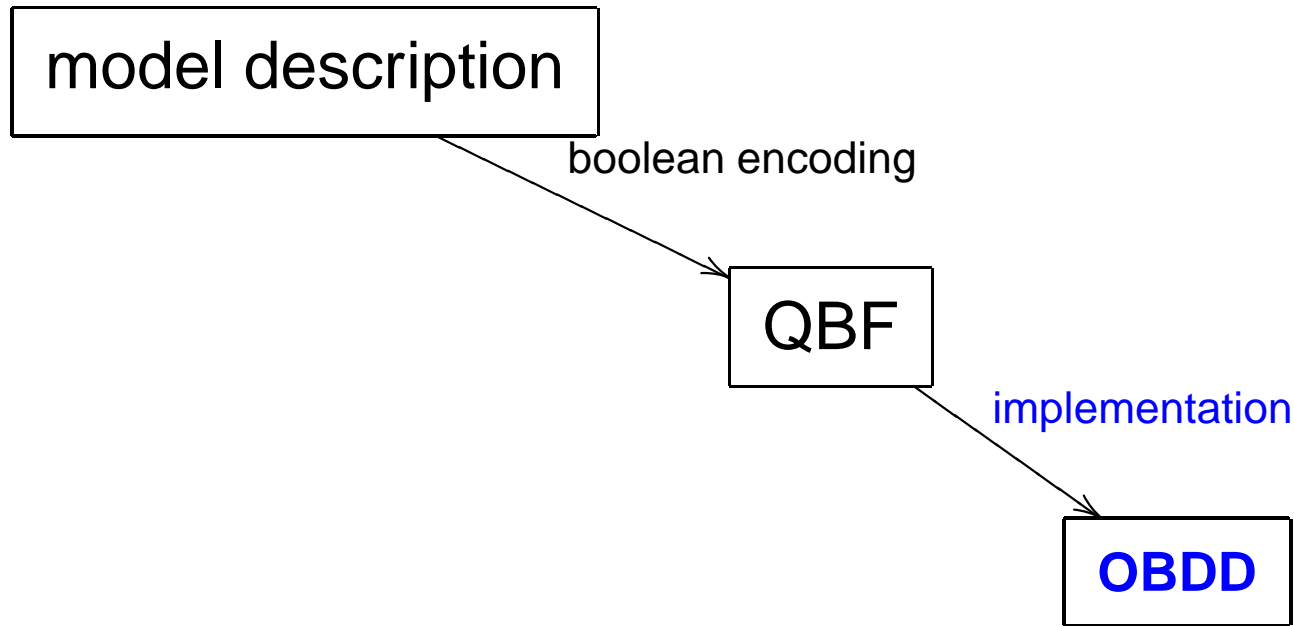
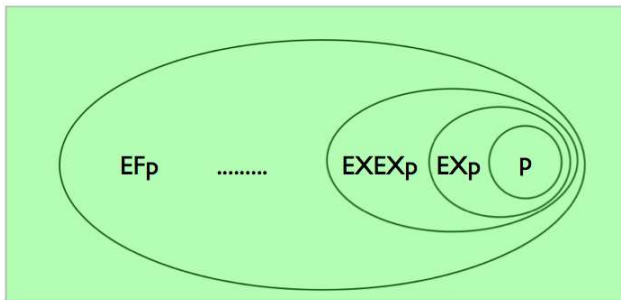


# Computer aided verification

## Lecture 8: Bounded model checking



**model checking = operations on OBDDs**



**Example:**  $\text{EF } p$  (safety)

pre-SMC:

$$\dots \supseteq p \cup \mathbf{EX}(p \cup \mathbf{EX} p) \supseteq p \cup \mathbf{EX} p \supseteq p \supseteq \text{false}$$

$$\dots \supseteq p \cup \text{pre}(p \cup \text{pre}(p)) \supseteq p \cup \text{pre}(p) \supseteq p \supseteq \text{false}$$

