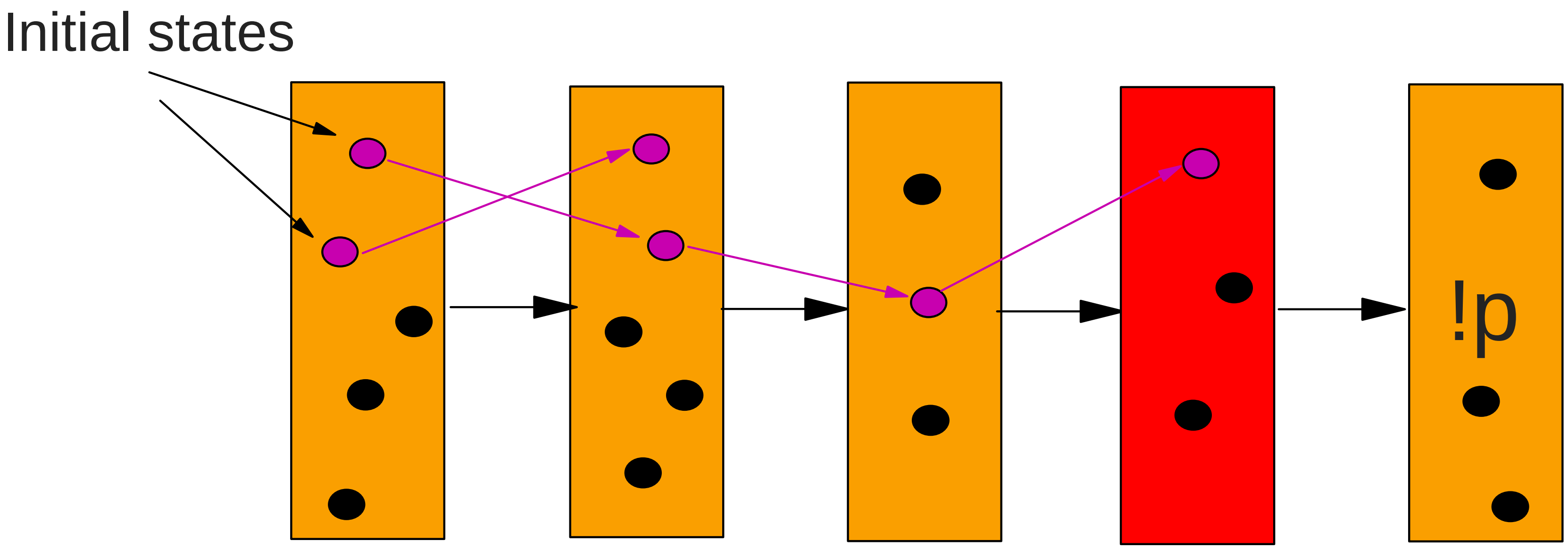


Reconstruction with Abstract State Replacement

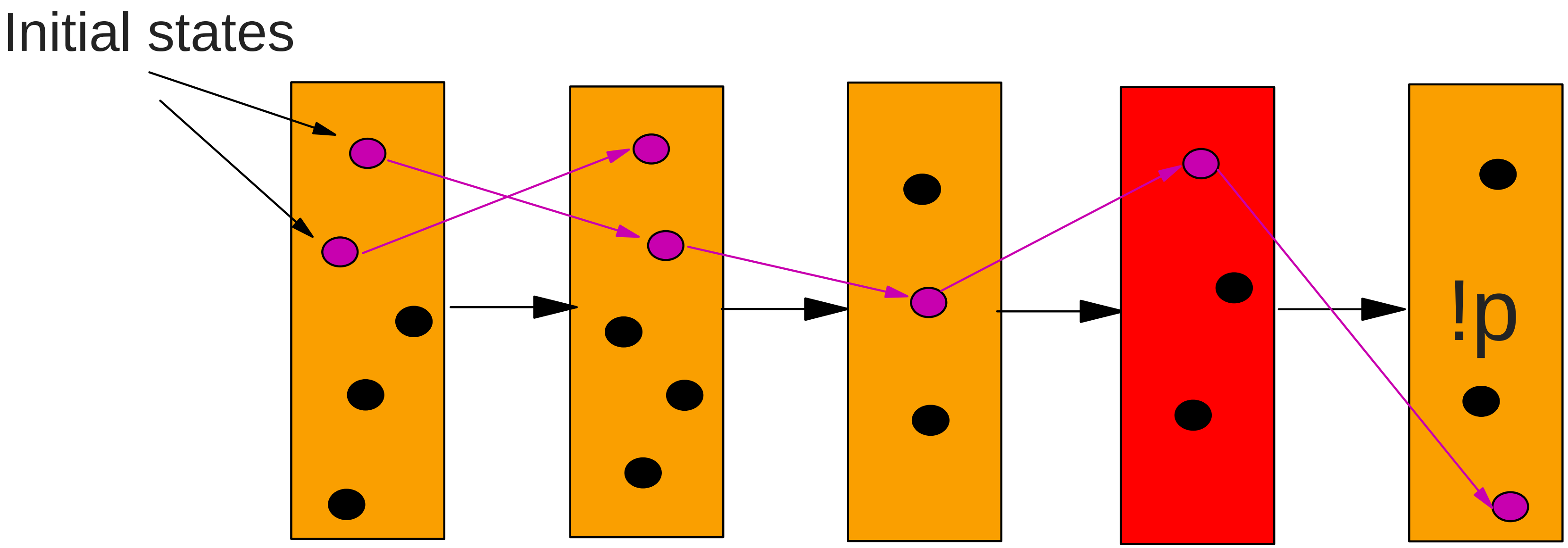


Different only in the values
given to the variables in the free cut.
