COL:750

Foundations of Automatic Verification

Instructor: Priyanka Golia

Course Webpage 

https://priyanka-golia.github.io/teaching/COL-750/index.html

Model Checking using Interpolants

Inductive Invariants

$$\text{Post-image (Q)} = \{ s' \mid \exists s \in Q \,.\, T(s, s') \}$$

Inductive invariant (I_s) for $\forall \square p$

1. I_s must include the set of initial states, $I \subseteq I_s$
2. I_s must not include a state that is labeled with $\neg p$, $\forall s \in I_s, s \vDash p$
3. I_s must be closed under transition relation, post-image(I_s) $\subseteq I_s$ holds.

If there exists a inductive invariant for $\forall \square P$, then $M \vDash \forall \square p$

Model Checking using Interpolants

Can you use interplants to compute inductive invariants?

1. Constructs an over-approximation of the reachable states
2. Terminates when it finds an inductive invariant or a counterexample

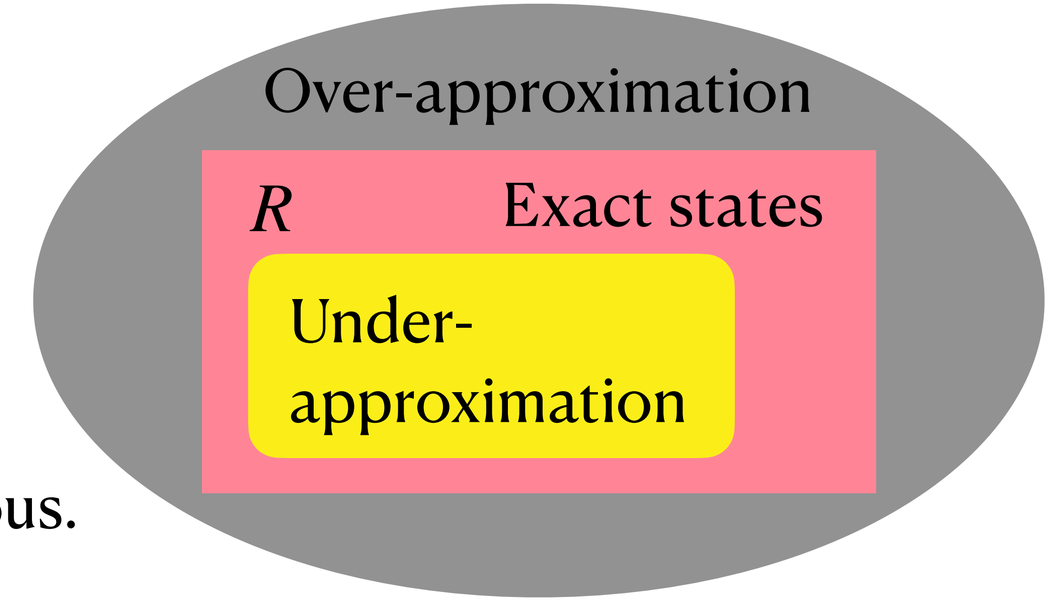
Actual reachable set: R

Over-approximation (O_p): $R \rightarrow O_p$

1. Proofs on over-approximation holds.
2. Counterexample can be spurious.

Under-approximation (U_p): $U_p \rightarrow R$

1. Proofs on over-approximation can be spurious.
2. Counterexample holds



Model Checking using Interpolants

General idea:

1. Perform BMC

2. If BMC is UNSAT:

Iteratively compute and refine an over-approximation of states reachable in K steps.

Compute Interpolant as over-approximation.
If interpolant is inductive
Return True.
else
use interpolant to over-approximate.

3. If BMC is SAT:

Check if over-approximation is same as initial states
otherwise increase K.

procedure *CraigReachability*(model M, $p \in AP$)
  if $S_0 \wedge \neg p$ is SAT **return** "$M \not\models \mathbf{AG}\, p$";
  $k := 1$;
  $Q := S_0$;
  while *true* **do**
    $A := Q(s_0) \wedge R(s_0, s_1)$;
    $B := \bigwedge_{i=1}^{k-1} R(s_i, s_{i+1}) \wedge \bigvee_{i=1}^{k} \neg p(s_i)$;
    if $A \wedge B$ is SAT **then**
      if $Q = S_0$ **then return** "$M \not\models \mathbf{AG}\, p$";
      Increase k
      $Q := S_0$
    else
      compute interpolant I for A and B
      if $I \subseteq Q$ **then return** "$M \models \mathbf{AG}\, p$";
      $Q := Q \cup I$
    end if
  end while
end procedure

Inductive Trace

An Inductive trace of a transition system T is a sequence of formula $[F_o, \ldots, F_K]$ such that:

1. $I \rightarrow F_o$
2. $\forall i \in [o, \ldots, K], F_i(s) \wedge T(s, s') \rightarrow F_{i+1}(s')$

Example: state variables $\{a, b\}$

initial condition $I = \neg a \wedge \neg b$

transition function. $next\ a = b \quad next\ b = a$

Checking for property $\forall \square \neg b$

Reachability to a state with b

$Let\ F_o = I = \neg a \wedge \neg b \qquad (\neg a_o \wedge \neg b_o) \wedge (a_1 \leftrightarrow b_o) \wedge (b_1 \leftrightarrow a_o) \wedge b_1$

A B

UNSAT

Interpolant $\neg b_1 \qquad F_o \wedge T(s_o, s_1) \rightarrow \neg b_1 \qquad F_1 = \neg b$

Inductive Trace

An Inductive trace of a transition system T is a sequence of formula $[F_o, \ldots, F_K]$ such that:

$I \rightarrow F_o$

$\forall i \in [o, \ldots, k-1], F_i(s) \wedge T(s, s') \rightarrow F_{i+1}(s')$

A Trace is *Good* iff $\forall i, F_i \rightarrow \neg Bad$ For all F_i, property doesn't hold True!!!

A Trace is *Monotone* iff $\forall i, F_i \subseteq F_{i+1}$

Monotonicity ensures that as time progresses, we do not "forget" any reachable state
It aligns with the notion that reachable states can only grow (never shrink) as time increases

A Trace is *Closed* iff $\exists 1 \leq i \leq K, F_i \rightarrow (F_o \vee \ldots \vee F_{i-1})$

Each F_i is called Frames

A transition system T is called SAFE if and only if it admits a good, monotone, closed trace.

IC3 : Incremental Construction of Inductive Clauses for Indubitable Correctness.

The goal is to find an inductive invariant

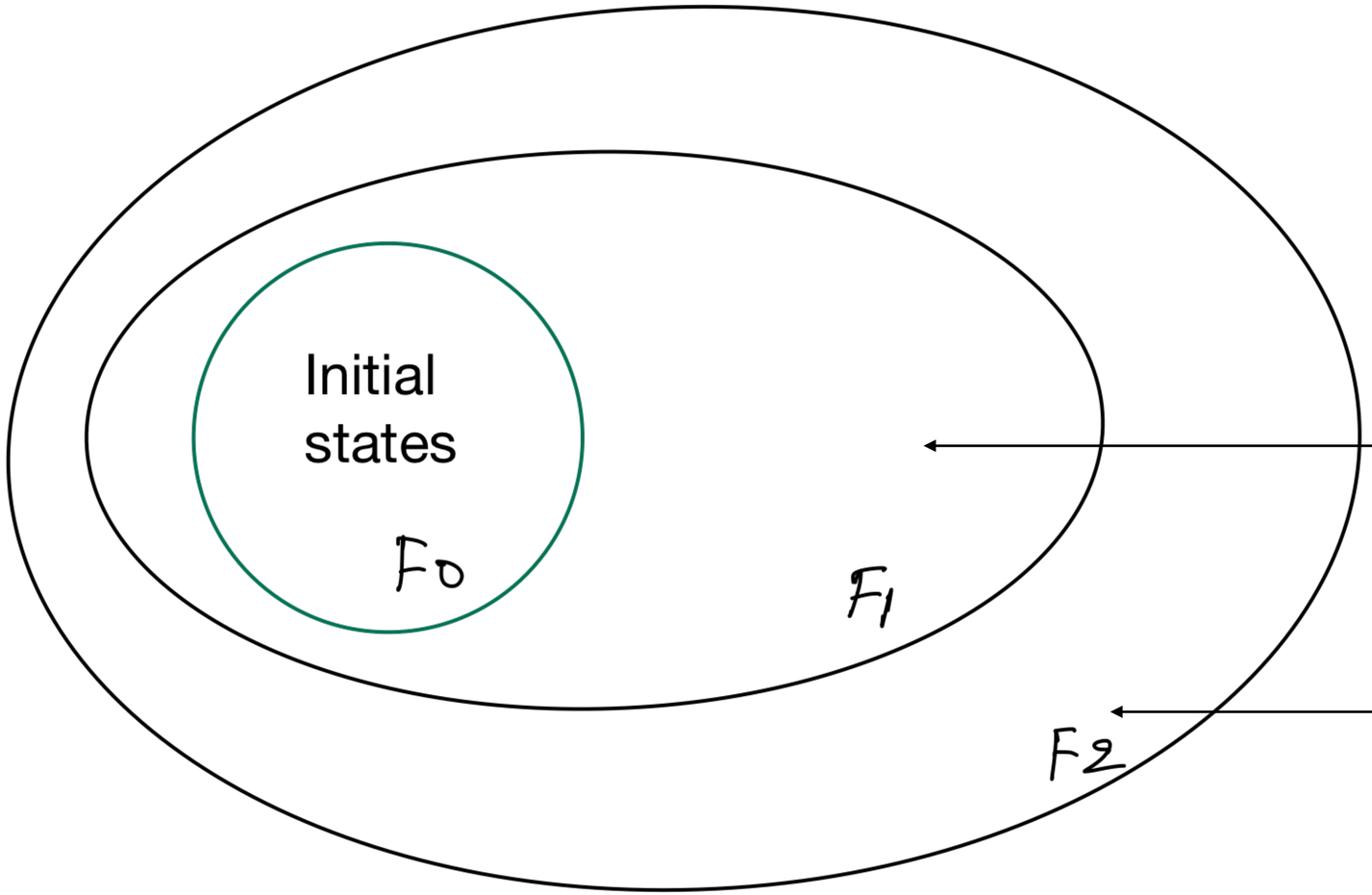
By learning inductive facts incrementally

