A Transferability Study of Interpolation-Based Hardware Model Checking for Software Verification

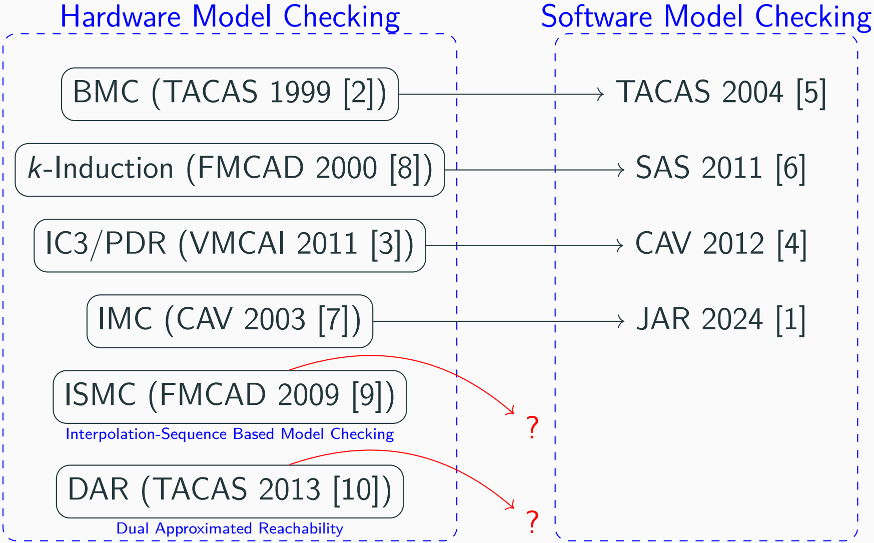
Presenter: Marek Jankola

Dirk Beyer, Po-Chun Chien, Nian-Ze Lee

July 17, 2024

LMU Munich, SoSy-Lab

Motivation



Hardware Model Checking

Software Model Checking

BMC (TACAS 1999 [2]) → TACAS 2004 [5]

k-Induction (FMCAD 2000 [8]) → SAS 2011 [6]

IC3/PDR (VMCAI 2011 [3]) → CAV 2012 [4]

IMC (CAV 2003 [7]) → JAR 2024 [1]

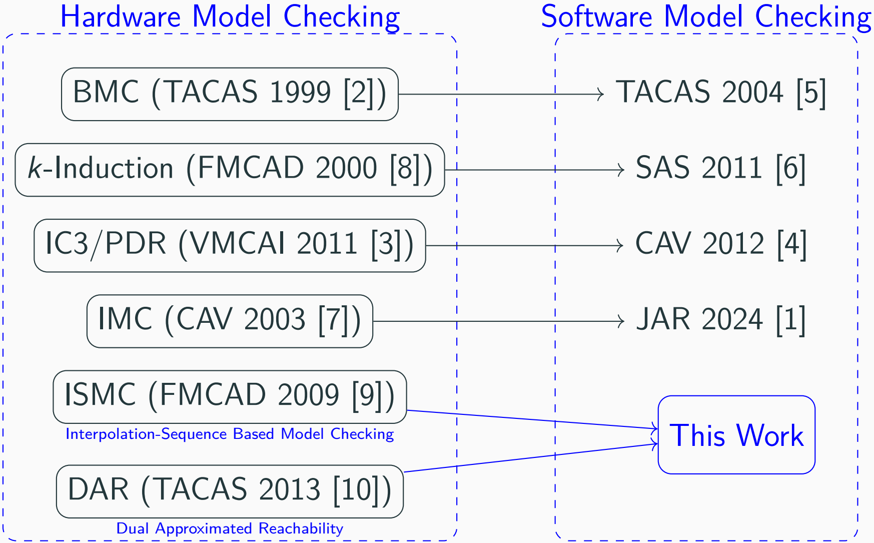
ISMC (FMCAD 2009 [9]) → ?

Interpolation-Sequence Based Model Checking

DAR (TACAS 2013 [10]) → ?