Changing their values does not contradict
their behavior in the abstract model.

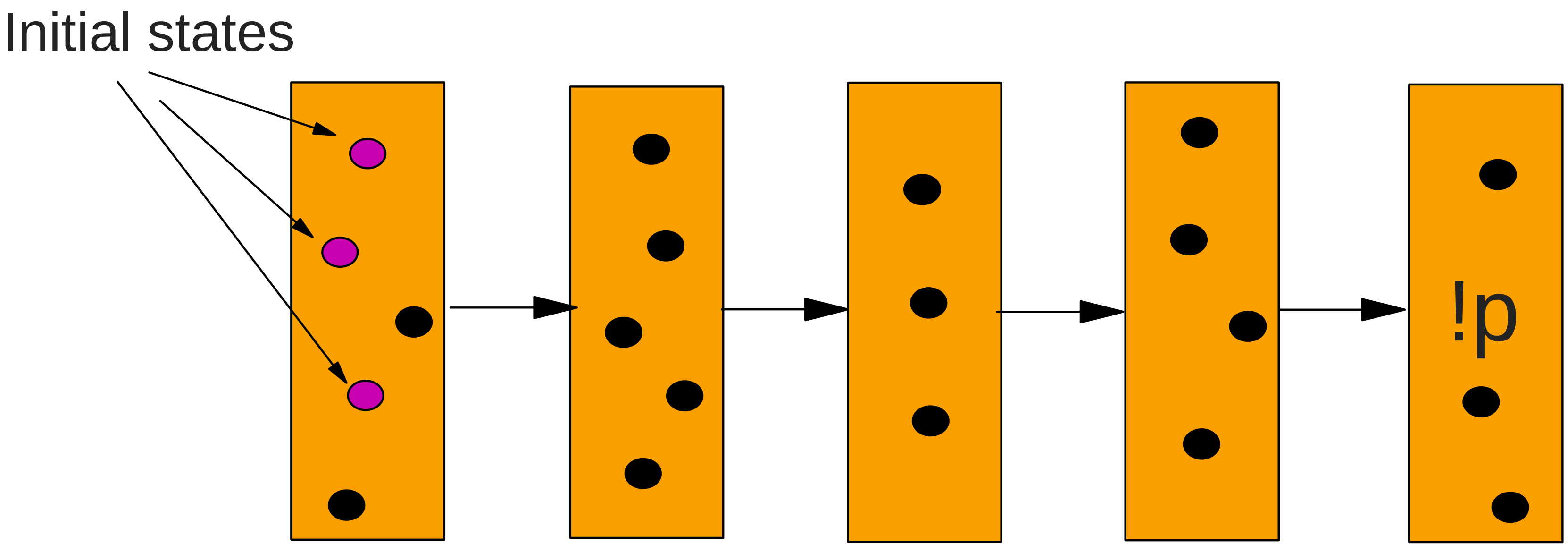
Reconstruction with Abstract State Replacement



