A model checking algorithm for CTL

Goal: $\underline{M}, s_0 \models \underline{\varphi}$

The basic labelling algorithm.

uses: $\perp$, $\neg$, $\wedge$, AF, EU, EX

(an adequate set of connectives)

Input A CTL model $M = (S, \to, L)$ (a labelled directed graph) & a CTL formula φ

Output: the set of states of M which satisfy φ.

① Rewrite φ in terms of $AF, EU, EX, \wedge, \neg, \perp$.

② Label the states of M with the subformulas of φ (starting from the smallest)

Suppose ψ is a subformula of φ; and all immediate subformulas of ψ have already been used.

If ψ is

$\perp$: then no states are labelled $\perp$

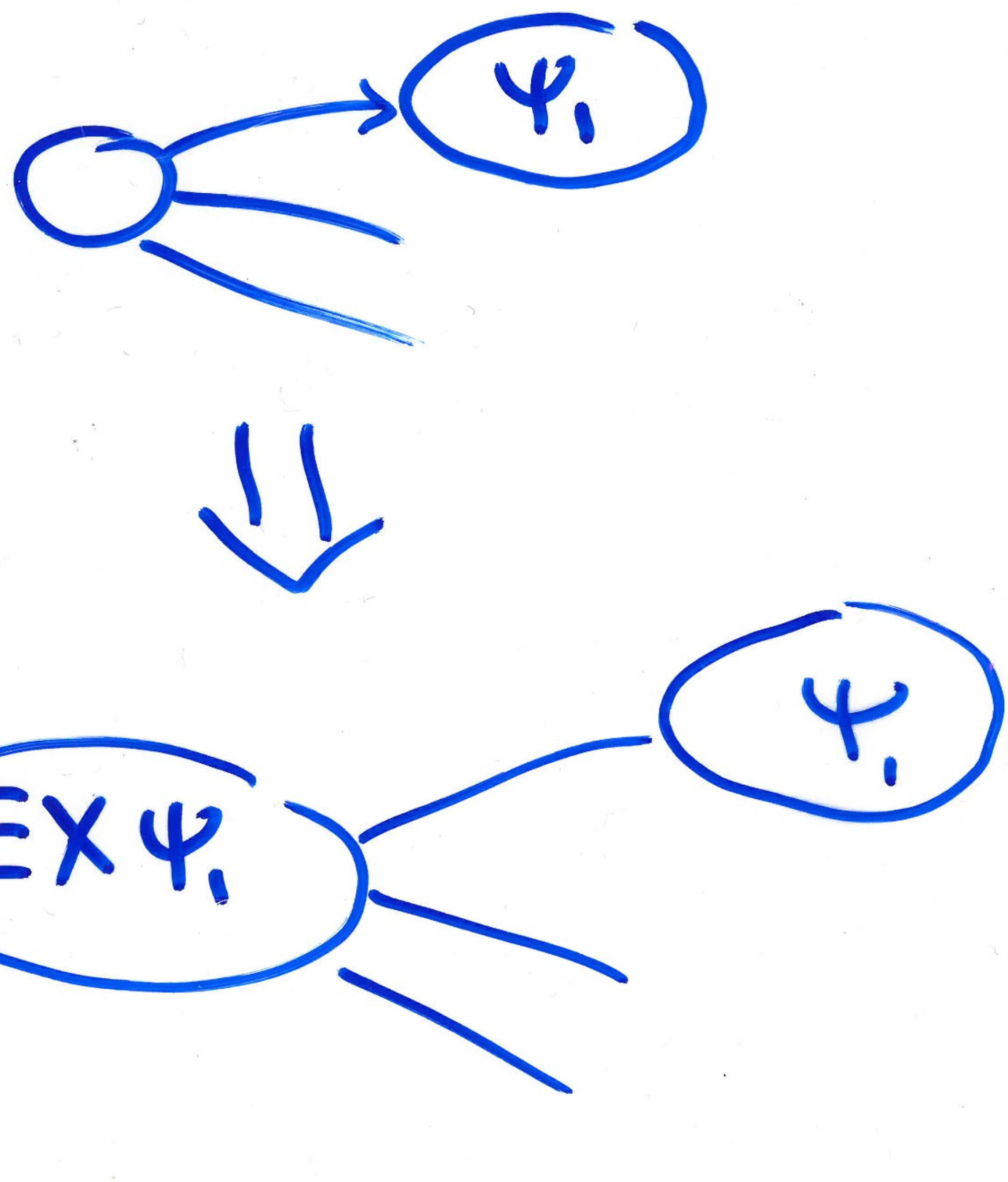
p : label s with p if $p \in L(s)$

$\psi_1 \wedge \psi_2$: label s with $\psi_1 \wedge \psi_2$ if s alredy has ψ_1, ψ_2 as its labels

$\neg\psi_1$: label s with $\neg\psi_1$ if s is <u>not</u> labelled ψ_1.

$EX\,\Psi_1$:

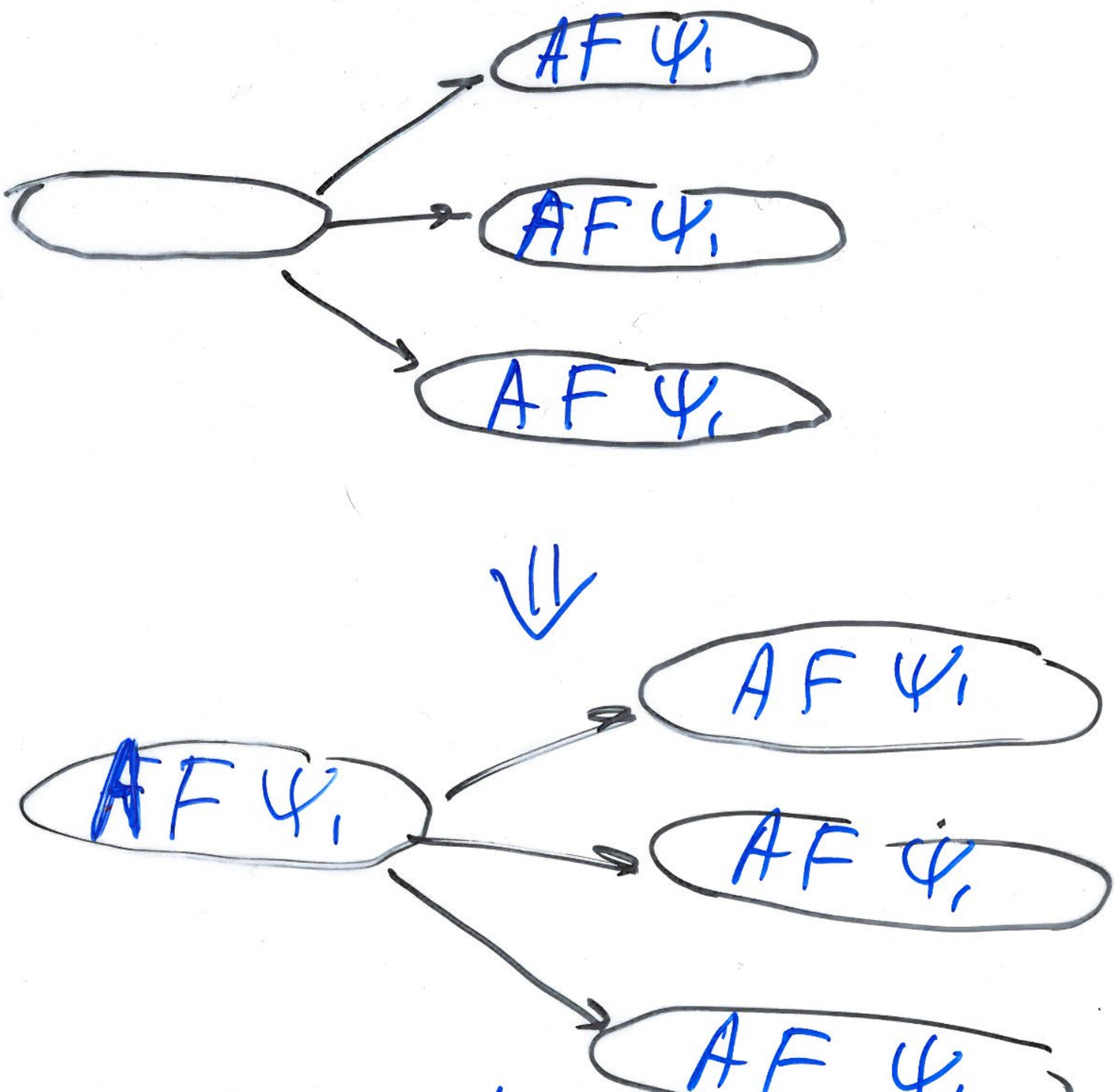
label any state with $EX\,\Psi_1$
if one of its successors
is labelled with Ψ_1