Dual Approximated Reachability

Motivation



Hardware Model Checking

BMC (TACAS 1999 [2]) → TACAS 2004 [5]

k-Induction (FMCAD 2000 [8]) → SAS 2011 [6]

IC3/PDR (VMCAI 2011 [3]) → CAV 2012 [4]

IMC (CAV 2003 [7]) → JAR 2024 [1]

ISMC (FMCAD 2009 [9]) → This Work

Interpolation-Sequence Based Model Checking

DAR (TACAS 2013 [10]) → This Work

Dual Approximated Reachability

Software Model Checking

Summary

- First systematic transferability study of hardware model checking
- ISMC and DAR are useful in software verification
- Evaluation confirms important claims for software
- Open-source implementation of ISMC and DAR in CPAchecker

Basic Definitions and Common Characteristics

- State-transition system: $\mathscr{M} = (I(s), T(s,s'))$ and a reachability-safety property $P(s)$
- BMC check: $I(s_0) \land T(s_0,s_1) \land \cdots \land T(s_{k-1},s_k) \land \neg P(s_k)$ (UNSAT $=$ No Error)
- Interpolation sequence: formulas $\langle \tau_1, \ldots, \tau_k \rangle$ overapproximating sets of states reachable in $1, 2, \ldots, k$ steps
- BMC checks vs fixed-point computation

Conceptual Differences between the Algorithms

	Fixed Point	Reuse Interpolants	Local/Global Checks
IMC	$\bigvee_{i=1}^{n} \tau_i$	No	Global
ISMC	$\langle R_1, \ldots, R_n \rangle,\ R_i = \bigwedge_{j=i}^{n} \tau_i^j$	Yes	Global
DAR	$\langle F_1, \ldots, F_n \rangle,\ \langle B_1, \ldots, B_n \rangle$	Yes	Local/Global

Study Design - ISMC vs IMC

		Confirmed	
	Hypothesis	Original	Our
H1.A	ISMC faster in finding bugs	✓	✓
H1.B	ISMC faster in proving property if high unrolling bound	✓	
H1.C	ISMC overall faster	✓	

Study Design - DAR vs IMC

		Confirmed	
	Hypothesis	Original	Our
H2.A	DAR performs more local phases than global	✓	✓
H2.B	DAR faster in proving property	✓	
H2.C	DAR computes more interpolants	✓	✓
H2.D	DAR's run-time more sensitive to sizes of interpolants	✓	
H2.E	DAR overall faster than IMC	✓	

Differences between Our and Previous Experimental Setups

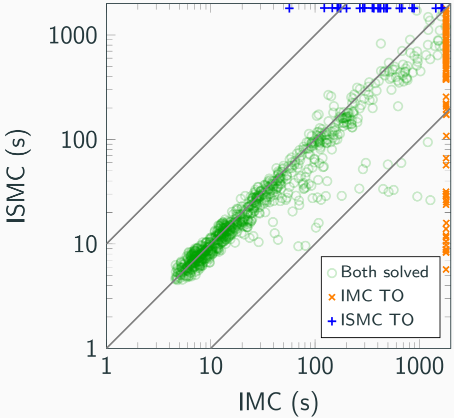
		Ours	ISMC [9]	DAR [10]
	platform	CPAchecker	Intel's tool	Cadence's Jasper
	solver	MathSAT5 (SMT)	Eureka (SAT)	a SAT solver
benchmark set	source	AWS C Comm., Linux, . . .	Intel CPUs	industrial HW designs
	type	C program	HW circuit	HW circuit
	#safe	6020	69	37
	#unsafe	2793	67	≥ 4

Experimental Setup

- ISMC and DAR in the common framework with IMC: CPAchecker
- Benchmark set: 8813 C programs from SV-COMP
- Machines with 3.40 GHz CPU (Intel Xeon E3-1230 v5)
- Cores limit: 2; CPU time limit: 1800 s; memory limit: 15 GB