An Inductive trace of a transition system T is a sequence of formula $[F_o, \ldots, F_K]$ such that:

1. $I \rightarrow F_o$
2. $\forall i \in [o, \ldots, K], F_i(s) \wedge T(s, s') \rightarrow F_{i+1}(s')$

A trace called clausal if every F_i is in CNF.

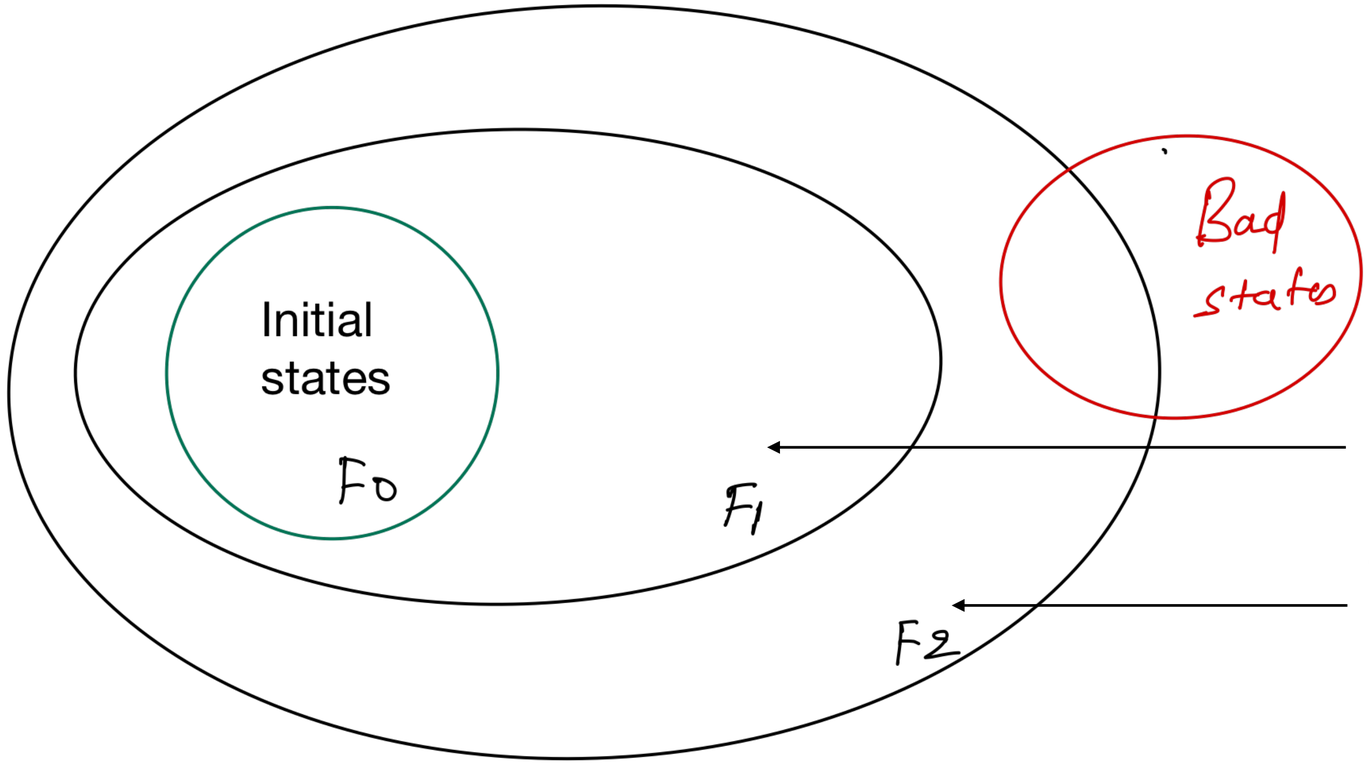
IC3 : Incremental Construction of Inductive Clauses for Indubitable Correctness.



Overapproximation of states reachable from F_o in 1 transition.

Overapproximation of states reachable from F_1 in 1 transition.

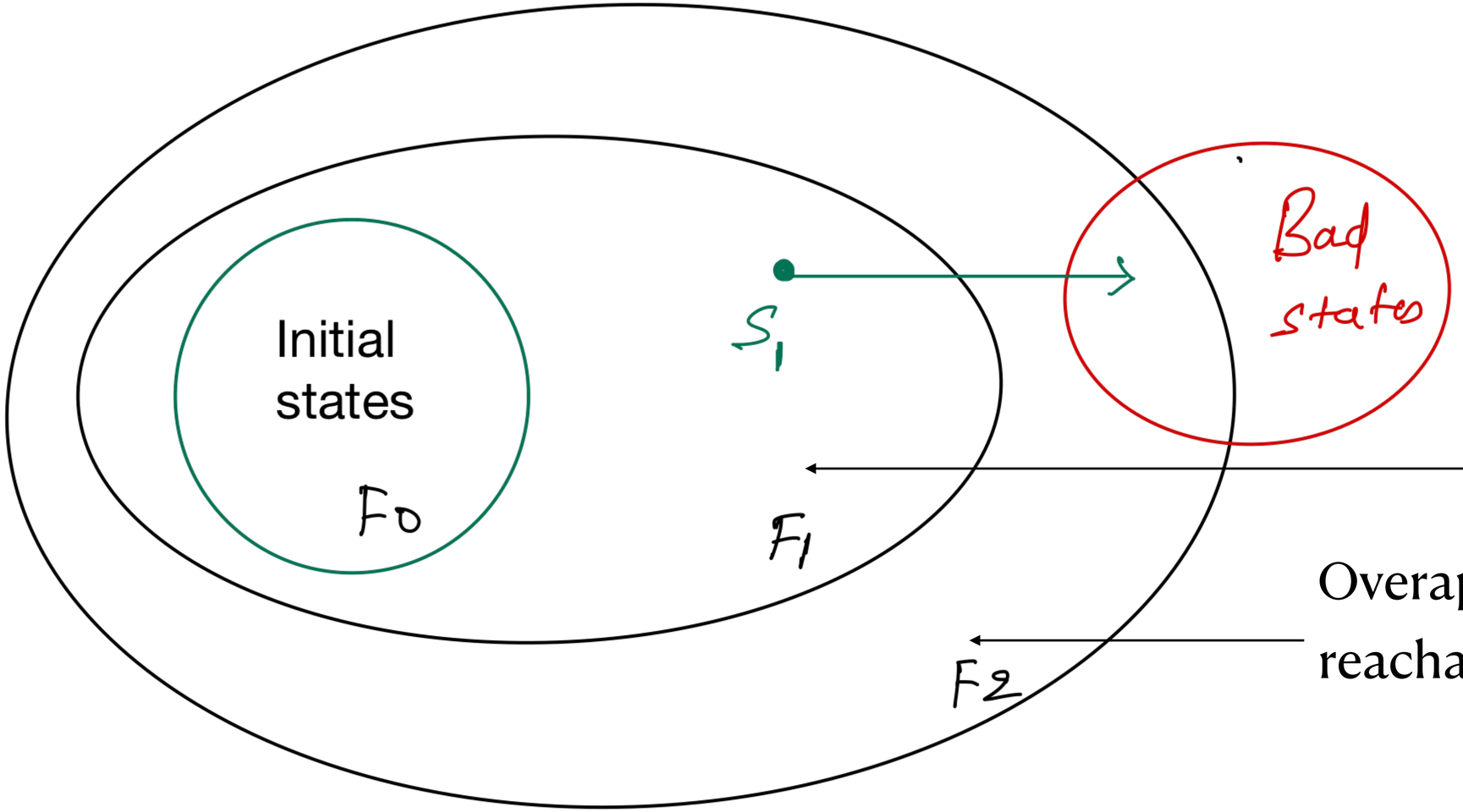
IC3 : Incremental Construction of Inductive Clauses for Indubitable Correctness.