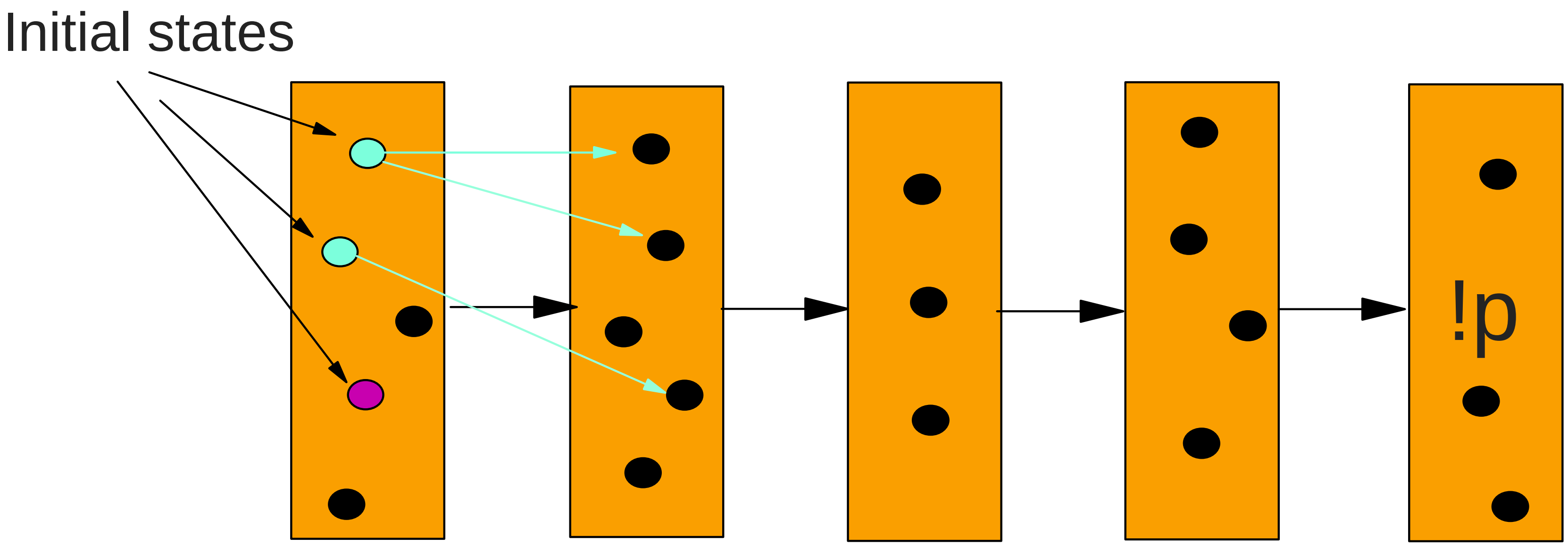
Reconstruction with Abstract State Replacement