AF ψ_1 :

a) If any state is labelled ψ_1, label it with AF ψ_1

b) Repeat :

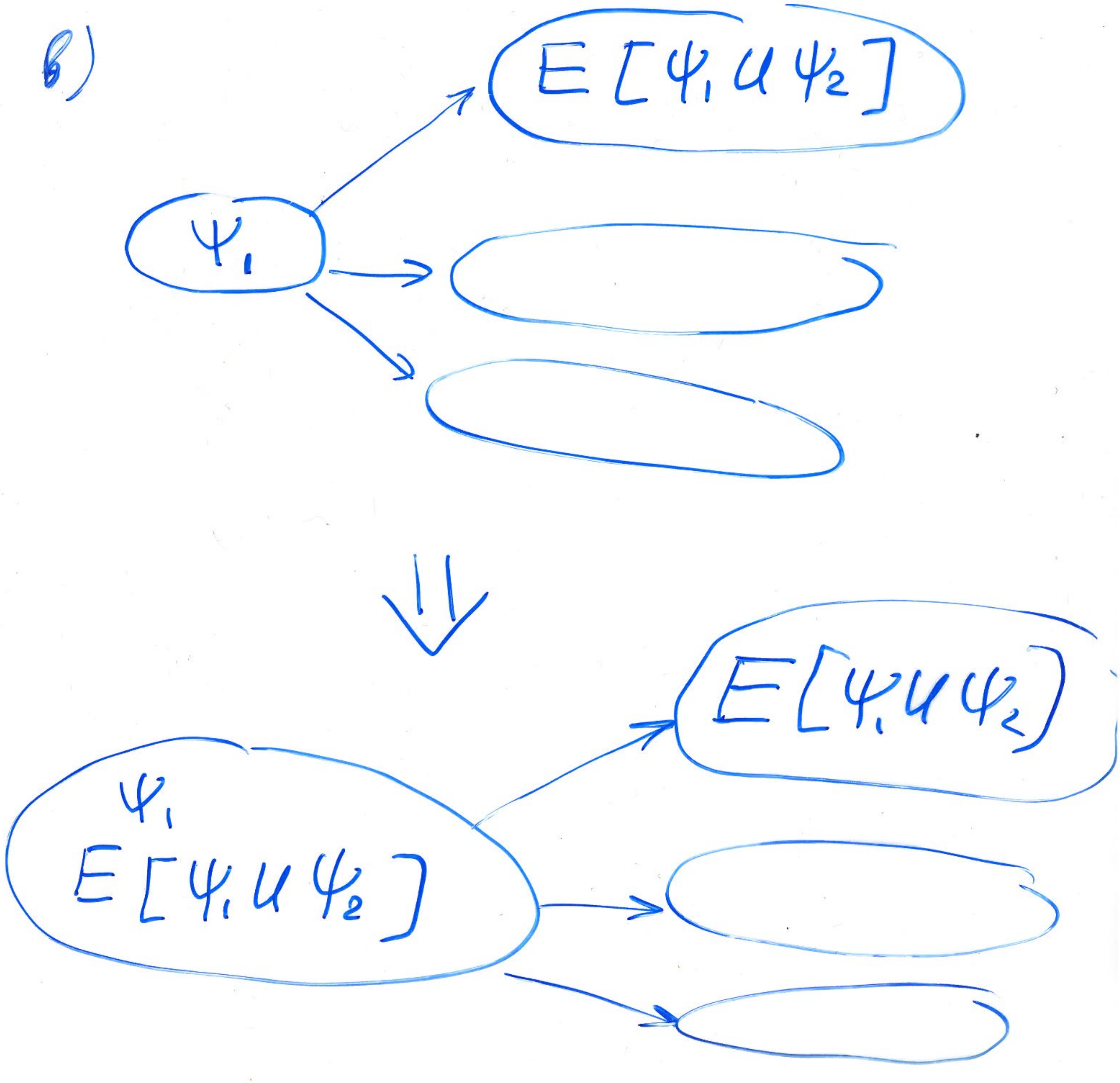


until no change.

