

BMC
ITP
K-ind

Incremental, Inductive Model Checking

Aaron Bradley

June 27, 2013

Notation

$S : (\bar{x}, \bar{i}, I(\bar{x}), T(\bar{x}, \bar{i}, \bar{x}'))$ Invariant property : P

- ▶ \bar{x} : State variables
- ▶ \bar{i} : Inputs
- ▶ $I(\bar{x})$: Initial condition
- ▶ $T(\bar{x}, \bar{i}, \bar{x}')$: Transition relation
- ▶ $P(\bar{x})$: Invariant property (“good states”)

Problem: Show all reachable states satisfy P

SAT-Based Model Checking:

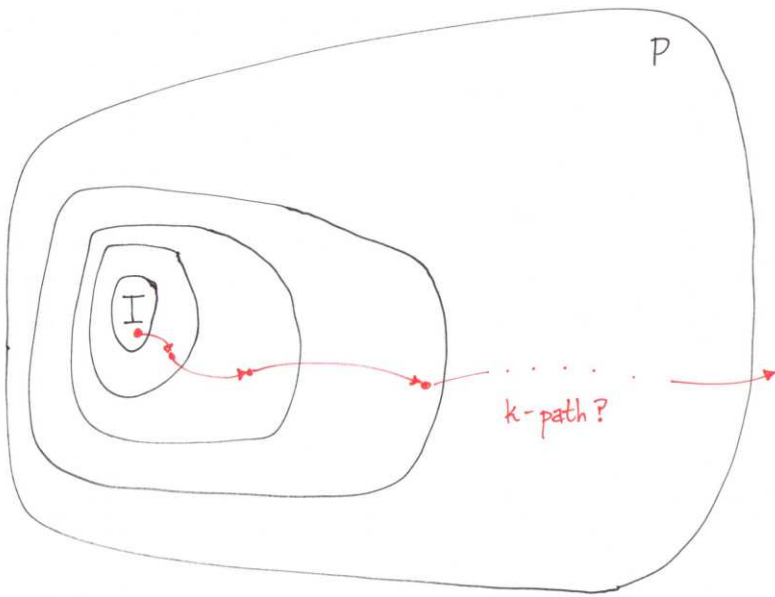
Just Unroll

Bounded Model Checking (BMC)

For $k = 0, 1, 2, \dots$, SAT query:

$$I(\bar{x}_0) \wedge \bigwedge_{j=1}^k T(\bar{x}_{j-1}, \bar{i}_{j-1}, \bar{x}_j) \wedge \neg P(\bar{x}_k)$$

until an error is found or the diameter is reached.



BMC

P

I

k-path?

Induction

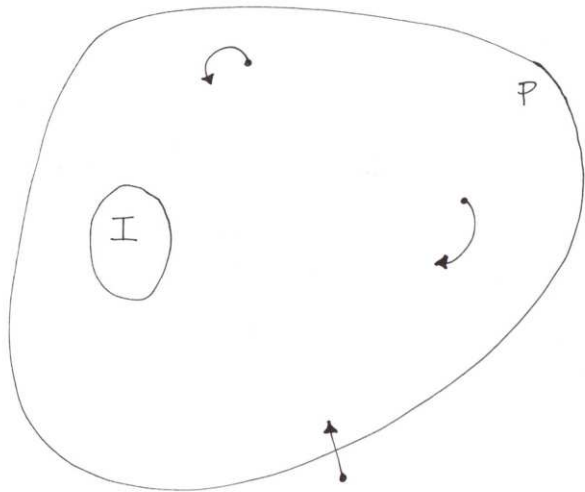
Mathematical induction over S :

$$\blacktriangleright I(\bar{x}) \Rightarrow P(\bar{x}) \quad (\textit{initiation})$$

$$\blacktriangleright P(\bar{x}) \wedge T(\bar{x}, \bar{i}, \bar{x}') \Rightarrow P(\bar{x}') \quad (\textit{consecution})$$

Failure does not imply that P does not hold.