Overapproximation of states reachable from F_o in 1 transition.

Overapproximation of states reachable from F_1 in 1 transition.

IC3 : Incremental Construction of Inductive Clauses for Indubitable Correctness.



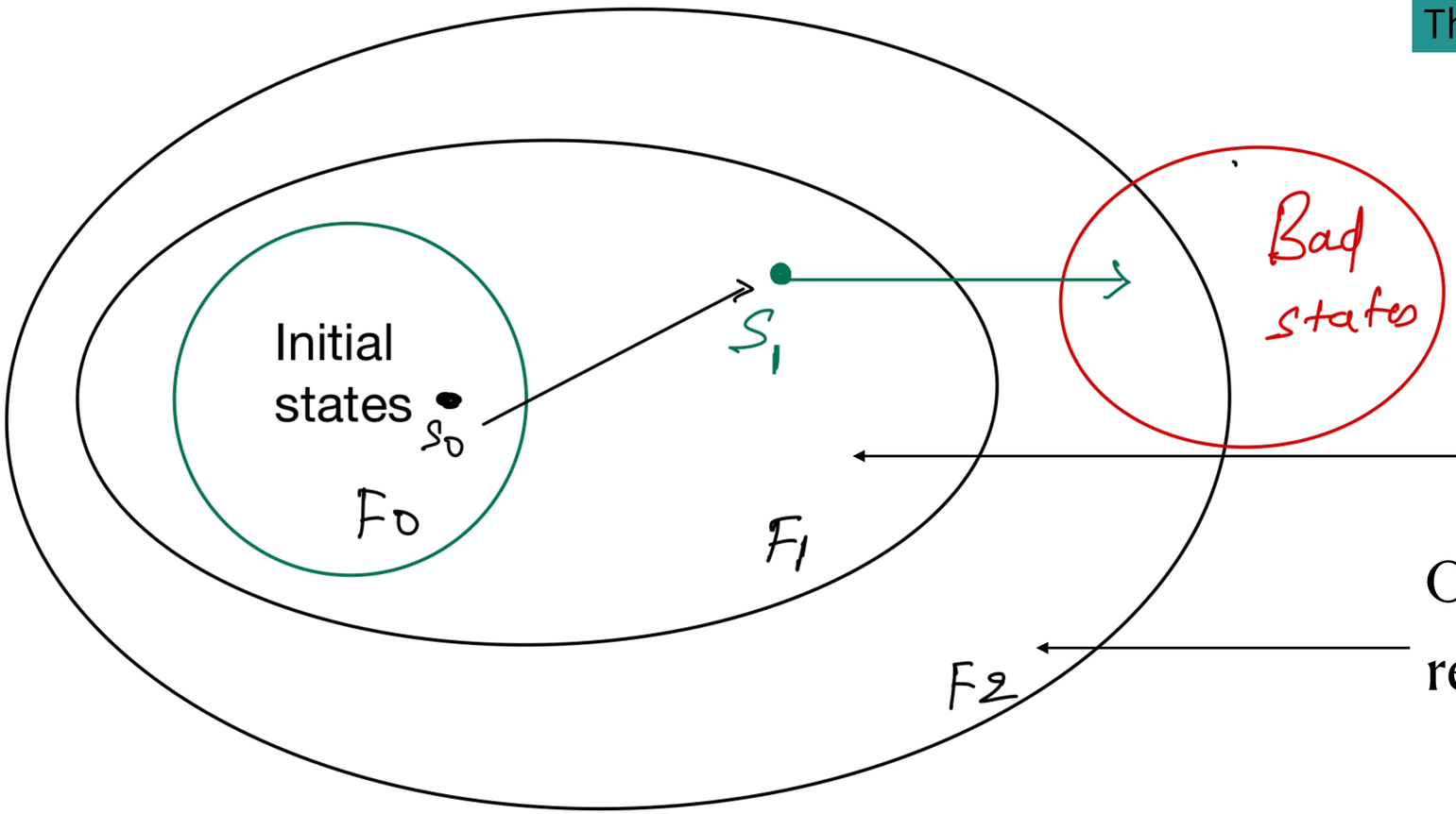
Overapproximation of states reachable from F_o in 1 transition.

Overapproximation of states reachable from F_1 in 1 transition.

IC3 : Incremental Construction of Inductive Clauses for Indubitable Correctness.

If $s_o -> s_1 -> Badstate$

Then found a counter example!!

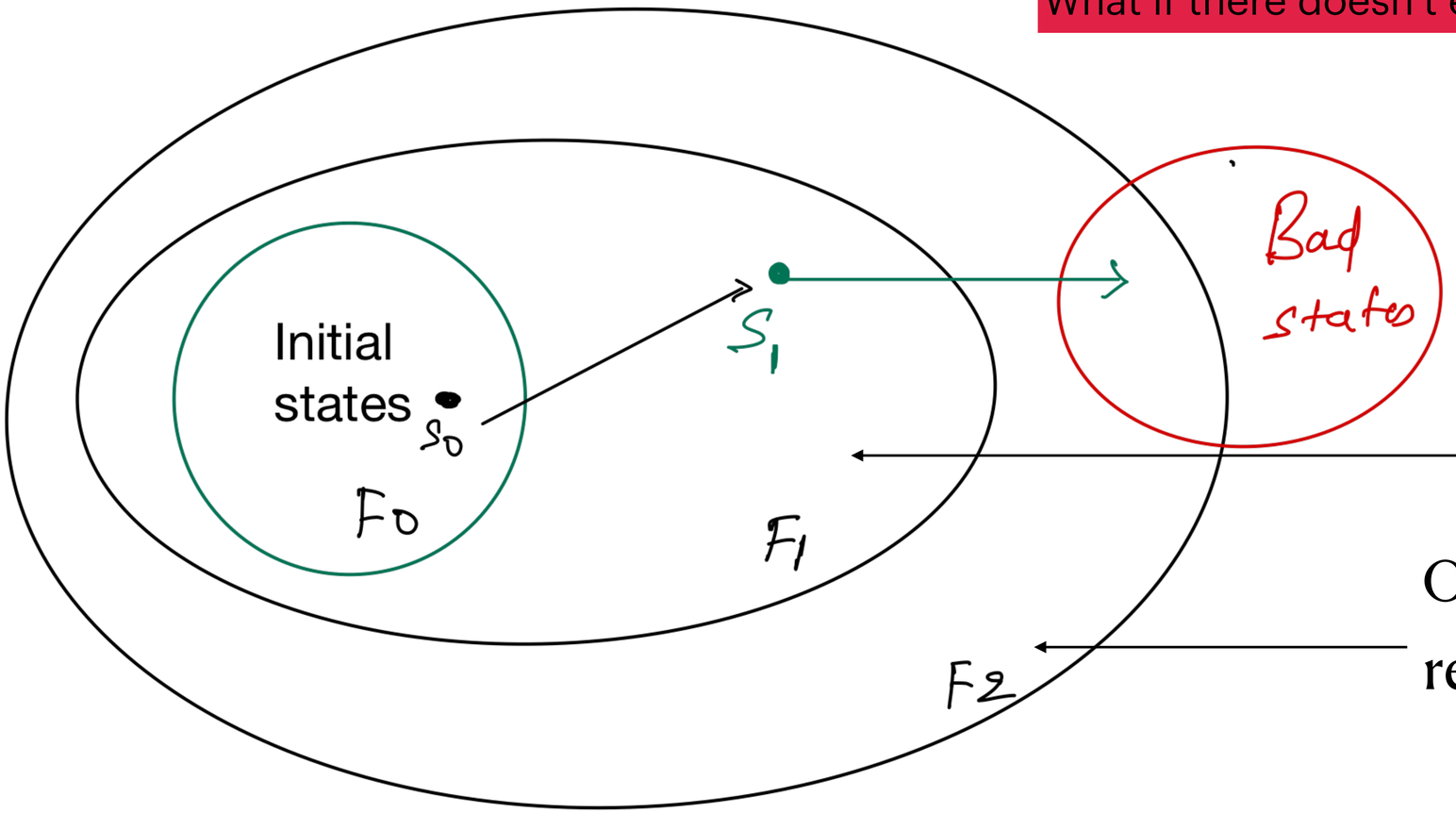


Overapproximation of states reachable from F_o in 1 transition.