Reconstruction with Backtracking

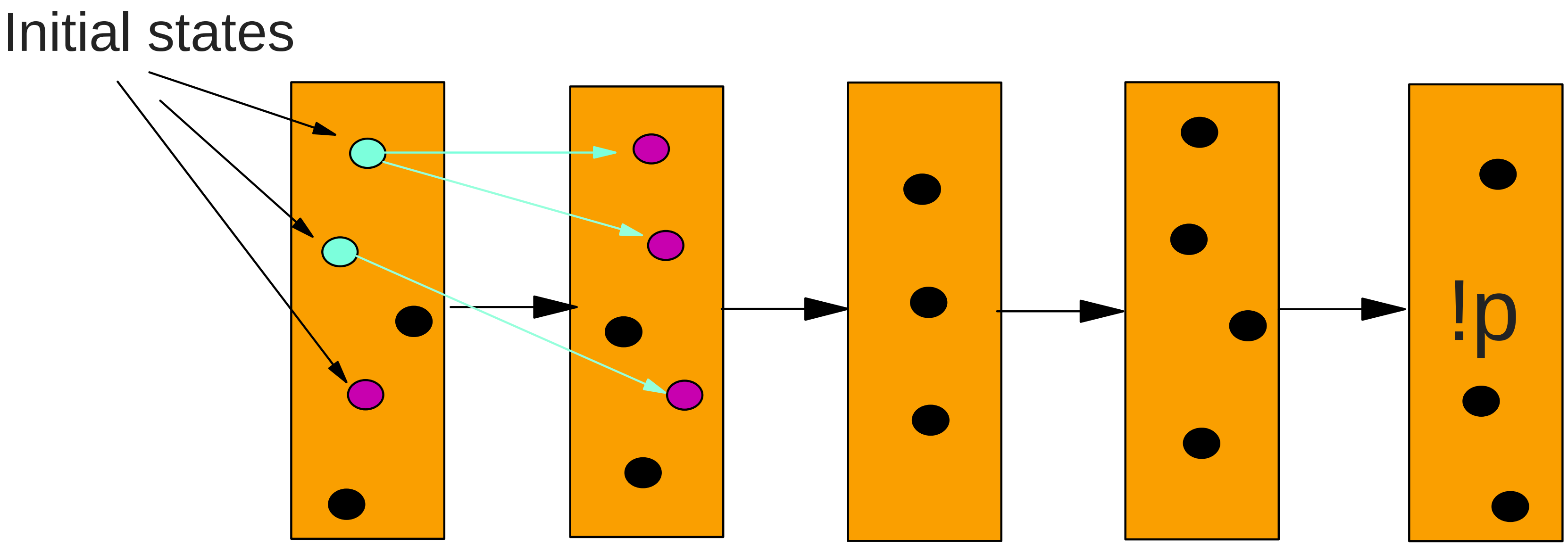


Reconstruction with Backtracking

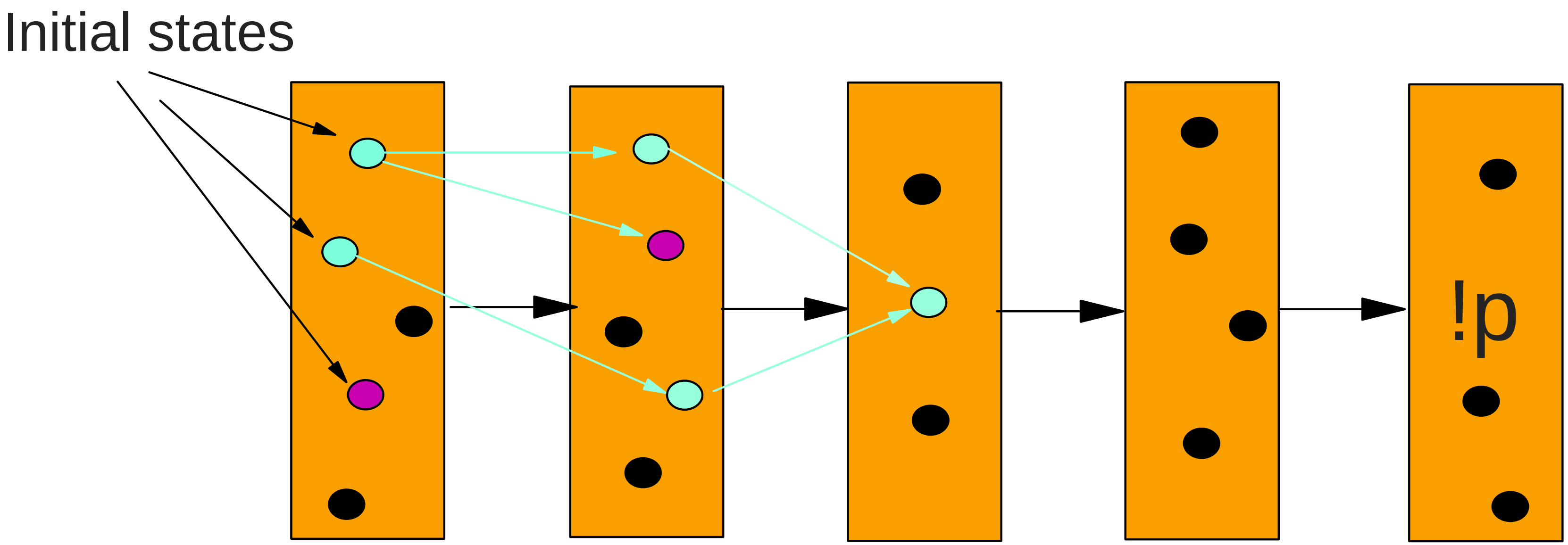


Only subset of the initial states are selected according to BDD criteria.