Inductive strengthening: F such that $F \wedge P$ is inductive



P is inductive

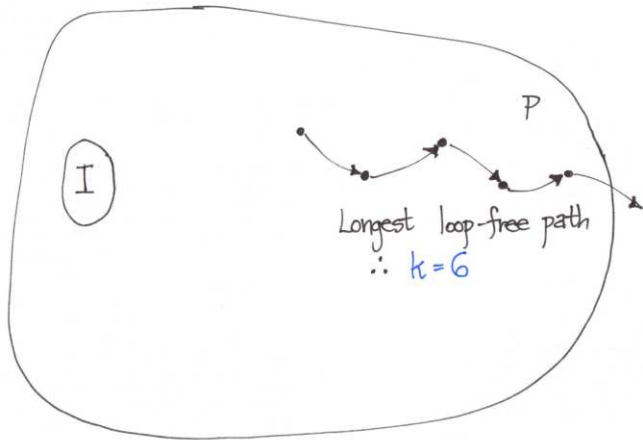
k-Induction

Initiation: (BMC)

$$I(\bar{x}_0) \wedge \bigwedge_{j=1}^k T(\bar{x}_{j-1}, \bar{i}_{j-1}, \bar{x}_j) \Rightarrow P(\bar{x}_k)$$

Consecution:

$$\text{LoopFree} \wedge \bigwedge_{j=1}^k (P(\bar{x}_{j-1}) \wedge T(\bar{x}_{j-1}, \bar{i}_{j-1}, \bar{x}_j)) \Rightarrow P(\bar{x}_k)$$



k-Induction

Interpolant-based Model Checking (ITP)

Post-condition operator:

$$\text{post}(F)(\bar{x}) = \exists \bar{x}_0, \bar{i}_0. F(\bar{x}_0) \wedge T(\bar{x}_0, \bar{i}_0, \bar{x})$$

Abstract post-condition operator:

$$\text{post}(F)(\bar{x}) \Rightarrow \widehat{\text{post}}(F)(\bar{x})$$

Interpolant-based Model Checking (ITP)

If this query is UNSAT

$$F(\bar{x}_0) \wedge \bigwedge_{j=1}^k T(\bar{x}_{j-1}, \bar{i}_{j-1}, \bar{x}_j) \Rightarrow P(\bar{x}_k)$$

then extract G such that

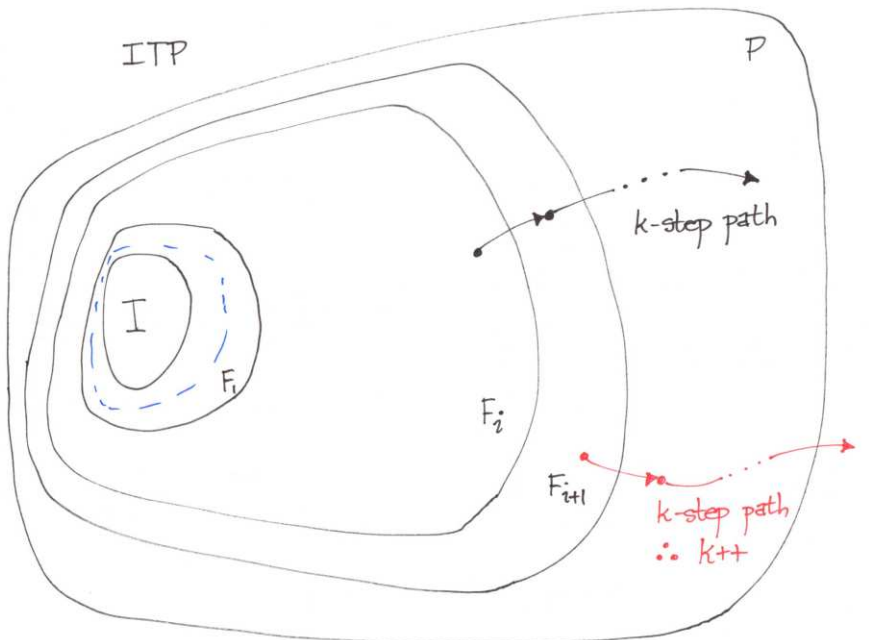
$$F(\bar{x}_0) \wedge T(\bar{x}_0, \bar{i}_0, \bar{x}_1) \Rightarrow G(\bar{x}_1)$$