Overapproximation of states reachable from F_1 in 1 transition.

IC3 : Incremental Construction of Inductive Clauses for Indubitable Correctness.

What if there doesn't exist a transition from $s_o -> s_1$?

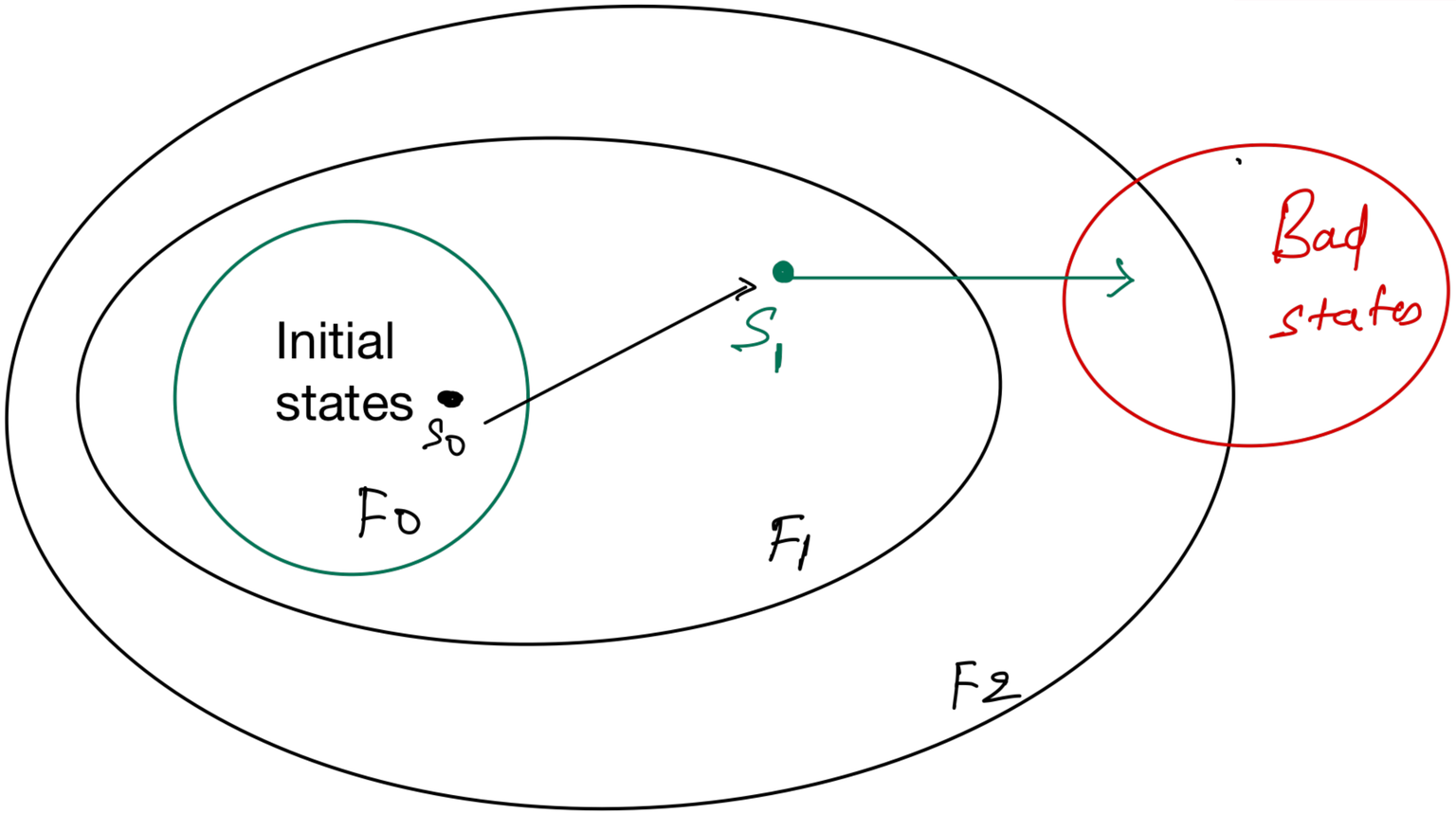


Overapproximation of states reachable from F_o in 1 transition.

Overapproximation of states reachable from F_1 in 1 transition.

IC3 : Incremental Construction of Inductive Clauses for Indubitable Correctness.

What if there doesn't exist a transition from $s_o -> s_1$?



We should block s_1 from F_1:

$$F_1 = F_1 \wedge \neg(\bigwedge_{\forall v_i \in s_1} v_i)$$

Clause!!

$$F_1 = F_1 \wedge (\bigvee_{\forall v_i \in s_1} \neg v_i)$$

IC3 : Incremental Construction of Inductive Clauses for Indubitable Correctness.

Example: state variables $\{a, b\}$

initial condition $I = \neg a \wedge \neg b$

transition function. $next\ a = b \quad next\ b = a$

Checking for property $\forall \square \neg b$

Reachability to a state with b

$Let\ F_o = I = \neg a \wedge \neg b \qquad \underbrace{(\neg a_o \wedge \neg b_o) \wedge (a_1 \leftrightarrow b_o) \wedge (b_1 \leftrightarrow a_o)}_{\text{A}} \wedge \underbrace{b_1}_{\text{B}}$ UNSAT

Interpolant $\neg b_1 \qquad F_o \wedge T(s_o, s_1) \rightarrow \neg b_1 \qquad F_1 = \neg b$

$F_1 \wedge T(s_1, s_2) \wedge b_2 \qquad \neg b_1 \wedge (a_2 \leftrightarrow b_1) \wedge (b_2 \leftrightarrow a_1) \wedge b_2$ SAT

From F_1 in one transition bad state is reachable!!! $\sigma : \langle a_1 = 1, b_1 = 0, a_2 = 0, b_2 = 1 \rangle$

IC3 : Incremental Construction of Inductive Clauses for Indubitable Correctness.

Example: state variables $\{a, b\}$

initial condition $I = \neg a \wedge \neg b$

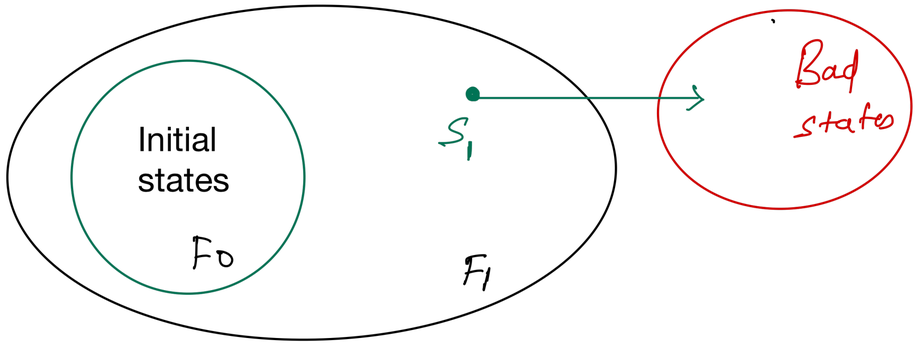
transition function. $next\ a = b \quad next\ b = a$

Checking for property $\forall \square \neg b$

Reachability to a state with b

$F_o = \neg a_o \wedge \neg b_o$ $\quad F_1 = \neg b_1$ $\quad \neg b_1 \wedge (a_2 \leftrightarrow b_1) \wedge (b_2 \leftrightarrow a_1) \wedge b_2$

$\sigma : \langle a_1 = 1, b_1 = 0, a_2 = 0, b_2 = 1 \rangle$



$s_1 = a_1 \wedge \neg b_1$

We extract state s_1 from $\sigma \models F_1 \wedge T(s_1, s_2) \wedge b$

We need to check if s_1 is reachable from F_o

IC3 : Incremental Construction of Inductive Clauses for Indubitable Correctness.