Reconstruction with Backtracking

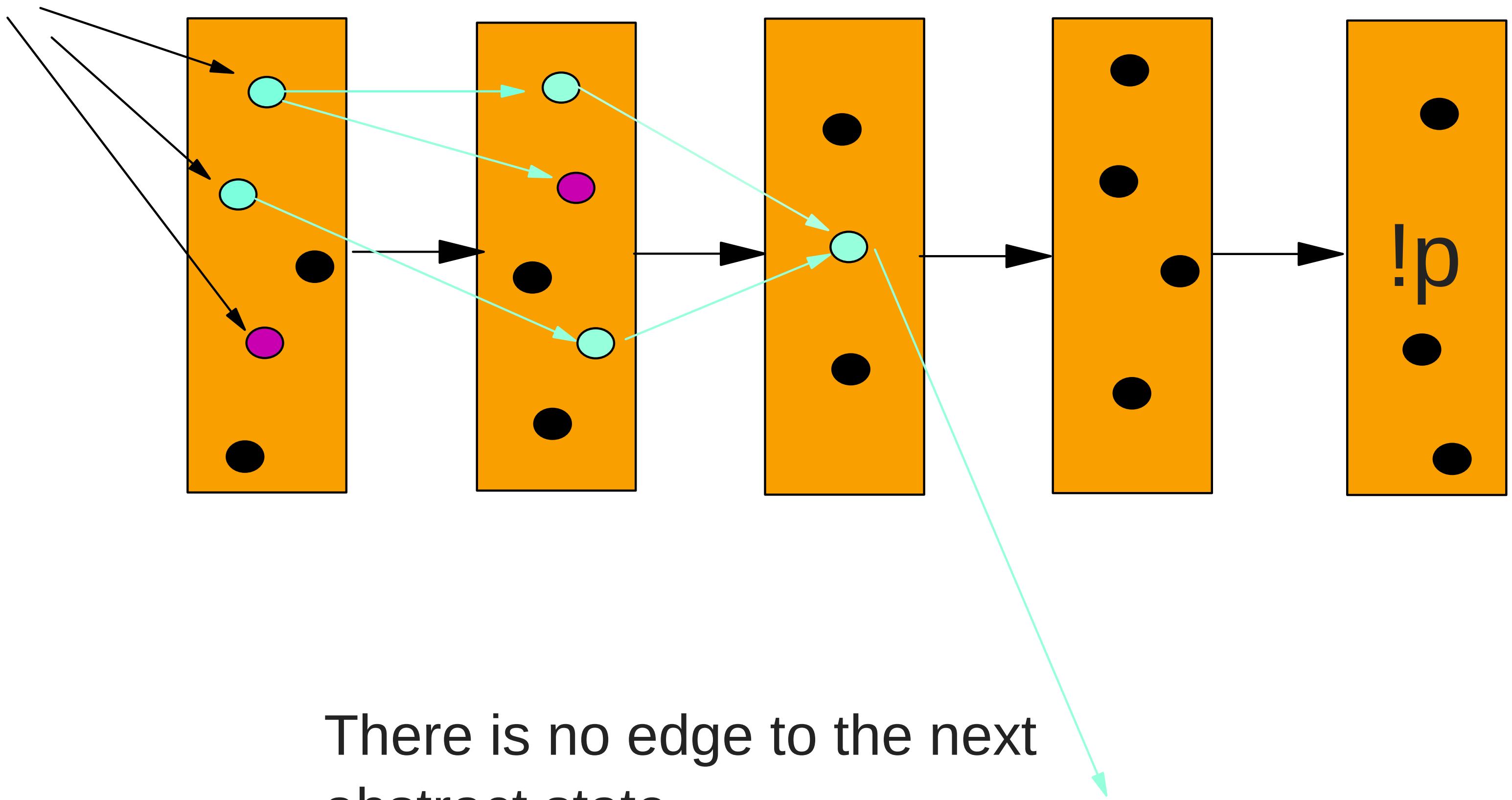


Reconstruction with Backtracking



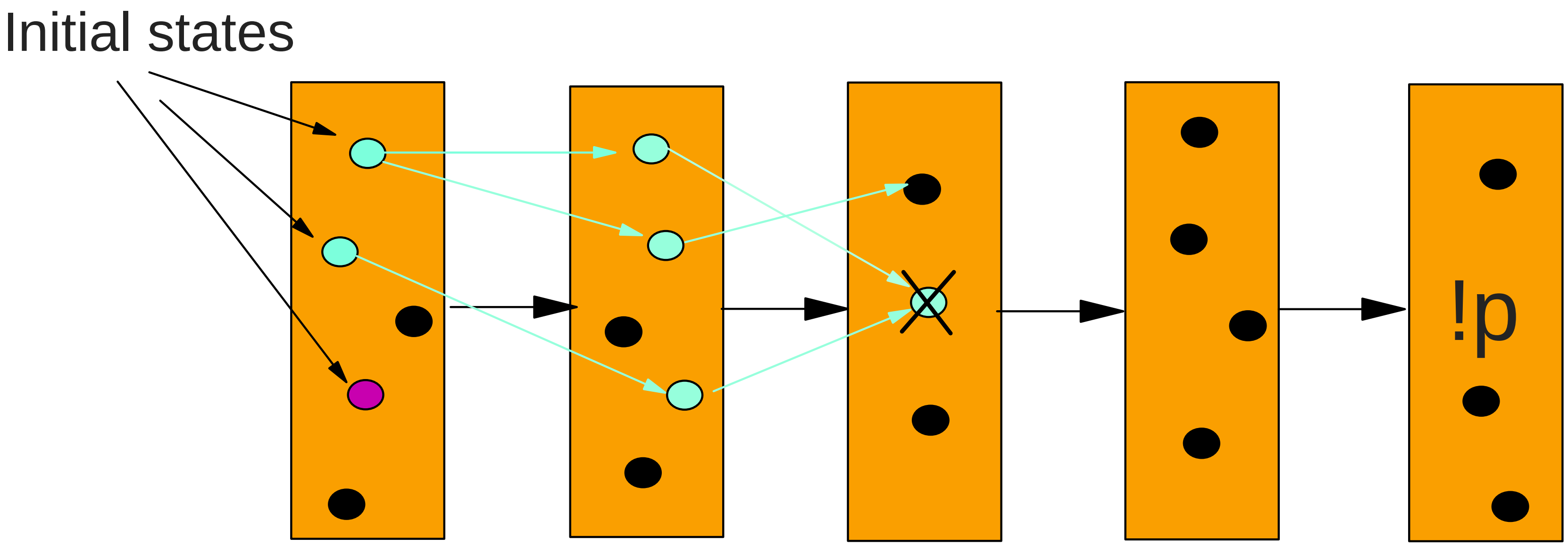