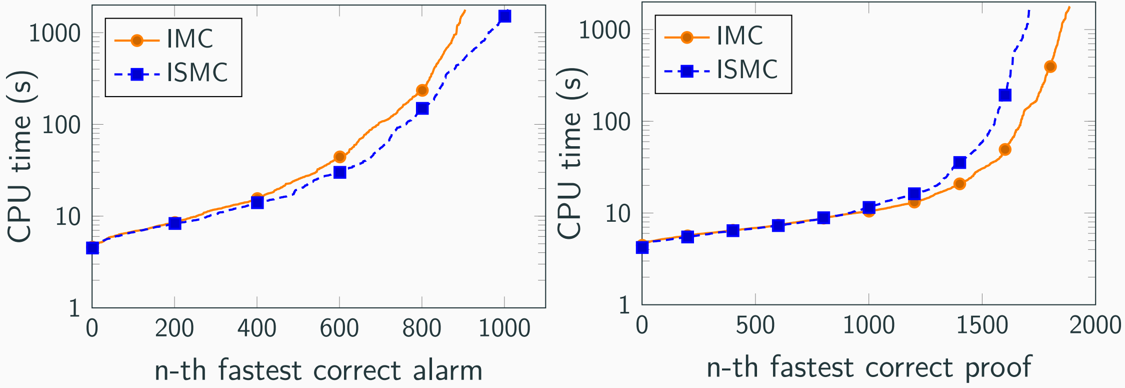
Evaluation - ISMC

H1.A: ISMC is faster than IMC on tasks with property violation. ✓



Evaluation - ISMC

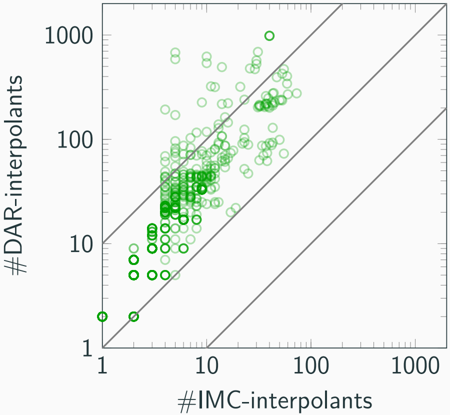
H1.C: Overall, ISMC is faster than IMC (by 30% in the original publication). **?**

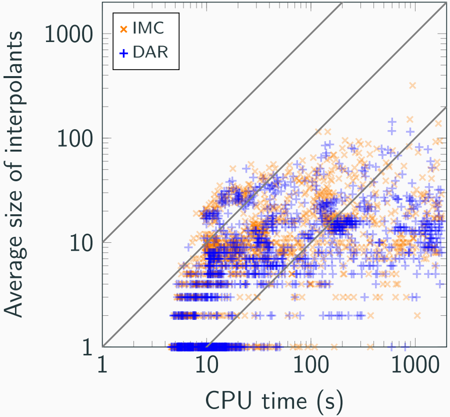


Evaluation - DAR

H2.C: DAR computes more interpolants than IMC. ✓

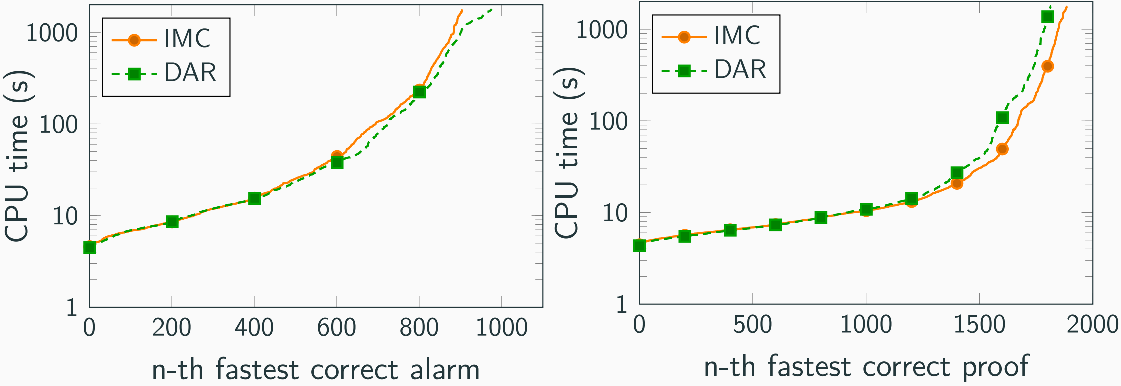
H2.D: DAR's run-time is more sensitive to the sizes of interpolants than IMC. ?