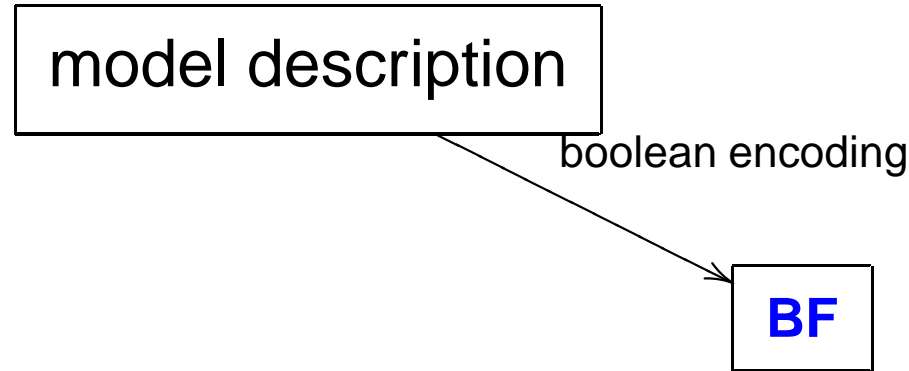
post-SMC:

$$S_0 \subseteq S_0 \cup \text{post}(S_0) \subseteq S_0 \cup \text{post}(S_0 \cup \text{post}(S_0)) \subseteq \dots$$

$$\exists \vec{z}_0 \exists \vec{z}_1 \dots \exists \vec{z}_k \quad \vec{z}_0 \longrightarrow \vec{z}_1 \longrightarrow \vec{z}_2 \longrightarrow \dots \longrightarrow \vec{z}_{k-1} \longrightarrow \vec{z}_k$$

$$\vec{z} = \vec{z}_0 \vee \vec{z} = \vec{z}_1 \vee \dots \vee \vec{z} = \vec{z}_k$$

# Bounded model checking (BMC)



**model checking = satisfiability of boolean formula**

$$\vec{z}_0 \longrightarrow \vec{z}_1 \longrightarrow \vec{z}_2 \longrightarrow \dots \longrightarrow \vec{z}_{k-1} \longrightarrow \vec{z}_k$$

$$p(\vec{z}_0) \vee p(\vec{z}_1) \vee p(\vec{z}_2) \vee \dots \vee p(\vec{z}_k)$$

- searching for a counterexample of **bounded** length
- symbolic path
- application of **SAT-solvers**

# I. BMC

$$M \models \mathbf{E} \phi \quad (\phi \in \mathbf{LTL}^+)$$

e.g.. instead of  $M \models \mathbf{AG} \phi$ , we check  $M \models \mathbf{EF} \neg \phi$

$M$  described by boolean formulas as usual:

$$R(\vec{z}, \vec{z}'), \quad L_p(\vec{z}), \quad S_0(\vec{z})$$

$$R(z_1, \dots, z_m, z'_1, \dots, z'_m), \quad L_p(z_1, \dots, z_m), \quad S_0(z_1, \dots, z_m)$$

**Idea:** Horizon  $k \geq 0$

$k$  system steps

$\Pi \vDash_k \phi$        $k$  steps are sufficient to decide that  $\Pi \vDash \phi$

**Lem.:**  $\Pi \vDash_k \phi \implies \Pi \vDash \phi$

**Lem.:**  $M \vDash \mathbf{E} \phi \implies \exists k \geq 0. M \vDash_k \mathbf{E} \phi$

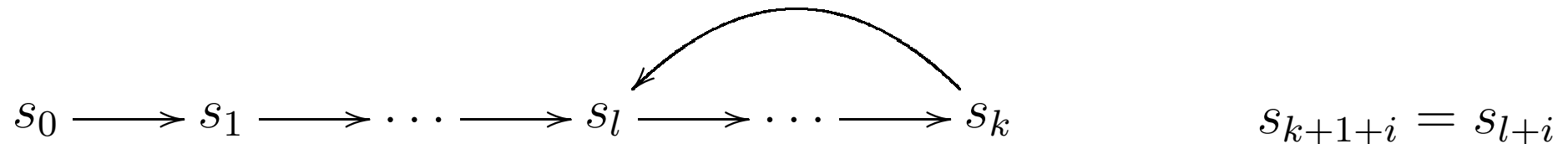
loops!