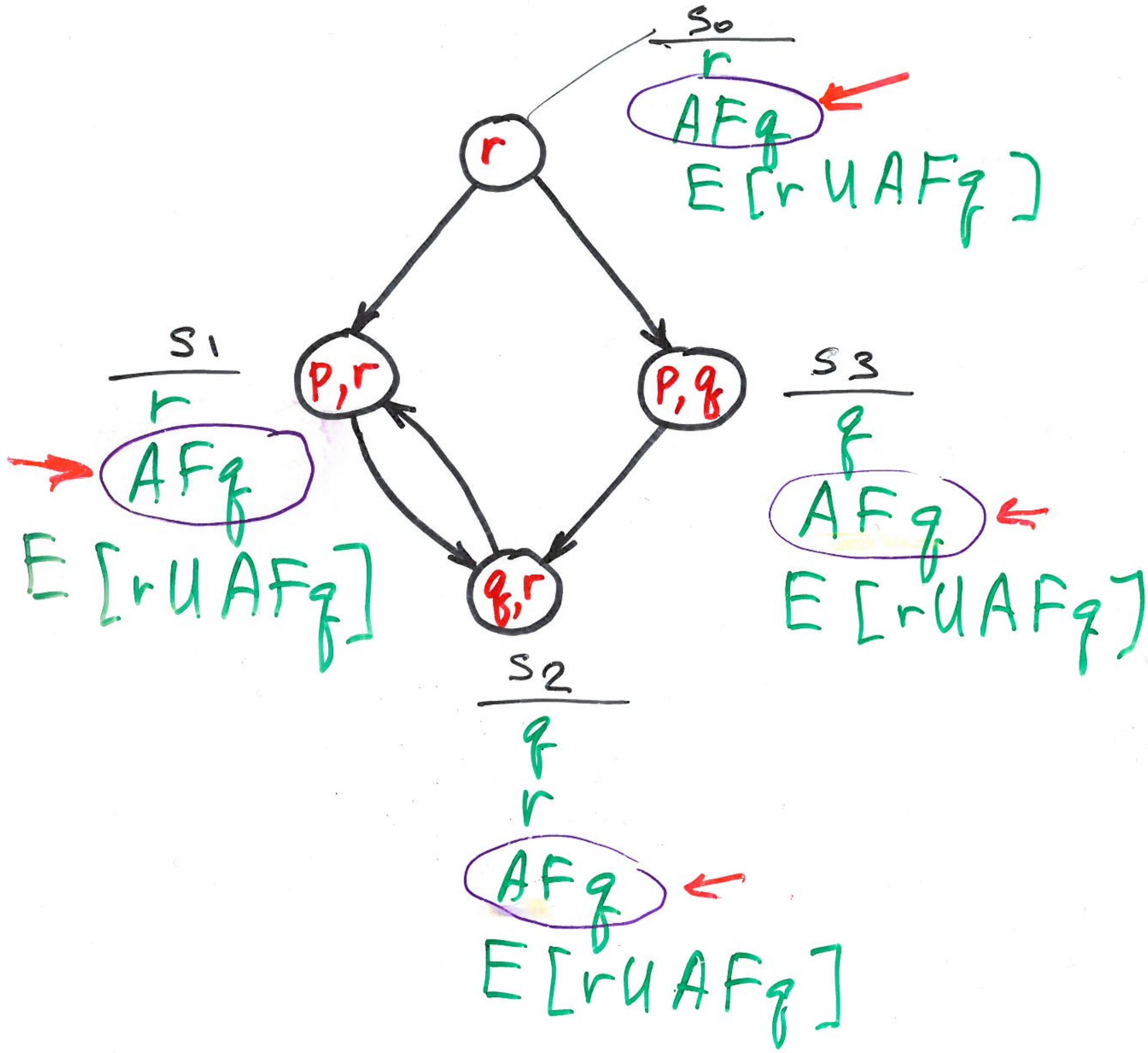
$E[\psi_1 \, U \, \psi_2]$

a) If any is labelled ψ_2, label it $E[\psi_1 \, U \, \psi_2]$

b)