and

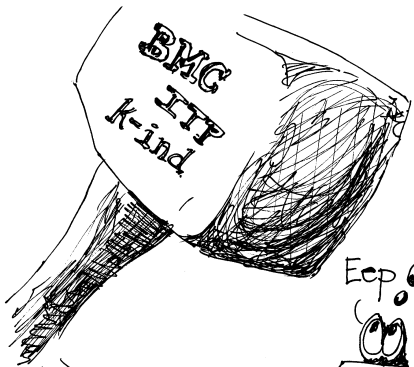
$$G(\bar{x}_1) \wedge \bigwedge_{j=2}^k T(\bar{x}_{j-1}, \bar{i}_{j-1}, \bar{x}_j) \Rightarrow P(\bar{x}_k)$$

Then

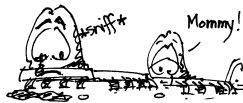
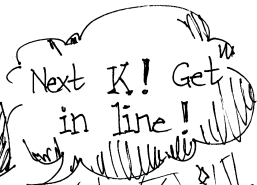
$$\widehat{\text{post}}(F)(\bar{x}) := G(\bar{x})$$

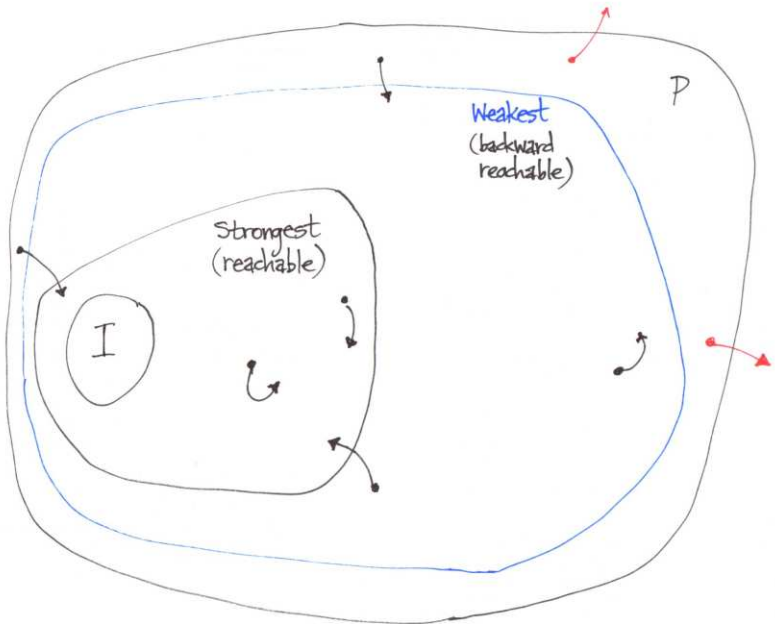


SAT-Based Model Checking: *Don't Unroll!*

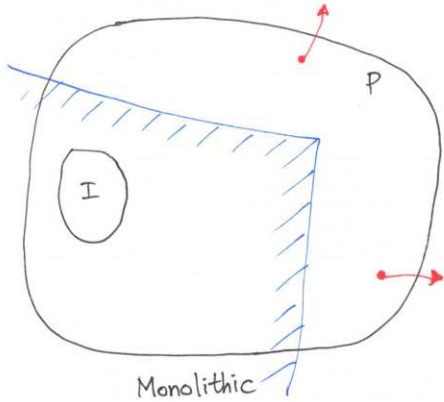
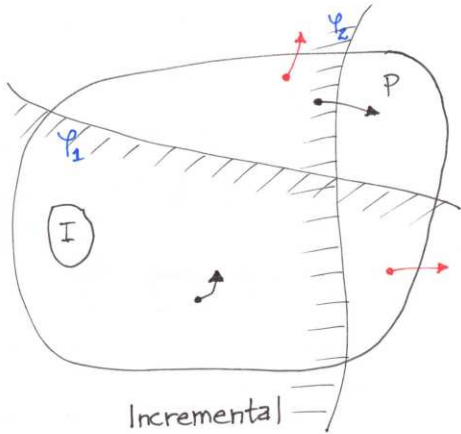


Eep!