**Thm.:**  $M \vDash \mathbf{E} \phi \iff \exists k \geq 0. M \vDash_k \mathbf{E} \phi$



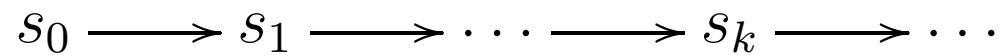
$$\Pi = s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \dots$$

$(k, l)$  - loop:



$(k)$  - loop

no-loop:



# Bounded semantics for $k$ -loops

$$\Pi \vDash_k \phi \iff \Pi \vDash \phi$$

$$\Pi = s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \dots$$

# Bounded semantics for no-loops

$$\Pi \models_k \phi \iff \Pi \models_k^0 \phi$$

$$\Pi = s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \dots$$

$$\models_k^i, \quad 0 \leq i \leq k$$

# Bounded semantics for no-loops

$$\Pi \models_k \phi \iff \Pi \models_k^0 \phi$$

$$\Pi = s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \dots$$

$$\models_k^i, \quad 0 \leq i \leq k$$

$$\Pi \models_k^i p \iff p \in L(s_i)$$

$$\Pi \models_k^i \neg p, \quad \phi_1 \wedge \phi_2, \quad \phi_1 \vee \phi_2 \iff \dots$$

$$\Pi \models_k^i \mathbf{X} \phi \iff i < k \text{ i } \Pi \models_k^{i+1} \phi$$

$$\Pi \models_k^i \mathbf{F} \phi \iff \exists j, i \leq j \leq k. \Pi \models_k^j \phi$$

$$\Pi \models_k^i \mathbf{G} \phi \quad \text{never}$$

$$\Pi \models_k^i \phi \mathbf{U} \psi \iff \exists j, i \leq j \leq k. \Pi \models_k^j \psi \wedge \forall l, i \leq l < j. \Pi \models_k^l \phi$$

$$\Pi \models_k^i \phi \mathbf{R} \psi \iff ?$$

**Tw.:**  $M \models \mathbf{E} \phi \iff \exists k \geq 0. M \models_k \mathbf{E} \phi$

- what  $k$  is sufficiently big? graph diameter?
- no completeness !
- efficiency of SMC depends on graph diameter

$$M \models_k \mathbf{E} \phi \quad (\phi \in \mathbf{LTL}^+)$$

$$M, \phi, k \mapsto [M, \phi]_k$$

$$M \models_k \phi \iff [M, \phi]_k \text{ satisfiable}$$

$M$  described by boolean formulas as usual:

$$R(\vec{z}, \vec{z}'), \quad L_p(\vec{z}), \quad S_0(\vec{z})$$

$$R(z_1, \dots, z_m, z'_1, \dots, z'_m), \quad L_p(z_1, \dots, z_m), \quad S_0(z_1, \dots, z_m)$$

## Symbolic path

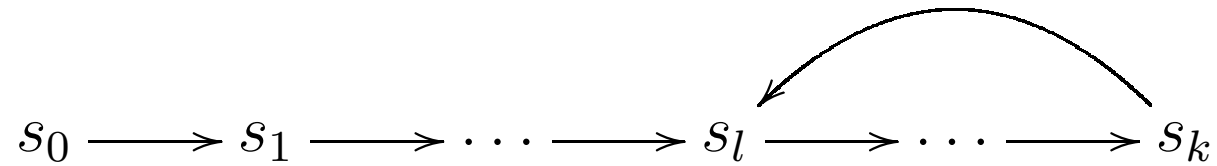


$$[M]_k := S_0(\vec{z}_0) \wedge \bigwedge_{i=0}^{k-1} R(\vec{z}_i, \vec{z}_{i+1})$$

$(k + 1) \cdot m$  boolean variables

$[M]_k$  is of size  $\mathcal{O}(k \cdot |M|)$

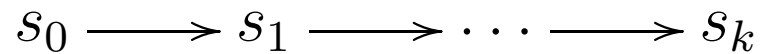
$(k, l)$ -loop:



$${}_l L_k \equiv R(\vec{z}_k, \vec{z}_l)$$

$$L_k \equiv \bigvee_{l=0}^k {}_l L_k$$

no-loop:





$$[M, \phi]_k := [M]_k \wedge \left( (\neg L_k \wedge [\phi]_k^0) \vee \bigvee_{l=0}^k (l L_k \wedge l[\phi]_k^0) \right)$$

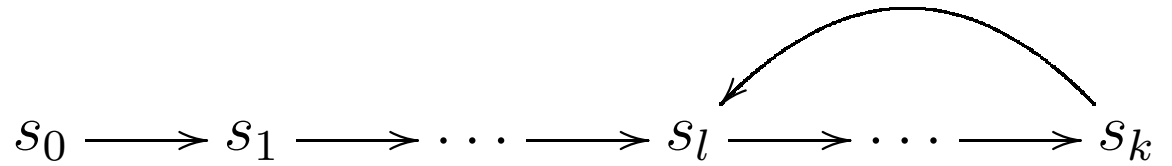
$l[\phi]_k^0$  – semantics of  $\phi$  on  $(k, l)$ -loop