Example: state variables $\{a, b\}$

initial condition $I = \neg a \wedge \neg b$

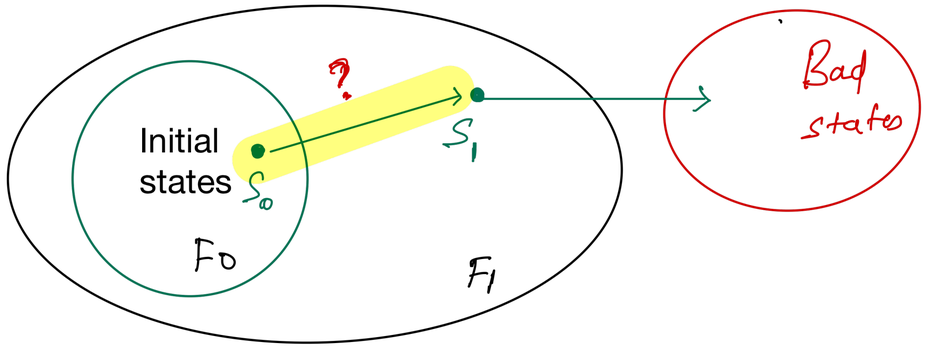
transition function. $next\ a = b \quad next\ b = a$

Checking for property $\forall \square \neg b$

Reachability to a state with b

$F_o = \neg a_o \wedge \neg b_o \qquad F_1^{old} = \neg b_1 \qquad s_1 = a_1 \wedge \neg b_1$

We need to check if s_1 is reachable from F_o



$$\neg a_o \wedge \neg b_o \wedge (a_1 \leftrightarrow b_o) \wedge (b_1 \leftrightarrow a_o) \wedge (a_1 \wedge \neg b_1)$$

UNSAT

s_1 is not reachable from F_o

We need to block s_1 from F_1

IC3 : Incremental Construction of Inductive Clauses for Indubitable Correctness.

Example: state variables $\{a, b\}$

initial condition $I = \neg a \wedge \neg b$

transition function. $next\ a = b \quad next\ b = a$

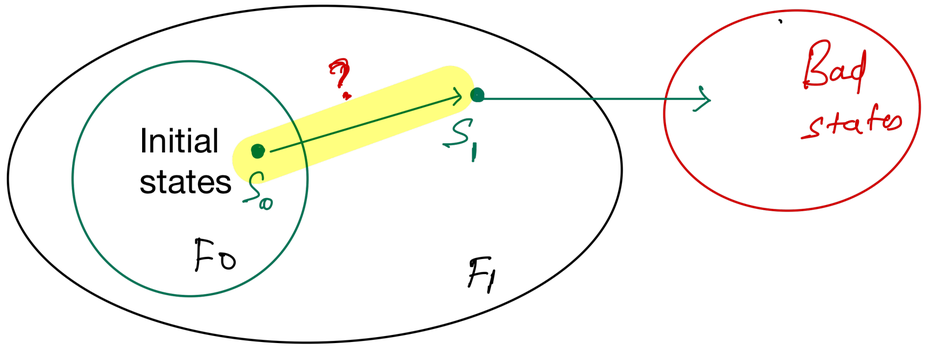
Checking for property $\forall \square \neg b$

Reachability to a state with b

$F_o = \neg a_o \wedge \neg b_o \qquad F_1^{old} = \neg b_1 \qquad s_1 = a_1 \wedge \neg b_1$

s_1 is not reachable from F_o

We need to block s_1 from F_1



$$F_1 = F_1^{old} \wedge \neg(a_1 \wedge \neg b_1)$$

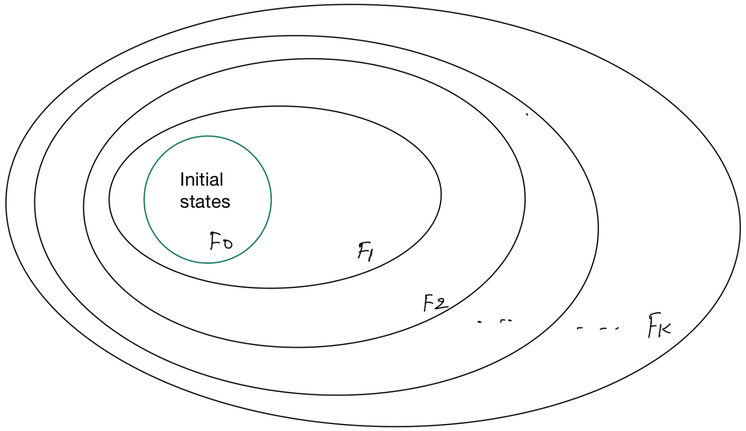
$$F_1 = F_1^{old} \wedge (\neg a_1 \vee b_1)$$

$$F_1 = \neg b_1 \wedge (\neg a_1 \vee b_1)$$

BlockClause c

IC3 : Incremental Construction of Inductive Clauses for Indubitable Correctness.

When to stop ?

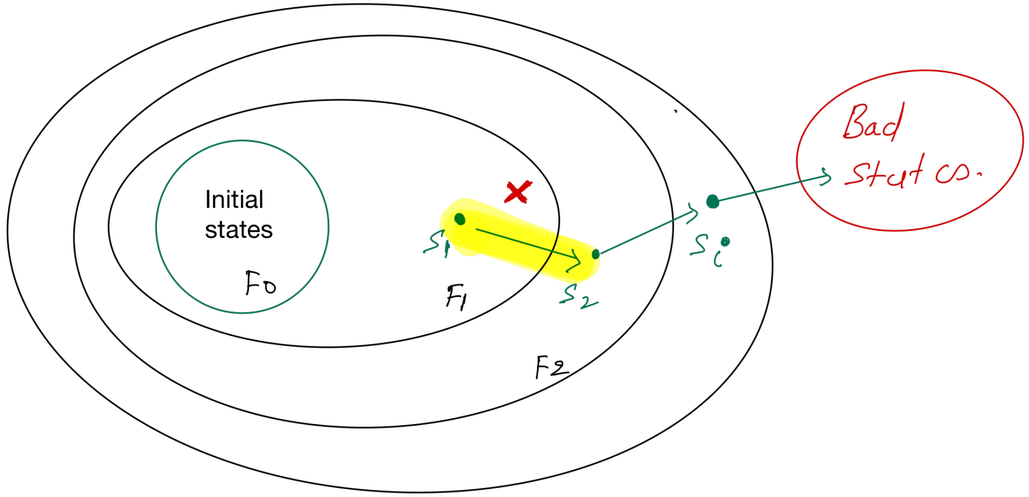


$\exists i F_i = F_{i+1}$

A transition system T is called SAFE if and only if it admits a good closed trace.

Or, found a counter example

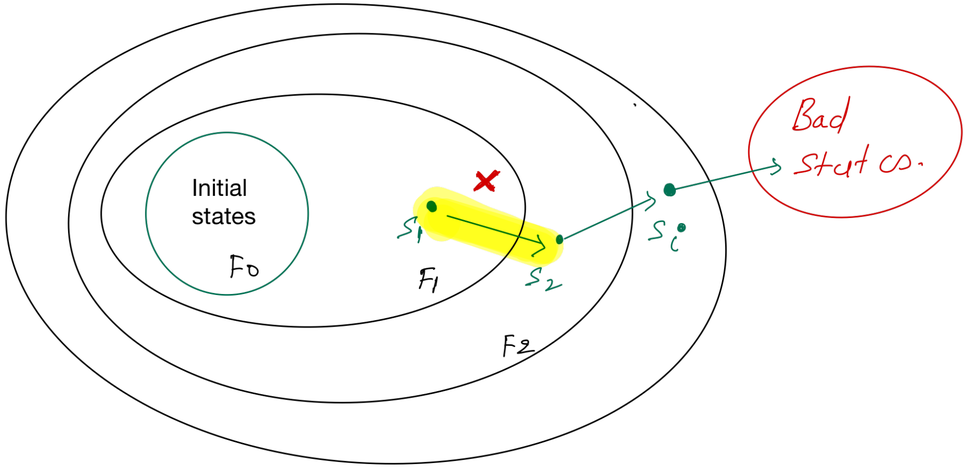
How to update "new" information?



Do we need to update $F_3, \ldots, F_i$?

Do we need to update F_1?

How to update "new" information?



Do we need to update $F_3, \ldots, F_i$?