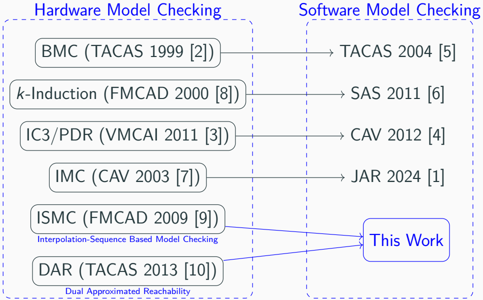
Evaluation - DAR

H2.E: Overall, DAR is faster than IMC (by 36% in the original publication). ?



Conclusion

- First systematic transferability study of hardware model checking
- ISMC and DAR are useful in software verification
- Evaluation confirms important claims for software
- Open-source implementation of ISMC and DAR in CPAchecker



References i

[1] Beyer, D., Lee, N.Z., Wendler, P.: Interpolation and SAT-based model checking revisited: Adoption to software verification. J. Autom. Reasoning (2024). https://doi.org/10.1007/s10817-024-09702-9, preprint: https://doi.org/10.48550/arXiv.2208.05046

[2] Biere, A., Cimatti, A., Clarke, E.M., Zhu, Y.: Symbolic model checking without BDDs. In: Proc. TACAS. pp. 193–207. LNCS 1579, Springer (1999). https://doi.org/10.1007/3-540-49059-0_14

[3] Bradley, A.R.: SAT-based model checking without unrolling. In: Proc. VMCAI. pp. 70–87. LNCS 6538, Springer (2011). https://doi.org/10.1007/978-3-642-18275-4_7

References ii

[4] Cimatti, A., Griggio, A.: Software model checking via IC3. In: Proc. CAV. pp. 277–293. LNCS 7358, Springer (2012). https://doi.org/10.1007/978-3-642-31424-7_23

[5] Clarke, E.M., Kröning, D., Lerda, F.: A tool for checking ANSI-C programs. In: Proc. TACAS. pp. 168–176. LNCS 2988, Springer (2004). https://doi.org/10.1007/978-3-540-24730-2_15

[6] Donaldson, A.F., Haller, L., Kröning, D., Rümmer, P.: Software verification using k-induction. In: Proc. SAS. pp. 351–368. LNCS 6887, Springer (2011). https://doi.org/10.1007/978-3-642-23702-7_26

References iii

[7] McMillan, K.L.: Interpolation and SAT-based model checking. In: Proc. CAV. pp. 1–13. LNCS 2725, Springer (2003). https://doi.org/10.1007/978-3-540-45069-6_1

[8] Sheeran, M., Singh, S., Stålmarck, G.: Checking safety properties using induction and a SAT-solver. In: Proc. FMCAD, pp. 127–144. LNCS 1954, Springer (2000). https://doi.org/10.1007/3-540-40922-X_8

[9] Vizel, Y., Grumberg, O.: Interpolation-sequence based model checking. In: Proc. FMCAD. pp. 1–8. IEEE (2009). https://doi.org/10.1109/FMCAD.2009.5351148