$[\phi]_k^0$  – semantics of  $\phi$  on no-loop

$[M, \phi]_k$  is of size  $\mathcal{O}(k \cdot (|M| + |\phi|))$

# Boolean encoding

$$l[\phi]_k^i \quad 0 \leq l, i \leq k$$



$$l[p]_k^i \iff L_p(\vec{z}_i)$$

$$l[\neg p]_k^i \quad l[\phi \wedge \psi]_k^i$$

$$l[\phi \vee \psi]_k^i \iff l[\phi]_k^i \vee l[\psi]_k^i$$

$$l[X\phi]_k^i \iff l[\phi]_k^{\text{succ}(i)}$$

$$\text{succ}(i) = \begin{cases} i + 1 & i < k \\ l & i = k \end{cases}$$

$$\mathbf{F}\phi \equiv \phi \vee \mathbf{X}\mathbf{F}\phi$$

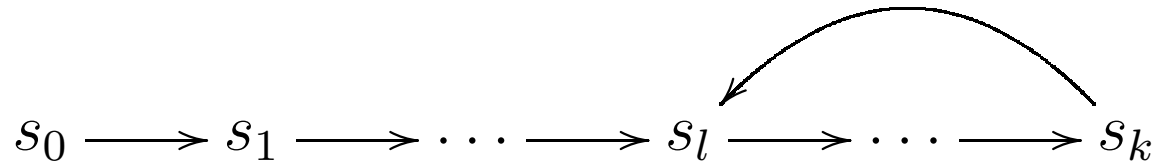
$$\phi \mathbf{U} \psi \equiv \psi \vee (\phi \wedge \mathbf{X}\phi \mathbf{U} \psi)$$

$$\mathbf{G}\phi \equiv \phi \wedge \mathbf{X}\mathbf{G}\phi$$

$$\phi \mathbf{R} \psi \equiv \psi \wedge (\phi \vee \mathbf{X}\phi \mathbf{R} \psi)$$

# Boolean encoding

$$l[\phi]_k^i \quad 0 \leq l, i \leq k$$



$$l[p]_k^i \iff L_p(\vec{z}_i)$$

$$l[\neg p]_k^i \quad l[\phi \wedge \psi]_k^i \quad l[\phi \vee \psi]_k^i \iff l[\phi]_k^i \vee l[\psi]_k^i$$

$$l[\mathbf{X} \phi]_k^i \iff l[\phi]_k^{\text{succ}(i)}$$

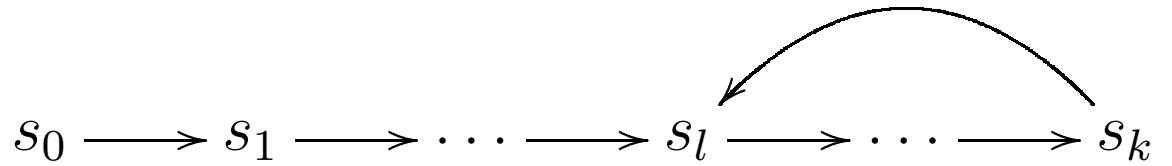
$$\text{succ}(i) = \begin{cases} i + 1 & i < k \\ l & i = k \end{cases}$$

$$l[\mathbf{F} \phi]_k^i \iff l[\phi]_k^i \vee l[\mathbf{F} \phi]_k^{\text{succ}(i)}$$

$$l[\mathbf{G} \phi]_k^i \iff l[\phi]_k^i \wedge l[\mathbf{G} \phi]_k^{\text{succ}(i)}$$

$$l[\phi \mathbf{U} \psi]_k^i \iff l[\psi]_k^i \vee (l[\phi]_k^i \wedge l[\phi \mathbf{U} \psi]_k^{\text{succ}(i)})$$