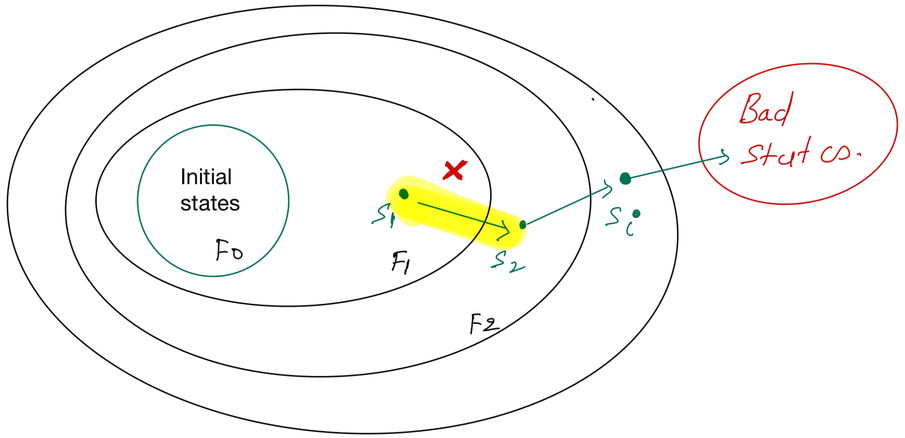
$\forall i : F_i(s) \wedge T(s, s') \rightarrow F_{i+1}(s')$ ✓

A Trace is *Monotone* iff $\forall i, F_i \subseteq F_{i+1}$ ✓

Monotonicity — S_2 is appearing in the later frames as well!!

Then, should we block S_2 from all later frames?

How to update "new" information?



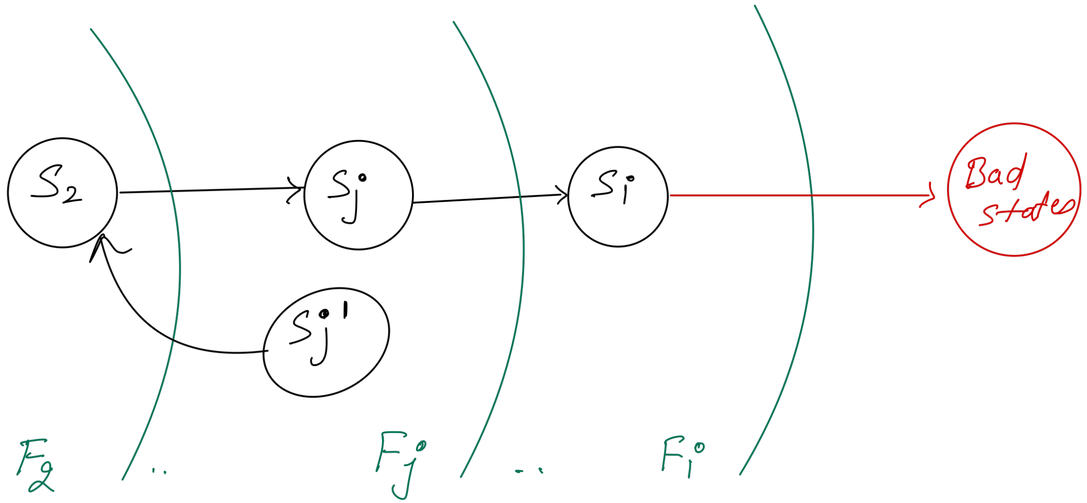
Do we need to update $F_3, \ldots, F_i$?

$$\forall i : F_i(s) \wedge T(s, s') \rightarrow F_{i+1}(s')$$

A Trace is *Monotone* iff $\forall i, F_i \subseteq F_{i+1}$

S_2 is appearing in the later frames as well!!

Then, should we block S_2 from all later frames?

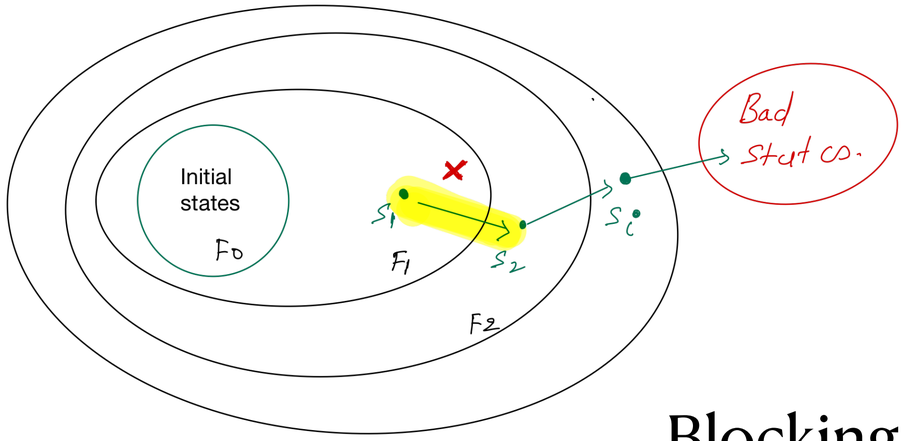


<- what about this case?

$$\forall i : F_i(s) \wedge T(s, s') \rightarrow F_{i+1}(s')$$

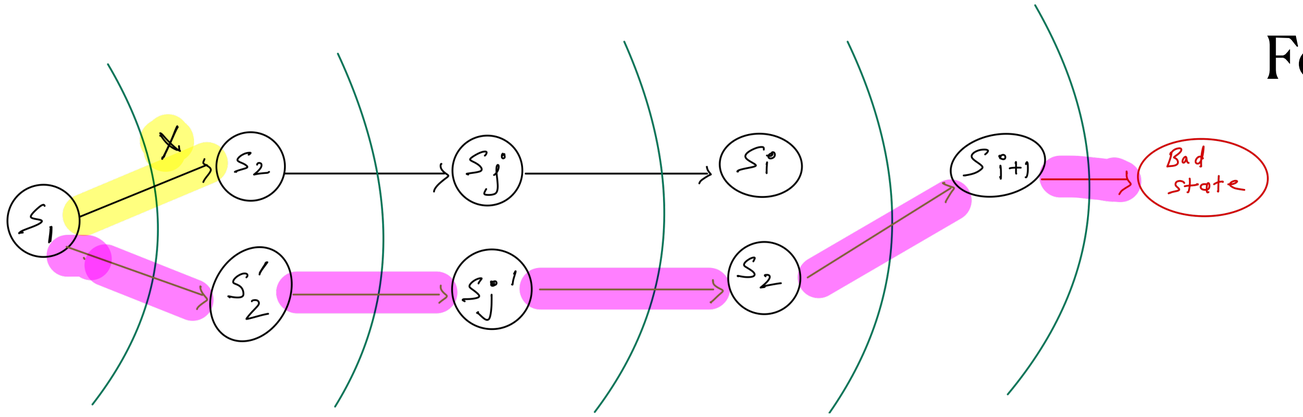
$S_2 \in F_i$ Then, we can't block it from F_i !!!

How to update "new" information?



Do we need to update $F_3, \ldots, F_i$?

Blocking clause for S_2 is c.



For $j \in [2,K]$

If $F_j \wedge T \rightarrow \neg c$, i.e. $SAT\{F_j \wedge T \wedge c\}$

Then Stop

Else

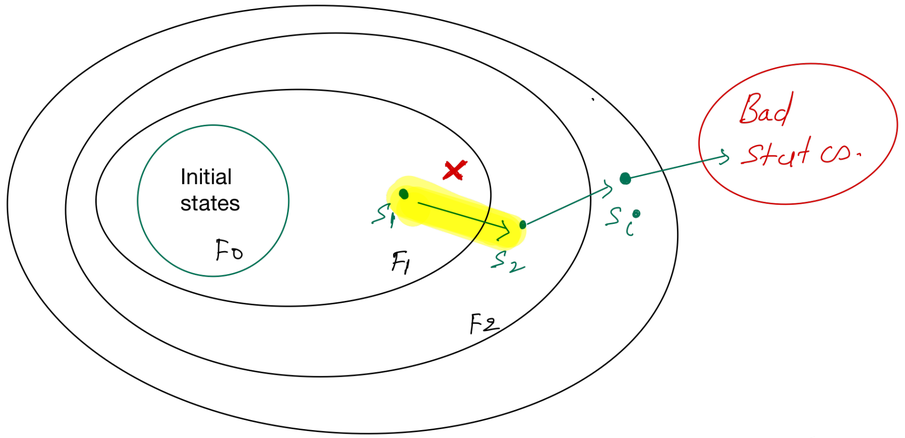
$F_j \leftarrow F_j \wedge \neg c$

Longer Cex may be there!!