Reconstruction with Backtracking

Initial states

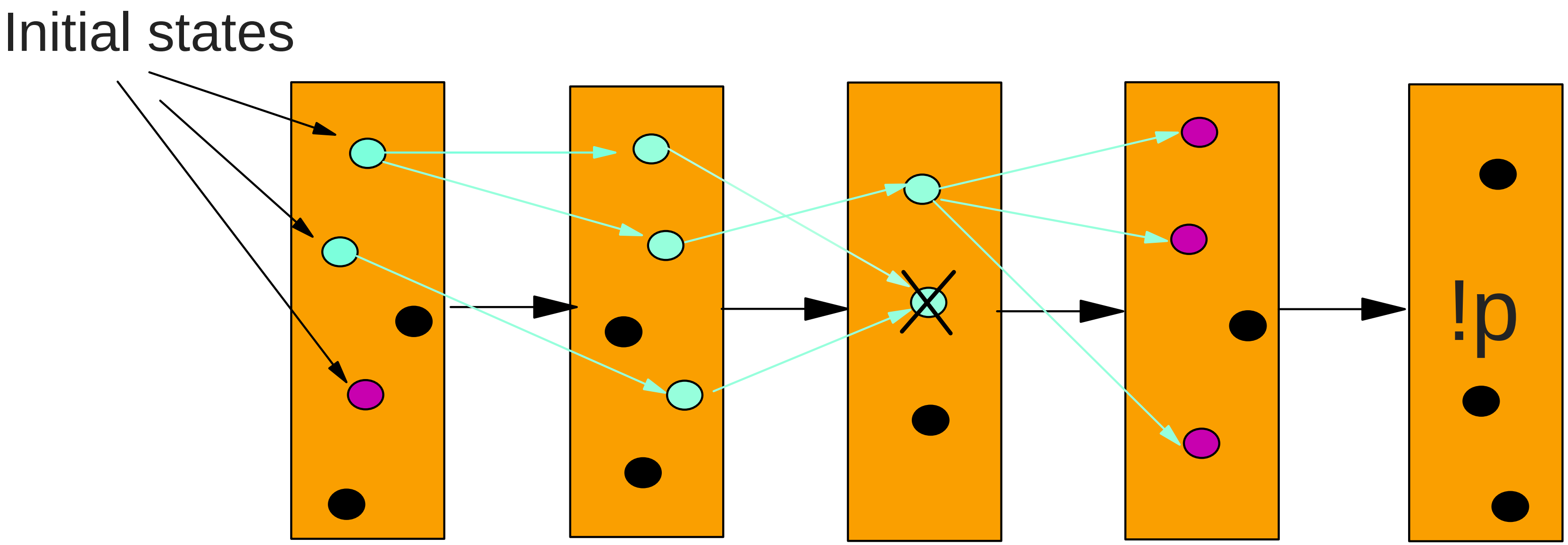


There is no edge to the next abstract state.