**Example:**



$${}_l[\mathbf{F} p]_k^0 \iff L_p(\vec{z}_0) \vee \dots \vee L_p(\vec{z}_k)$$

$${}_l[\mathbf{G} p]_k^0 \iff L_p(\vec{z}_0) \wedge \dots \wedge L_p(\vec{z}_k)$$

$${}_l[p \mathbf{U} q]_k^0 \iff L_q(\vec{z}_0) \vee (L_p(\vec{z}_0) \wedge (L_q(\vec{z}_1) \vee (\dots \\ L_q(\vec{z}_{k-1}) \vee (L_p(\vec{z}_{k-1}) \wedge L_q(\vec{z}_k)) \dots))))$$

$${}_l[\mathbf{F} \mathbf{G} p]_k^0 \iff (L_p(\vec{z}_0) \wedge L_p(\vec{z}_1) \wedge \dots \wedge L_p(\vec{z}_k)) \vee \\ (L_p(\vec{z}_1) \wedge \dots \wedge L_p(\vec{z}_k)) \vee \dots \vee (L_p(\vec{z}_l) \wedge \dots \wedge L_p(\vec{z}_k))$$

$$\iff (L_p(\vec{z}_l) \wedge \dots \wedge L_p(\vec{z}_k))$$

$\iota[\phi]_k^0$  is of size  $\mathcal{O}(k \cdot |\phi|)$

# Boolean encoding

$$[\phi]_k^i \quad 0 \leq i \leq k$$

$$s_0 \longrightarrow s_1 \longrightarrow \dots \longrightarrow s_k$$

$$[p]_k^i \quad [\neg p]_k^i \quad [\phi \wedge \psi]_k^i \quad [\phi \vee \psi]_k^i \iff \dots \quad \text{j. w.}$$

$$[\mathbf{X} \phi]_k^i \iff [\phi]_k^{i+1}$$

$$[\phi]_k^{k+1} \iff \text{false}$$

$$\mathbf{F} \phi \equiv \phi \vee \mathbf{X} \mathbf{F} \phi$$

$$\phi \mathbf{U} \psi \equiv \psi \vee (\phi \wedge \mathbf{X} \phi \mathbf{U} \psi)$$

$$\mathbf{G} \phi \equiv \phi \wedge \mathbf{X} \mathbf{G} \phi$$

$$\phi \mathbf{R} \psi \equiv \psi \wedge (\phi \vee \mathbf{X} \phi \mathbf{R} \psi)$$

## Example:

$$s_0 \longrightarrow s_1 \longrightarrow \dots \longrightarrow s_k$$

$$[\mathbf{F} p]_k^0 \iff L_p(\vec{z}_0) \vee \dots \vee L_p(\vec{z}_k) \vee \mathbf{false}$$

$$[\mathbf{G} p]_k^0 \iff L_p(\vec{z}_0) \wedge \dots \wedge L_p(\vec{z}_k) \wedge \mathbf{false} \equiv \mathbf{false}$$

$$[p \mathbf{U} q]_k^0 \iff L_q(\vec{z}_0) \vee (L_p(\vec{z}_0) \wedge (L_q(\vec{z}_1) \vee (\dots \\ L_q(\vec{z}_{k-1}) \vee (L_p(\vec{z}_{k-1}) \wedge (L_q(\vec{z}_k) \vee (L_p(\vec{z}_k) \wedge \mathbf{false})))) \dots)))$$

$$[\mathbf{F} \mathbf{G} p]_k^0 \iff \mathbf{false}$$

$$[M, \phi]_k := [M]_k \wedge \left( (\neg L_k \wedge [\phi]_k^0) \vee \bigvee_{l=0}^k (l L_k \wedge l[\phi]_k^0) \right)$$

**Thm.:**  $M \models_k \mathbf{E} \phi \iff [M, \phi]_k$  is satisfiable.

Repeat:

(1)  $k := k_0$

(2) if  $[M, \phi]_k$  satisfiable then stop

$M \models \mathbf{E} \phi$

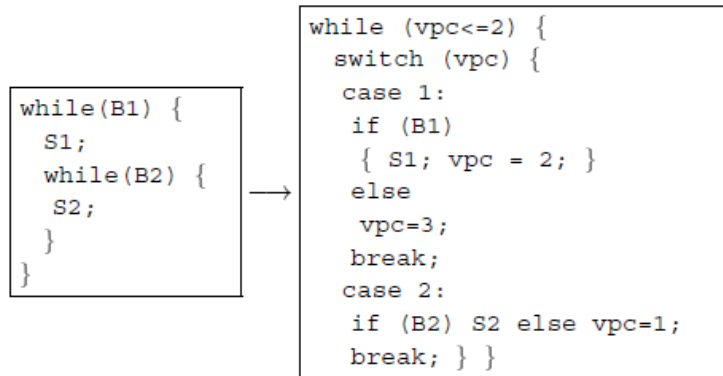
(3) increment  $k$  and continue



# Other boolean encodings

- hardware verification:
  - circuit structure kept, designated SAT-solvers
- LTL with past
- software verification (e.g. CBMC)
  - loop unfolding, heap model
- concurrent programs
- ...





```

x=x+y;
if(x!=1) {
  x=2;
  if(z) x++;
}
assert(x<=3);