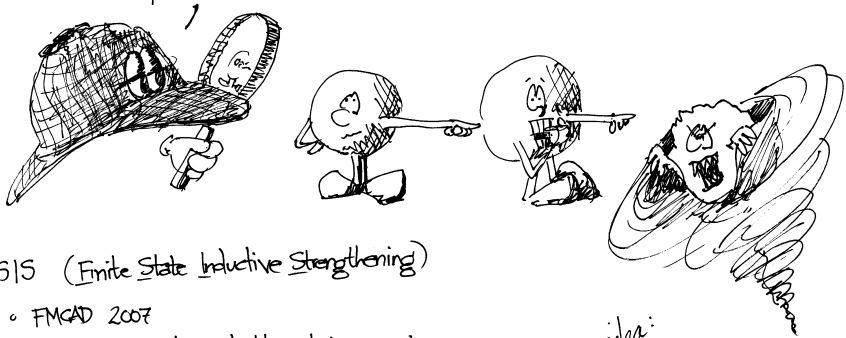




Inductive Strengthening



Impossible?
Yes, but is there
a simple reason?



FSIS (Finite State Inductive Strengthening)

- FMCAD 2007
- From backward reachable state s to $G \subseteq \neg I$ s.t.

$$F \wedge G \wedge I \Rightarrow G' \quad (\text{and } I \Rightarrow G)$$

- On top of explicit backward enumeration.

Lasting idea:

Relative
Inductive
Generalization

Inductive Generalization

Given: cube s (usually based on backward-reachable state)

Find: $c \subseteq \neg s$ such that

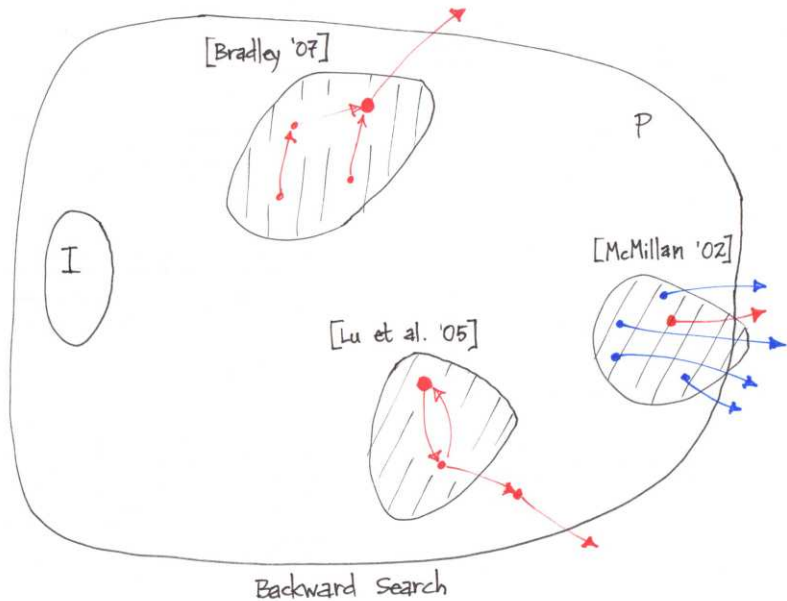
- ▶ Initiation:

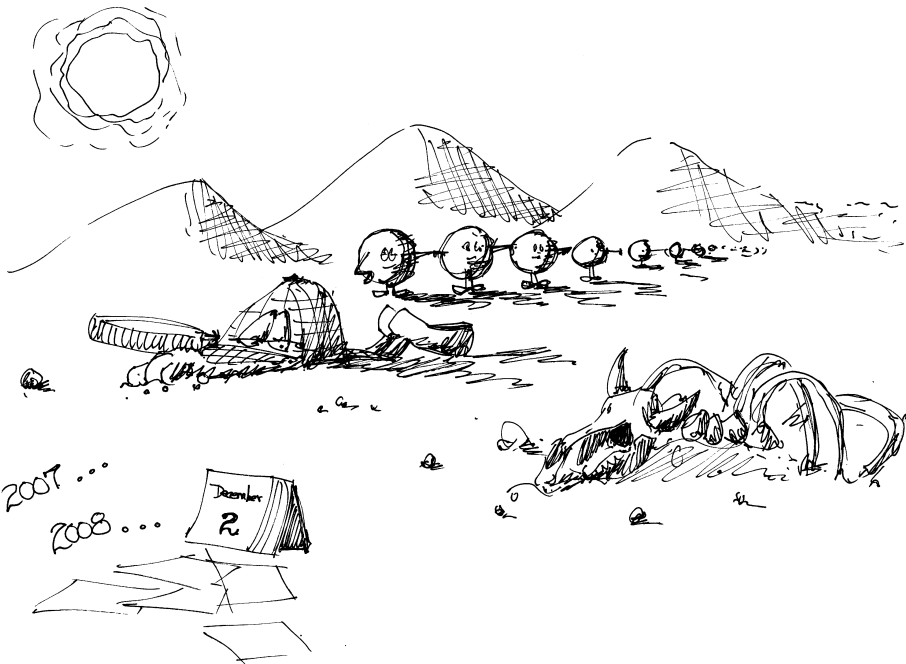
$$I(\bar{x}) \Rightarrow c(\bar{x})$$

- ▶ Consecution (relative to information G):

$$G(\bar{x}) \wedge c(\bar{x}) \wedge T(\bar{x}, \bar{i}, \bar{x}') \Rightarrow c(\bar{x}')$$

- ▶ Minimality: No strict subclause of c is inductive relative to G





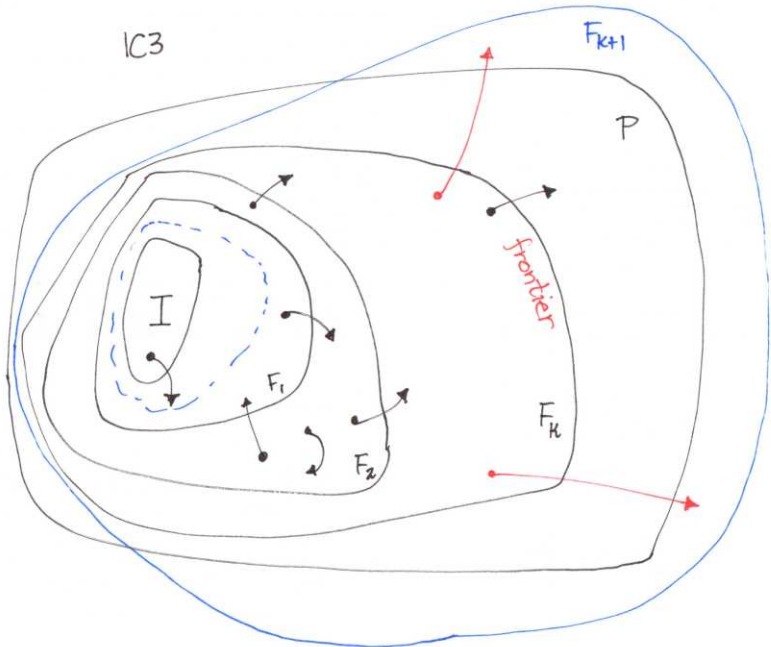
Use **inductive generalization** to **incrementally** construct over-approximating sets.

F_i : over-approximates set of states reachable in at most i steps

Four invariants:

1. $I(\bar{x}) \Rightarrow F_0(\bar{x})$
2. $\forall i. F_i(\bar{x}) \Rightarrow F_{i+1}(\bar{x})$
3. $\forall i. F_i(\bar{x}) \wedge T(\bar{x}, \bar{i}, \bar{x}') \Rightarrow F_{i+1}(\bar{x}')$
4. $\forall i \leq k. F_i(\bar{x}) \Rightarrow P(\bar{x})$

¹*Incremental Construction of Inductive Clauses for Indubitable Correctness*
Sometimes called *Property Directed Reachability* (PDR)