References iv

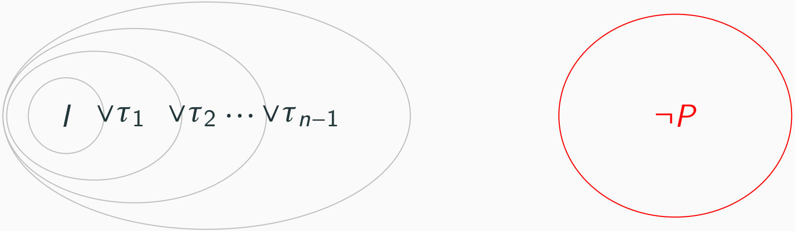
[10] Vizel, Y., Grumberg, O., Shoham, S.: Intertwined forward-backward reachability analysis using interpolants. In: Proc. TACAS. pp. 308–323. LNCS 7795, Springer (2013). https://doi.org/10.1007/978-3-642-36742-7_22

IMC - Fixed Point

- BMC check: $\tau_i(s_0) \wedge T(s_0, s_1) \cdots \wedge T(s_{k-1}, s_k) \wedge P(s_k) \rightarrow$ UNSAT
- Repeat until $I \vee \bigvee \tau_i$ becomes a fixed point

IMC - Fixed Point

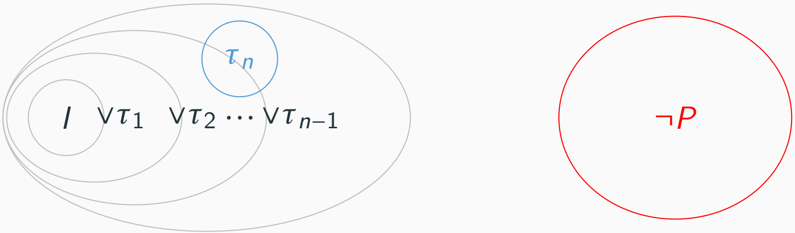
- BMC check: $\tau_i(s_0) \wedge T(s_0, s_1) \cdots \wedge T(s_{k-1}, s_k) \wedge P(s_k) \rightarrow$ UNSAT
- Repeat until $I \vee \bigvee \tau_i$ becomes a fixed point



IMC - Fixed Point

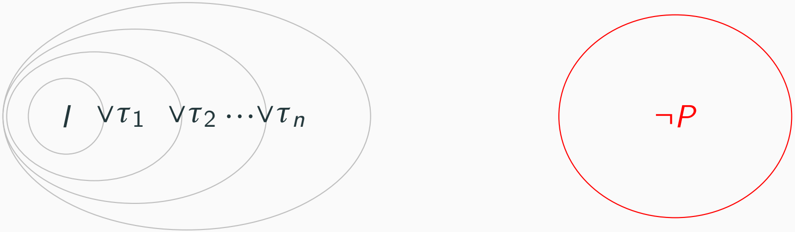
- BMC check: $\tau_i(s_0) \land T(s_0, s_1) \cdots \land T(s_{k-1}, s_k) \land P(s_k) \rightarrow$ UNSAT
- Repeat until $I \lor \bigvee \tau_i$ becomes a fixed point