```



```

x1=x0+y0;
if(x1!=1) {
  x2=2;
  if(z0) x3=x2+1;
}
assert(x3<=3);

```



```

C := (x1=x0+y0) ∧
     x2=((x1 ≠ 1)?2:x1) ∧
     x3=((x1 ≠ 1 ∧ z0)?x2+1:x2)
P := x3 ≤ 3

```

[Kroening, Clarke, Yorav 2003]

## II. Completeness ?

## Is completeness achievable ?

- completeness threshold for  $k$  (safety  $G p$ )
- simultaneous checks of  $\phi$  and  $\neg\phi$  (liveness  $F p$ )
- induction (safety  $G p$ )

Reachability diameter – bound for  $k$

the least  $i$  such that

$$\forall \vec{z}_0, \dots, \vec{z}_n. \exists \vec{z}'_0, \dots, \vec{z}'_t, t \leq i. S_0(\vec{z}_0) \wedge \bigwedge_{j=0}^{n-1} R(\vec{z}_j, \vec{z}_{j+1}) \implies$$

$$S_0(\vec{z}'_0) \wedge \left( \bigwedge_{j=0}^{t-1} R(\vec{z}'_j, \vec{z}'_{j+1}) \right) \wedge \vec{z}'_t = \vec{z}_n$$

Reachability diameter – bound for  $k$

the least  $i$  such that

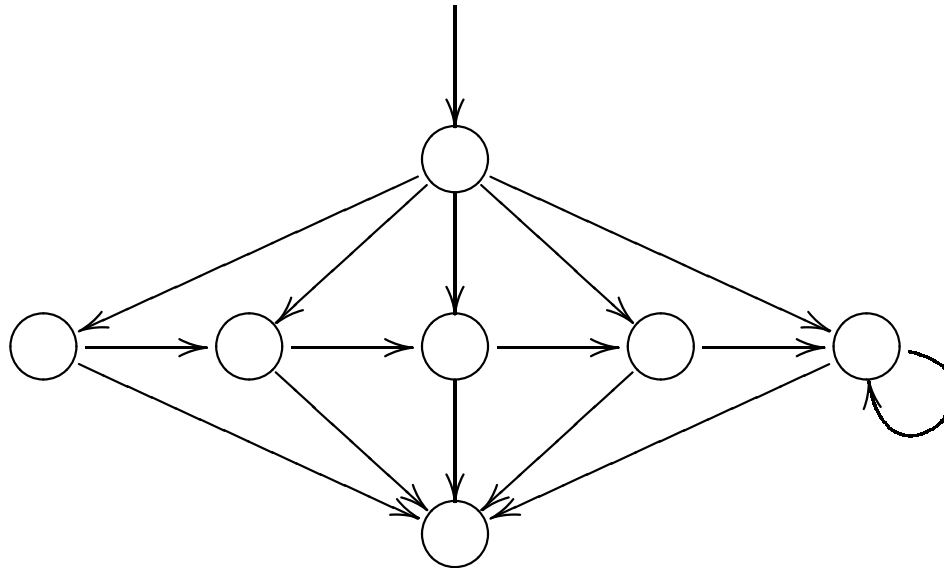
$$\forall \vec{z}_0, \dots, \vec{z}_{i+1}. \exists \vec{z}'_0, \dots, \vec{z}'_i. S_0(\vec{z}_0) \wedge \bigwedge_{j=0}^i R(\vec{z}_j, \vec{z}_{j+1}) \implies$$

$$S_0(\vec{z}'_0) \wedge \left( \bigwedge_{j=0}^{i-1} R(\vec{z}'_j, \vec{z}'_{j+1}) \right) \wedge \bigvee_{j=0}^i \vec{z}'_j = \vec{z}_{i+1}$$

the longest loop-free path – bound for  $k$

the greatest  $i$  such that

$$\exists \vec{z}'_0, \dots, \vec{z}_i. S_0(\vec{z}_0) \wedge \left( \bigwedge_{j=0}^{i-1} R(\vec{z}_j, \vec{z}_{j+1}) \right) \wedge \bigwedge_{j=0}^{i-1} \bigwedge_{l=j+1}^i \vec{z}_j \neq \vec{z}_l$$