$$M, s_0 \models E[r \, U \, \underbrace{AFq}]$$



Complexity of the basic labelling alg. (BLA)

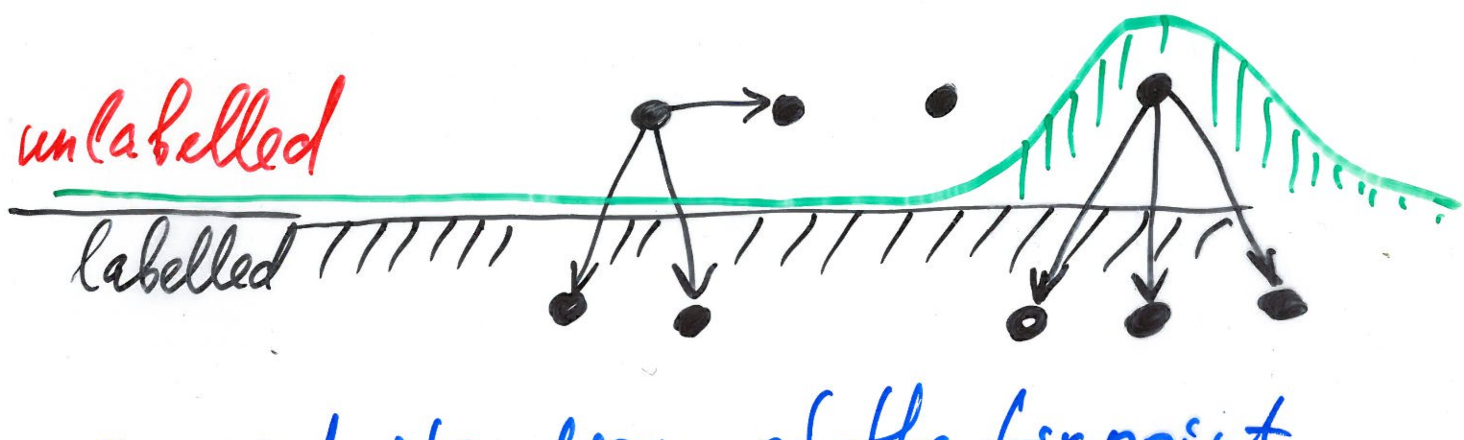
linear in the size of the formula
quadratic in the size of the model

$$O(f \cdot V \cdot (V+E))$$

f: # of connectives in φ

V: # of states

E: # of transitions



on each iteration of the fixpoint computation, visit all unlabelled vertices and all outgoing edges

In the worst case, $V+E$

How many iterations?

On each iteration, in the worst case, only one vertex is added.

So, V iterations.

We can deal with EG directly:

- Label all states $EG\,\psi$.
- If any state s is not labelled ψ, delete the label $EG\,\psi$

Repeat: delete $EG\,\psi$ if none of its successors is labelled $EG\,\psi$.