$\tau_n \Rightarrow I \lor \bigvee_{i=1}^{n-1} \tau_i$ holds



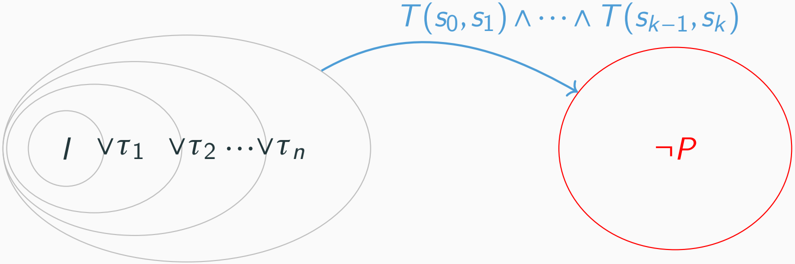
IMC - Unrolling for BMC Check

- Increment unrolling bound k for BMC check



IMC - Unrolling for BMC Check

- Increment unrolling bound k for BMC check



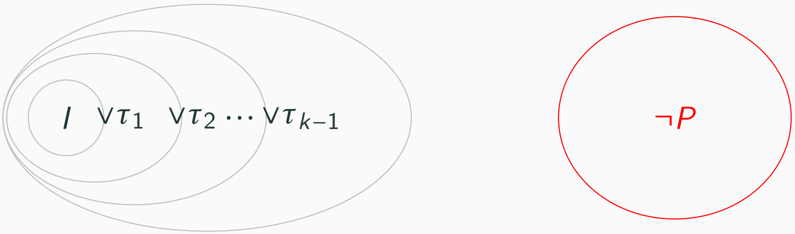
$T(s_0, s_1) \wedge \cdots \wedge T(s_{k-1}, s_k)$
$I \ \vee\tau_1 \ \vee\tau_2 \cdots \vee\tau_n$
$\neg P$

ISMC - Fixed Point

- BMC check: $I(s_0) \wedge T(s_0, s_1) \wedge \cdots \wedge T(s_{k-1}, s_k) \wedge P(s_k) \rightarrow \text{UNSAT}$
- Directly construct the whole sequence

ISMC - Fixed Point

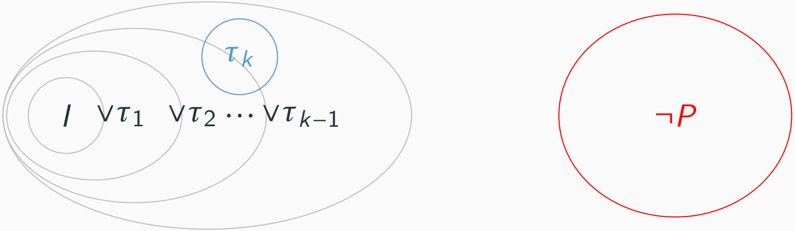
- BMC check: $I(s_0) \wedge T(s_0, s_1) \wedge \cdots \wedge T(s_{k-1}, s_k) \wedge P(s_k) \rightarrow$ UNSAT
- Directly construct the whole sequence



ISMC - Fixed Point

- BMC check: $I(s_0) \wedge T(s_0, s_1) \wedge \cdots \wedge T(s_{k-1}, s_k) \wedge P(s_k) \rightarrow$ UNSAT
- Directly construct the whole sequence

$\tau_k \Rightarrow I \vee \bigvee_{i=1}^{k-1} \tau_i$ holds

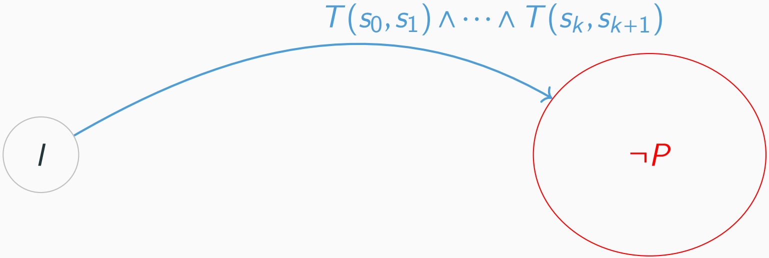


ISMC - Unrolling for BMC Check

- In the next step unroll to $k + 1$

ISMC - Unrolling for BMC Check

- In the next step unroll to $k+1$

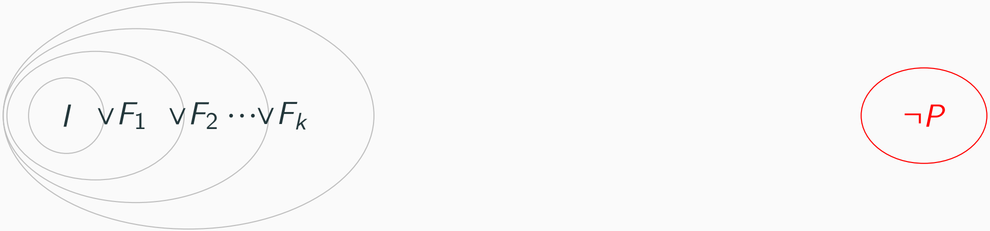


DAR - Fixed Point

- Local BMC checks: $F_i(s_0) \wedge T(s_0, s_1) \wedge B_{k-i-1}(s_1) \rightarrow$ UNSAT