$M \models \mathbf{EG} \neg p \iff \exists k$  s.t. the following formula is satisfiable:

$$\mathbf{No}_k(\vec{z}_0, \dots, \vec{z}_{k+1}) \equiv S_0(\vec{z}_0) \wedge \bigwedge_{j=0}^{k-1} R(\vec{z}_j, \vec{z}_{j+1}) \wedge \left( \bigvee_{l=0}^k l L_k \right) \wedge \bigwedge_{j=0}^k \neg L_p(\vec{z}_j)$$

$M \models \mathbf{AF} p \iff \exists k$  s.t. the following formula is a tautology:

$$\mathbf{Yes}_k(\vec{z}_0, \dots, \vec{z}_{k+1}) \equiv S_0(\vec{z}_0) \wedge \bigwedge_{j=0}^{k-1} R(\vec{z}_j, \vec{z}_{j+1}) \implies \bigvee_{j=0}^k L_p(\vec{z}_j)$$



$M \models \mathbf{EG} \neg p \iff \exists k$  s.t. the following formula is satisfiable:

$$\mathbf{No}_k(\vec{z}_0, \dots, \vec{z}_{k+1}) \equiv S_0(\vec{z}_0) \wedge \bigwedge_{j=0}^{k-1} R(\vec{z}_j, \vec{z}_{j+1}) \wedge \left( \bigvee_{l=0}^k l L_k \right) \wedge \bigwedge_{j=0}^k \neg L_p(\vec{z}_j)$$

$M \models \mathbf{AF} p \iff \exists k$  s.t. the following formula is unsatisfiable:

$$\neg \mathbf{Yes}_k(\vec{z}_0, \dots, \vec{z}_{k+1}) \equiv S_0(\vec{z}_0) \wedge \bigwedge_{j=0}^{k-1} R(\vec{z}_j, \vec{z}_{j+1}) \wedge \bigwedge_{j=0}^k \neg L_p(\vec{z}_j)$$

Repeat:

(1)  $k := 0$

(2) if  $No_k$  satisfiable then stop

(3) if  $\neg Yes_k$  unsatisfiable then stop

(4) increase  $k$  and continue

$$M \models EG \neg \phi$$

$$M \models Fp$$

# Completeness for $G_p$ – induction

## – induction base

check that the following formula is unsatisfiable:

$$\mathbf{Base}_k(\vec{z}_0, \dots, \vec{z}_k) \equiv S_0(\vec{z}_0) \wedge \bigwedge_{j=0}^{k-1} R(\vec{z}_j, \vec{z}_{j+1}) \wedge \bigvee_{j=0}^k \neg L_p(\vec{z}_j)$$

## – induction step

check that the following formula is unsatisfiable:

$$\mathbf{Step}_k(\vec{z}_0, \dots, \vec{z}_{k+1}) \equiv \bigwedge_{j=0}^k (L_p(\vec{z}_j) \wedge R(\vec{z}_j, \vec{z}_{j+1})) \wedge \neg L_p(\vec{z}_{k+1})$$

# Why do we need $k > 1$ ?

Repeat:

(1)  $k := 0$

(2) if Base <sub>$k$</sub>  satisfiable then stop

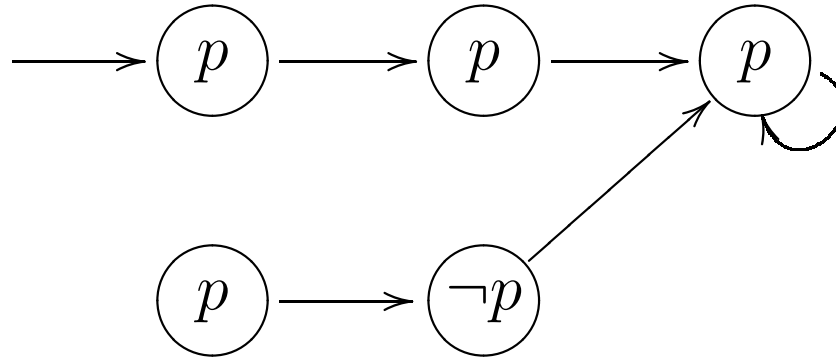
$$M \models \text{EF } \neg \phi$$

(3) if Step <sub>$k$</sub>  unsatisfiable then stop

$$M \models \text{G } p$$

(4) increase  $k$  and continue

# Why do we need $k > 1$ ?



Repeat:

(1)  $k := 0$

(2) if  $\text{Base}_k$  satisfiable then stop

$$M \models \mathbf{EF} \neg \phi$$

(3) if  $\text{Step}_k$  unsatisfiable then stop

$$M \models \mathbf{G} p$$

(4) increase  $k$  and continue

# III. BDD or SAT ?

# Comparison

bit	$k$	SMV <sub>2</sub>	MB	PROVER	MB
0	1	25	79	< 1	1
1	2	25	79	< 1	1
2	3	26	80	< 1	1
3	4	27	82	1	2
4	5	33	92	1	2
5	6	67	102	1	2
6	7	258	172	2	2
7	8	1741	492	7	3
8	9		> 1GB	29	3
9	10			58	3
10	11			91	3
11	12			125	3
12	13			156	4
13	14			186	4
14	15			226	4
15	16			183	5

[Biere, Cimatti, Clarke, Strichman, Zhu 2003]

- the methods seem complementary
- SAT more predictable
- BDD is memory-consuming and SAT time-consuming
- BDD is complete, SAT no
- unbounded model checking UMC:
  - CNF instead of BDD in SMC
- BDD + SAT



1981–1982: EMC —  $10^5$  states

1990–1992: SMV —  $10^{100}$  states

2000': BMC + CEGAR —  $10^{1000}$  states

# IV. SAT-solvers

- CNF
- algorithm DPLL
  - searching through partial valuations tree
  - Boolean Constraint Propagation (BCP)
  - **conflicts** – pruning of the tree
- heuristics

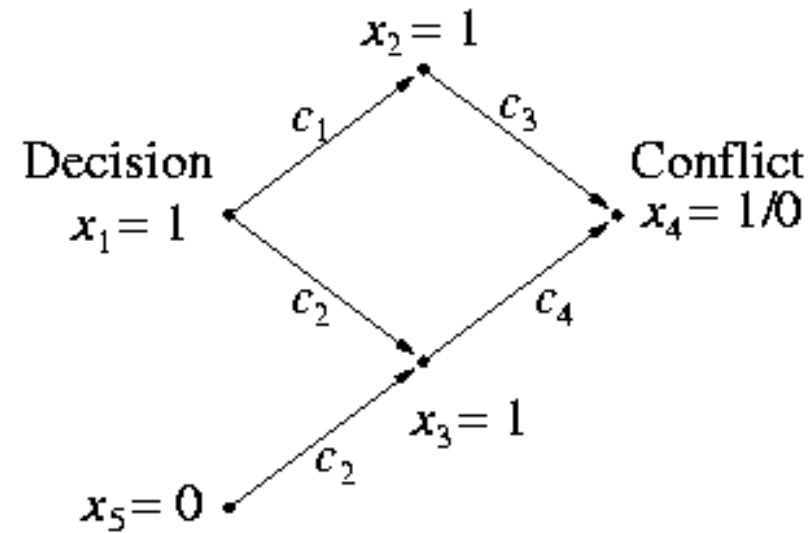
# Implications graph and BCP

$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

$$c_4 = (\neg x_3 \vee \neg x_4)$$



[Biere, Cimatti, Clarke, Strichman, Zhu 2003]

```

// Input arg: Current decision level  $d$ 
// Return value:
//   SAT():      {SAT, UNSAT}
//   Decide():   {DECISION, ALL-DECIDED}
//   Deduce():   {OK, CONFLICT}
//   Diagnose(): {SWAP, BACK-TRACK} also calculates  $\beta$ 

```

```

SAT (d)
{
  l1:   if (Decide (d) == ALL-DECIDED) return SAT;
  l2:   while (TRUE) {
  l3:     if (Deduce(d) != CONFLICT) {
  l4:       if (SAT (d+1) == SAT) return SAT;
  l5:       else if ( $\beta < d$  ||  $d == 0$ )
  l6:         { Erase (d); return UNSAT; }
        }
  l7:   if (Diagnose (d) == BACK-TRACK) return UNSAT;
}
}

```

[Biere, Cimatti, Clarke, Strichman, Zhu 2003]

## Why SAT-solvers are so fast?

- conflict learning – adding conflict clauses
- non-chronological backtracking
- heuristics for decisions
- efficient data structures
- incremental satisfiability