until no change

A more efficient variant

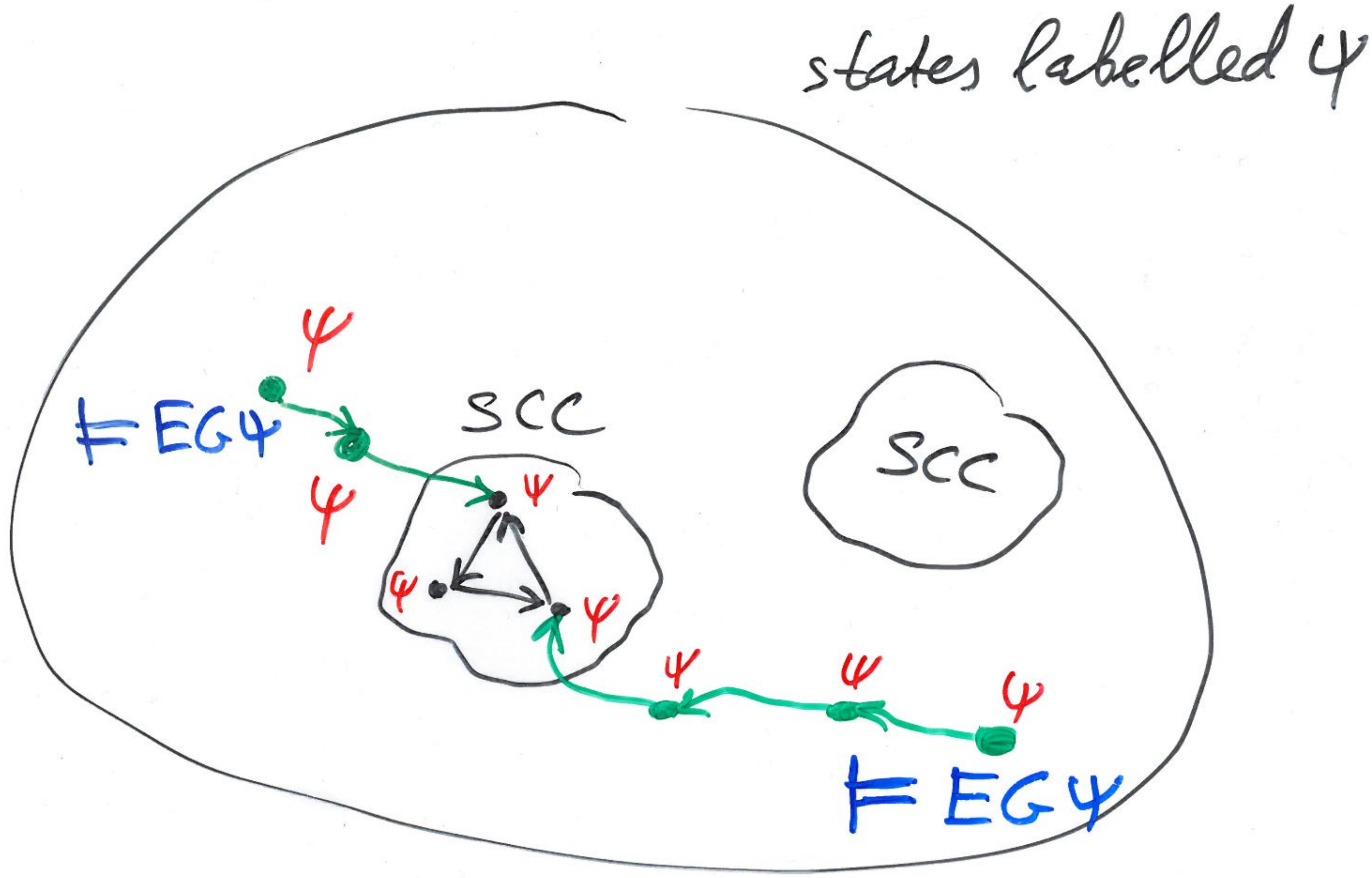
(use EX, EU, $\underline{EG}$)

better handling of $EG\underline{\psi}$.

- Restrict the graph to the states satisfying ψ
 (delete all other states & their transitions)

- Find maximal SCC
 (strongly connected components
 = maximal subgraphs where each state has a finite path to every other state)

- Use backwards breadth-first search to find any state that can reach a SCC.



For EX, EU use backwards Breadth-first search.

Complexity: $O(f \cdot (V+E))$

The state explosion problem

The algorithm is linear in the size of the system, but the system is exponential in the # of atoms

$\Rightarrow$ adding one atom doubles the complexity of verification!

To overcome:

- efficient data structures

OBDDs (Bryant)

Introduced to MC by Ken McMillan in his PhD thesis (1993)

- ...
- ....