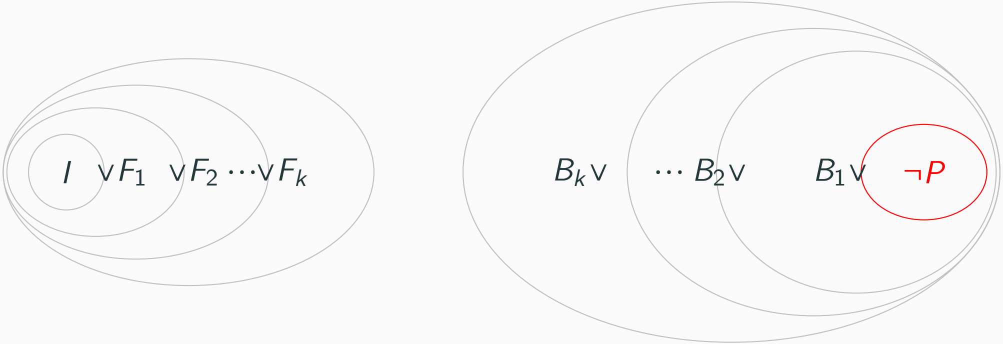
DAR - Fixed Point

- Local BMC checks: $F_i(s_0) \wedge T(s_0, s_1) \wedge B_{k-i-1}(s_1) \rightarrow$ UNSAT



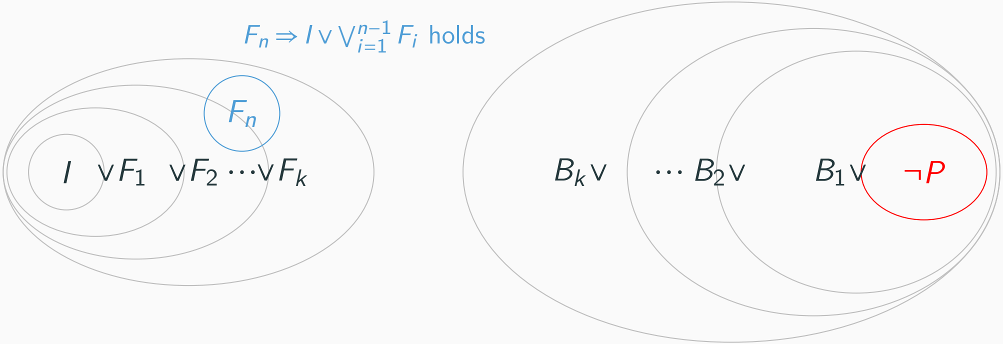
DAR - Fixed Point

- Local BMC checks: $F_i(s_0) \wedge T(s_0, s_1) \wedge B_{k-i-1}(s_1) \rightarrow \text{UNSAT}$



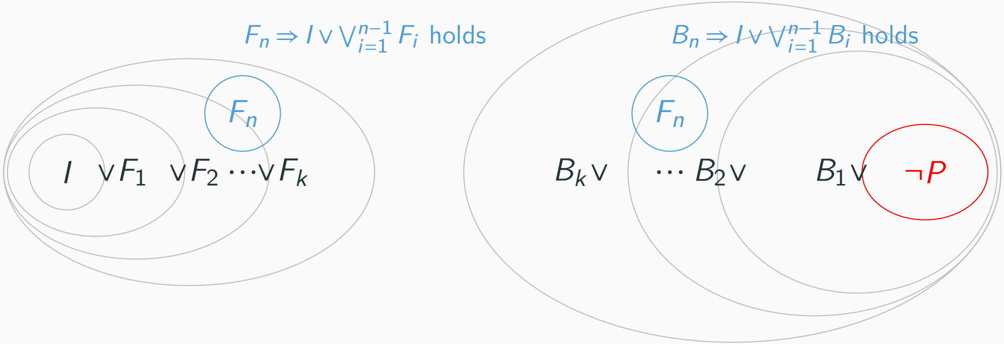
DAR - Fixed Point

- Local BMC checks: $F_i(s_0) \wedge T(s_0, s_1) \wedge B_{k-i-1}(s_1) \rightarrow$ UNSAT