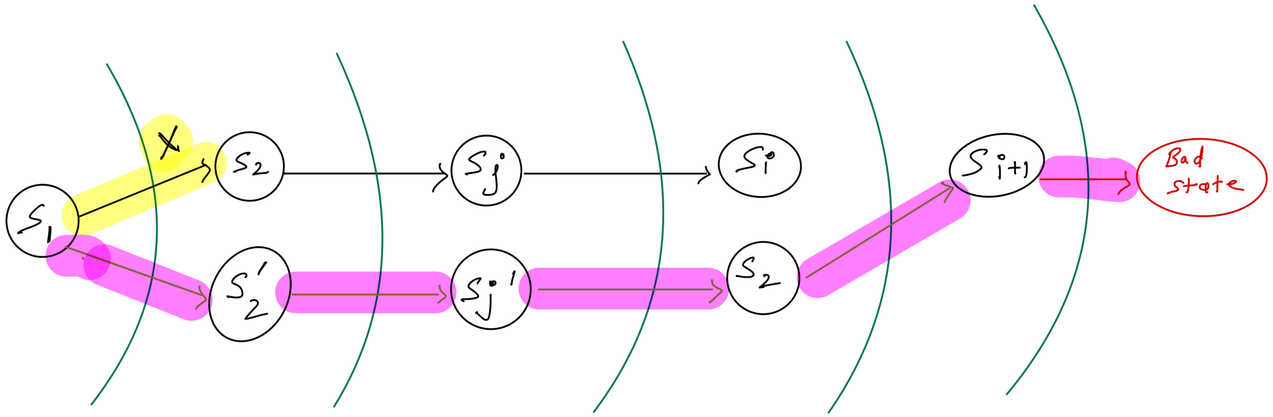
Block S_2 from F_j

But, can't block S_2 from F_i

How to update "new" information?



Do we need to update $F_3, \ldots, F_i$?



Longer Cex may be there!!

Block S_2 from F_j

But, can't block S_2 from F_i

PushForward (F, K, i) {

For $j \in [i, K]$

For every clause c in F_i

If $F_j \wedge T \rightarrow \neg c$, i.e. $SAT\{F_j \wedge T \wedge c\}$

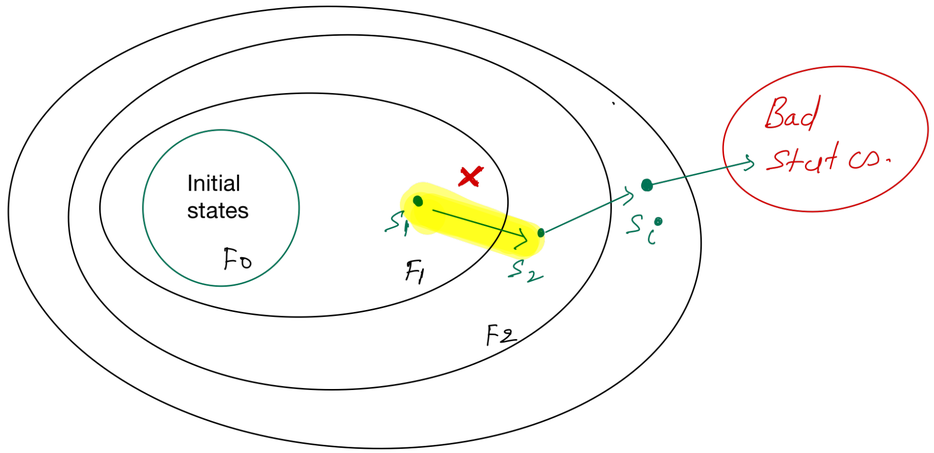
Then continue

Else

$F_j \leftarrow F_j \wedge \neg c$

}

How to update "new" information?



Do we need to update F_1?

$\forall i : F_i(s) \wedge T(s, s') \rightarrow F_{i+1}(s')$ ✓

A Trace is *Monotone* iff $\forall i, F_i \subseteq F_{i+1}$

Block Clause: c

Now, we have updated $F_2 = F_2^{old} \wedge c$!

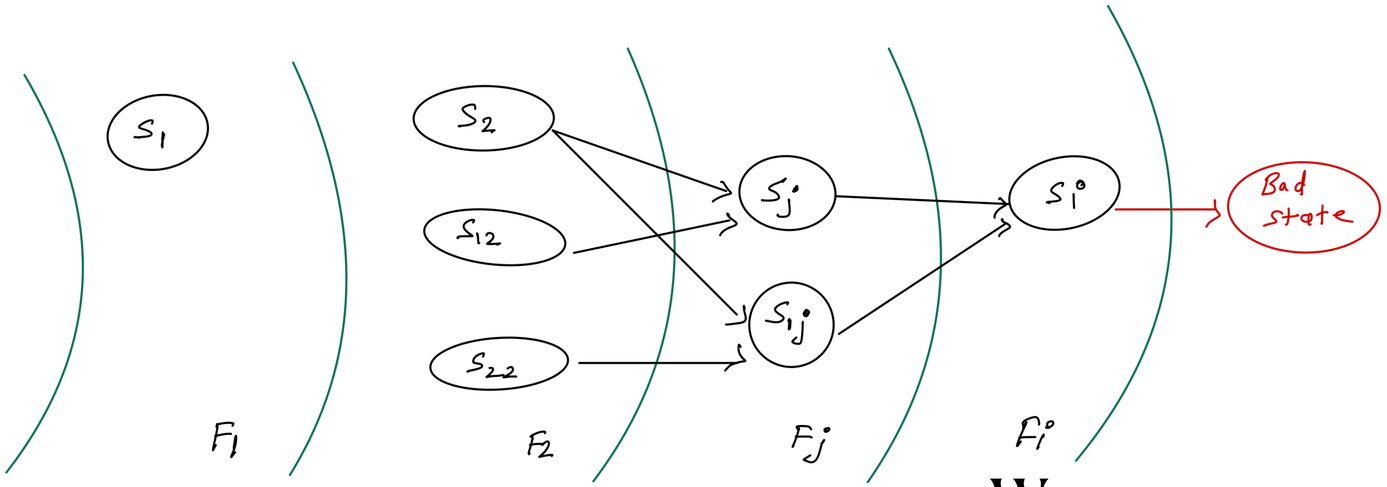
Is still the case $F_1 \subseteq F_2$

Yes, because S_2 was anyway not reachable from F_1, that is, $S_2 \notin F_1$!!

How to update "new" information?

No harm in blocking $S_i, S_j, S_{1j}, S_{2,j}$ from F_2

No harm in blocking S_i from F_j



We block S_2, S_{12}, S_{22} from F_2

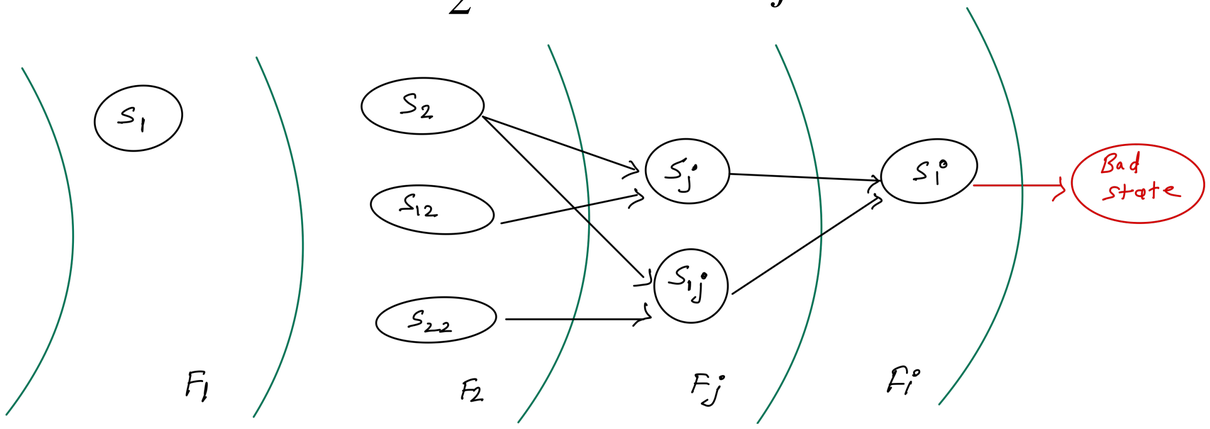
We can also block S_j, S_{1j} from F_j

We can also block S_i from F_i

How to update "new" information?