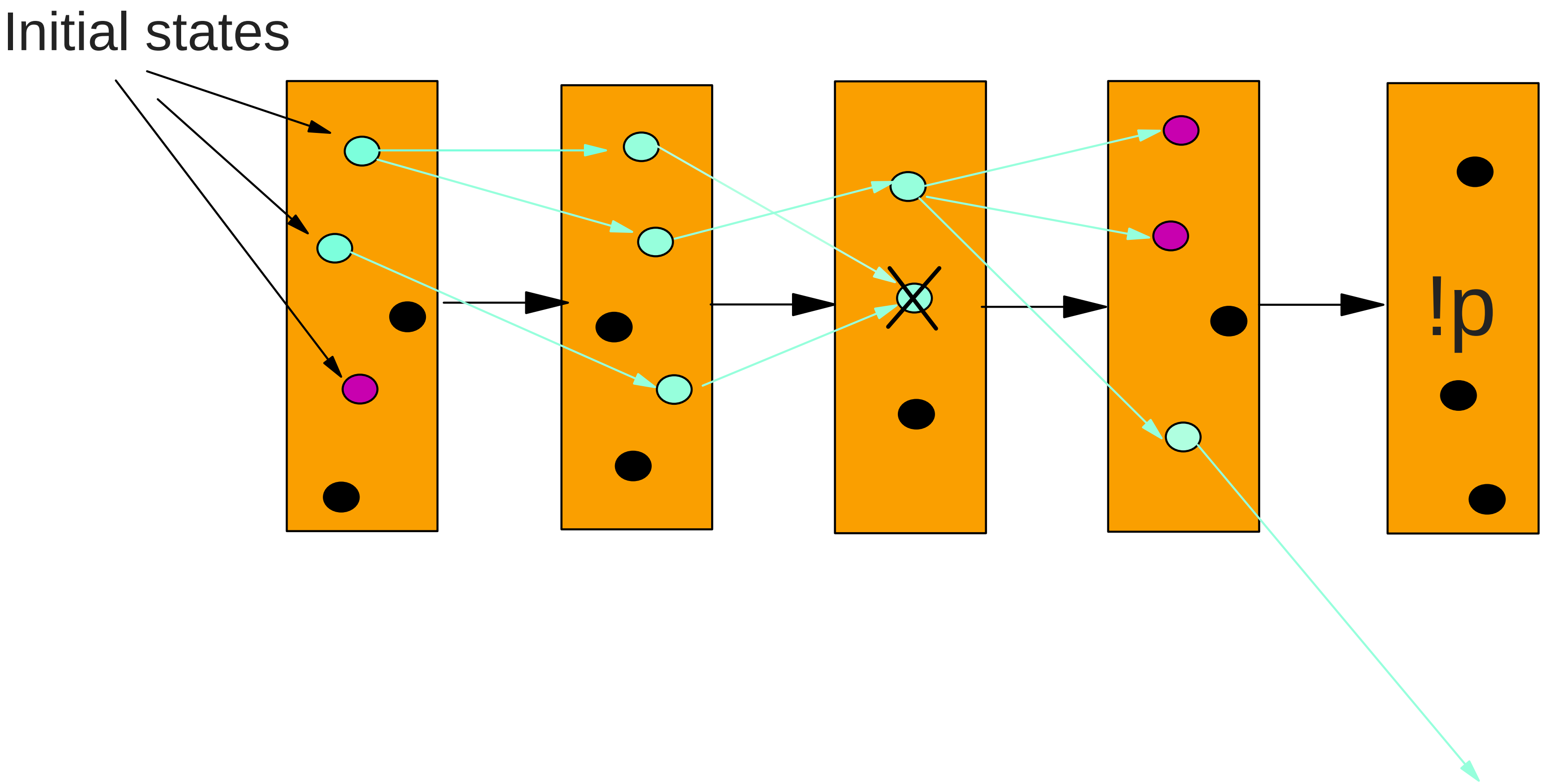
Reconstruction with Backtracking



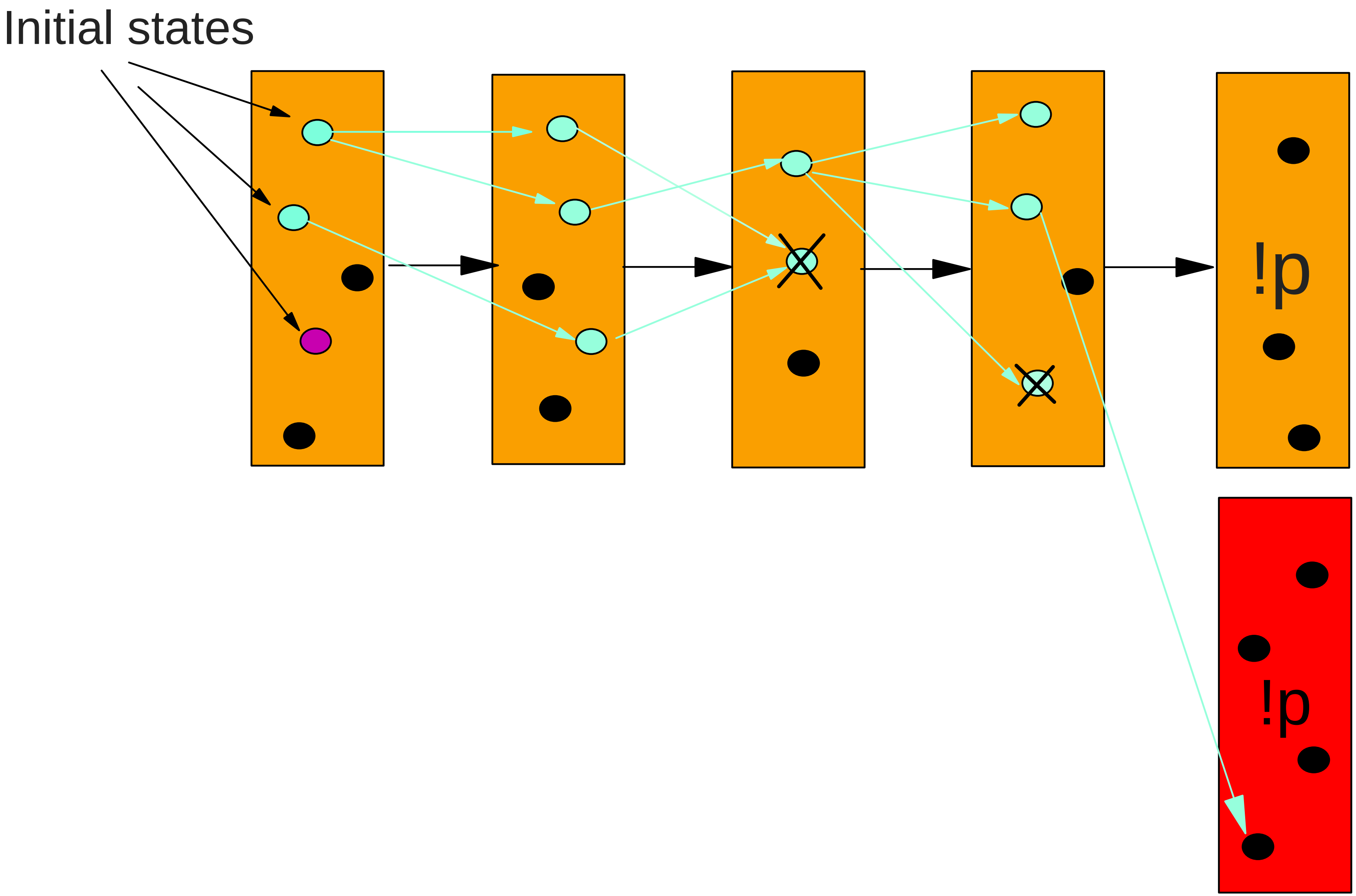
Reconstruction with Backtracking



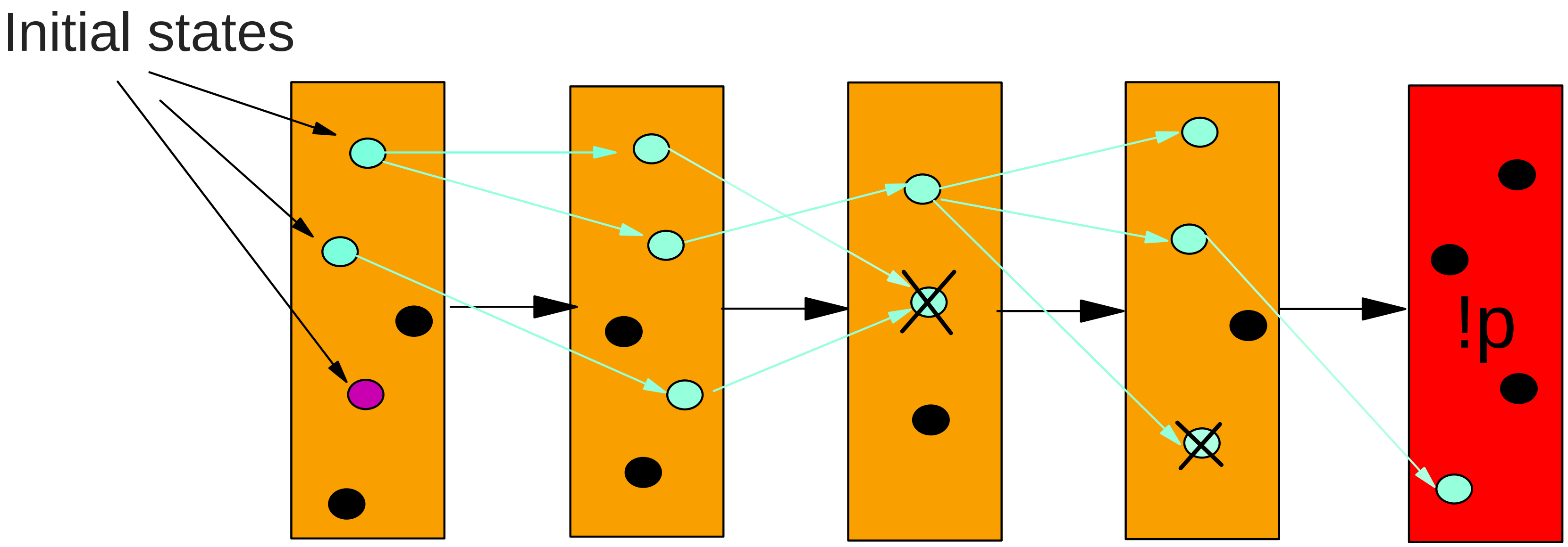
Reconstruction with Backtracking



Reconstruction with Backtracking



Reconstruction with Backtracking



Counter example is found

Results

Design Unit	Result	Number of State Vars	No Local	Clarke et al.	Layer+ Alg1	Layer+Alg2
Infiniband 1	passed	396(93)	Mem	Mem	54s/43M	137s/47M
Infiniband 2	passed	377(7)	Mem	* 0.95s/33M	4.32s/33M	4.19s/33M
Ethernet 1	passed	96(79)	1601s/87M	657s/189M	243s/88M	287s/88M
Ethernet 2	passed	156(36)	599s/92M	Mem	85s/99M	335s/93M
Queue CRM	passed	79(70)	148s/45M	75s/42M	34s/41M	28s/41M
CPU 1	passed	123(65)	Mem	14211s/185M	Mem	9.4s/31M
CPU 2	failed	105(35)	595s/62M	N/A	405s/50M	229s/50M
CPU 3	failed	167(66)	11943s/96M	Mem	28963s/192M	1096s/103M

* no refinement was needed