$F_n \Rightarrow I \vee \bigvee_{i=1}^{n-1} F_i$ holds
F_n
I
$\vee F_1$
$\vee F_2 \cdots \vee F_k$
$B_k \vee$
$\cdots B_2 \vee$
$B_1 \vee$
$\neg P$

DAR - Fixed Point

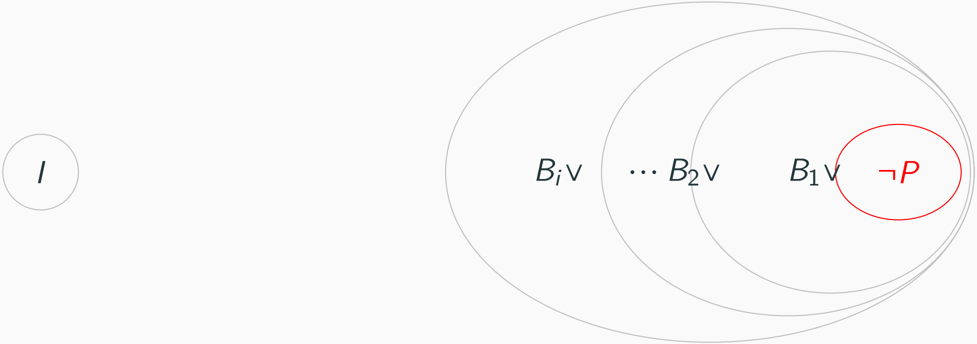
- Local BMC checks: $F_i(s_0) \land T(s_0, s_1) \land B_{k-i-1}(s_1) \rightarrow$ UNSAT



$F_n \Rightarrow I \lor \bigvee_{i=1}^{n-1} F_i$ holds
F_n
I
$\lor F_1$
$\lor F_2 \cdots \lor F_k$
$B_n \Rightarrow I \lor \bigvee_{i=1}^{n-1} B_i$ holds
F_n
$B_k \lor$
$\cdots B_2 \lor$
$B_1 \lor$
$\neg P$

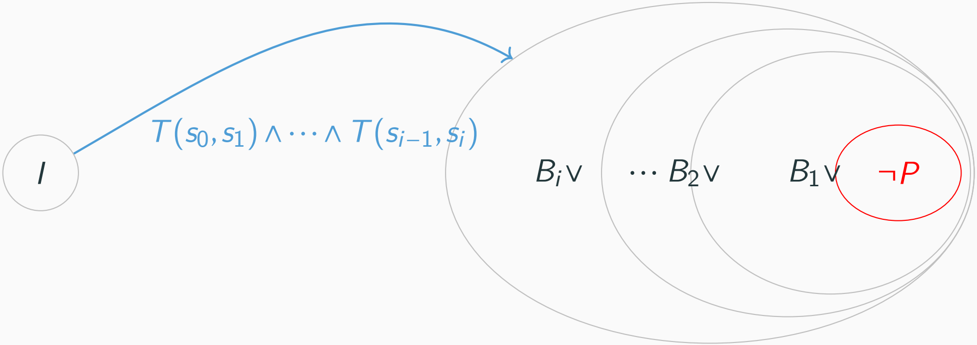
DAR - Unrolling for BMC Check

- Global BMC checks



DAR - Unrolling for BMC Check

- Global BMC checks


I
$T(s_0, s_1) \wedge \cdots \wedge T(s_{i-1}, s_i)$
$B_i \vee$
$\cdots B_2 \vee$
$B_1 \vee$
$\neg P$