No harm in blocking $S_i, S_j, S_{1j}, S_{2,j}$ from F_2

No harm in blocking S_i from F_j



We block S_2, S_{12}, S_{22} from F_2

We can also block S_j, S_{1j} from F_j

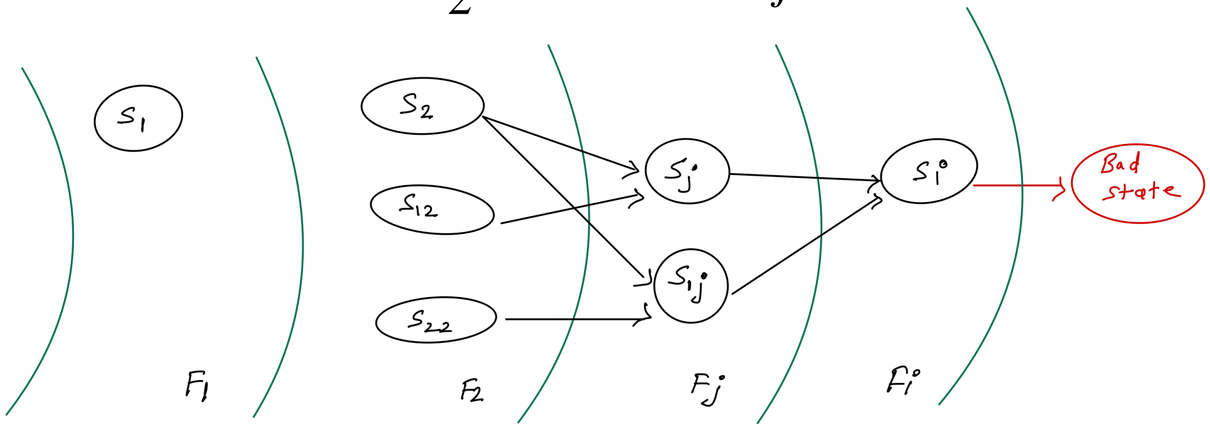
We can also block S_i from F_i

BackwordPropogration(F,T,s,i)
{
While CheckSAT{$F_i \wedge T \wedge s$} do:
$s_i \leftarrow$ predecessor of s extracted from satisfying assigned
For $j \in [0,i]$
$F_j \leftarrow F_j \wedge \neg s_i$
BackwordPropogration(F, T,s_i,$i-1$)
}

How to update "new" information?

No harm in blocking $S_i, S_j, S_{1j}, S_{2,j}$ from F_2

No harm in blocking S_i from F_j



We block S_2, S_{12}, S_{22} from F_2

We can also block S_j, S_{1j} from F_j

We can also block S_i from F_i

BackwordPropogration(F,T,s,i)

{

While CheckSAT{$F_i \wedge T \wedge s$} {

If $i == 0$:

found CEX

Return

$s_i \leftarrow$ predecessor of s extracted from satisfying assigned

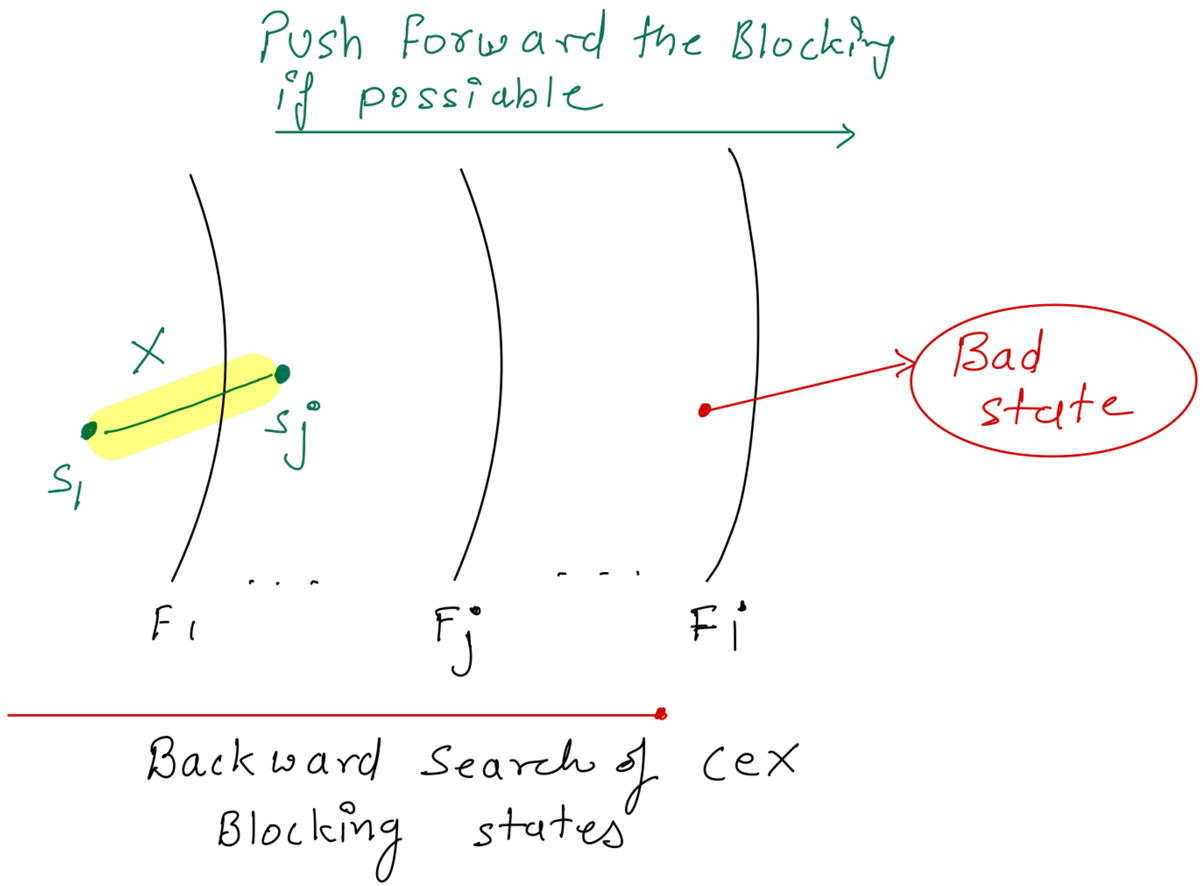
For $j \in [0,i]$

$F_j \leftarrow F_j \wedge \neg s_i$

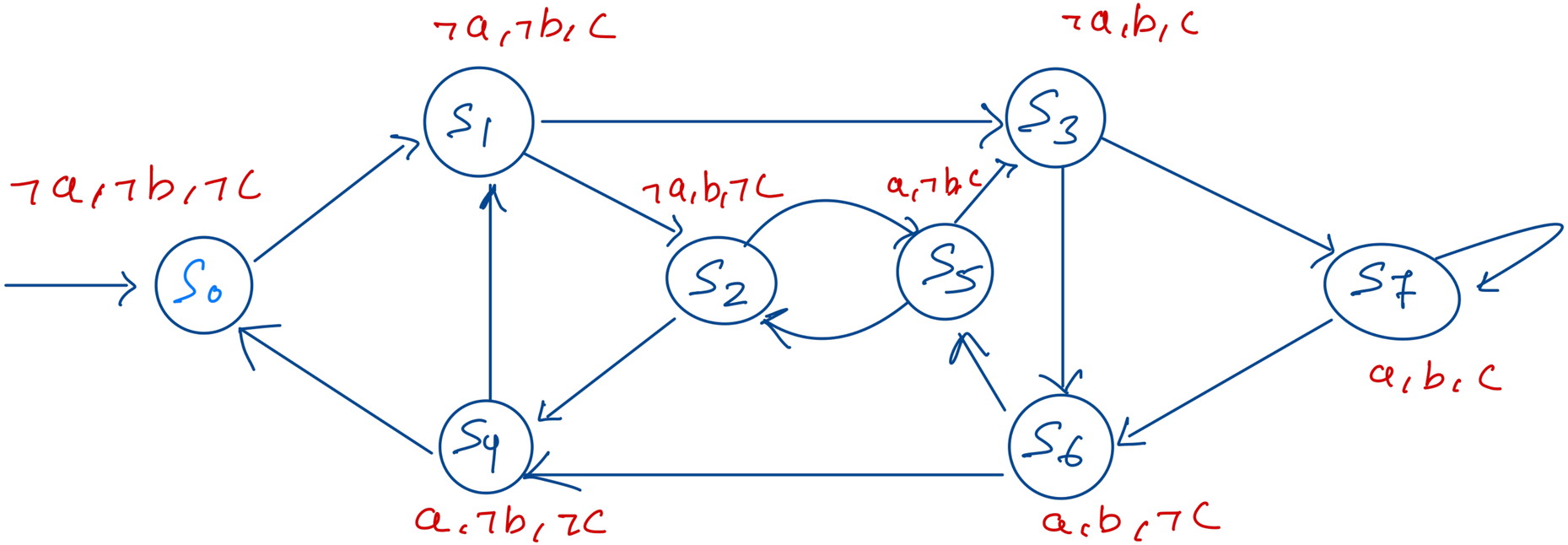
BackwordPropogration(F, T,s_i,$i-1$)}

}

IC3 : Incremental Construction of Inductive Clauses for Indubitable Correctness.



IC3 : Incremental Construction of Inductive Clauses for Indubitable Correctness.



$$\mathrm{T}(a, b, c, a', b', c') = (a' \leftrightarrow b) \wedge (b' \leftrightarrow c)$$

$$\forall \square \neg a \vee \neg b \vee \neg c$$