IC3

F_{k+1}

P

I

F_1

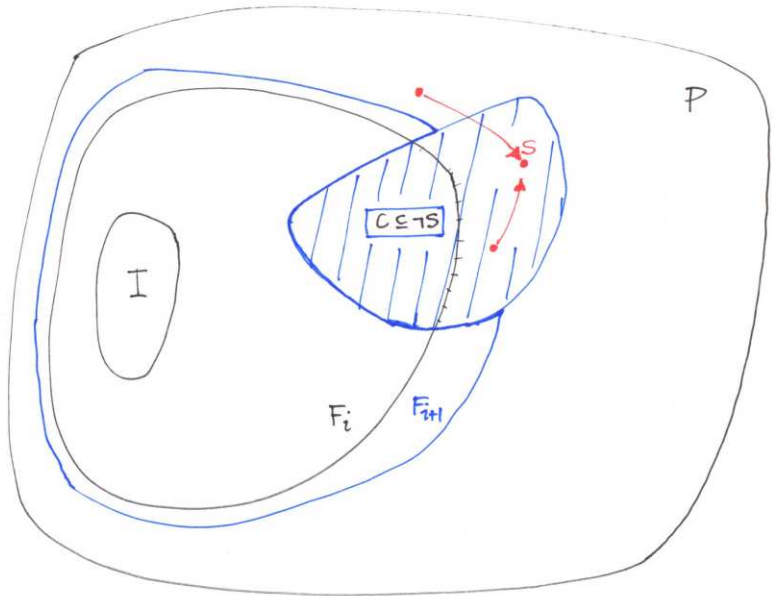
F_2

F_k

frontier

Refinement: In response to **proof obligation** $\langle s, j \rangle$,

- ▶ Attempt inductive generalization relative to F_j : $c \subseteq \neg s$
- ▶ Success: Conjoin c to F_1, \dots, F_{j+1}
- ▶ Failure:
 - ▶ Predecessor t
 - ▶ Enqueue new obligation $\langle t, j - 1 \rangle$

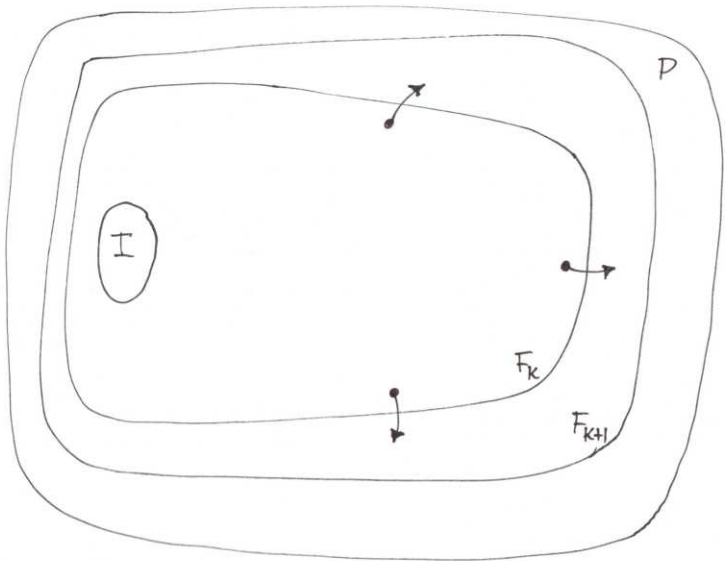


Inductive Generalization

When

$$F_k \wedge T(\bar{x}, \bar{i}, \bar{x}') \Rightarrow P(\bar{x}')$$

- ▶ Propagate clauses forward with relative induction
- ▶ Increment k (unless converged)



$$F_k \wedge P \wedge T \Rightarrow P'$$

Converges when $\exists j \leq k. F_j = F_{j+1}$. Then:

1. $I(\bar{x}) \Rightarrow F_j(\bar{x})$
2. $F_j(\bar{x}) \wedge T(\bar{x}, \bar{i}, \bar{x}') \Rightarrow F_j(\bar{x}')$
3. $F_j(\bar{x}) \Rightarrow P(\bar{x})$

$\therefore F_j$ is an inductive strengthening of P .

Research Inspired by IC3:
Incremental, Inductive Model Checking

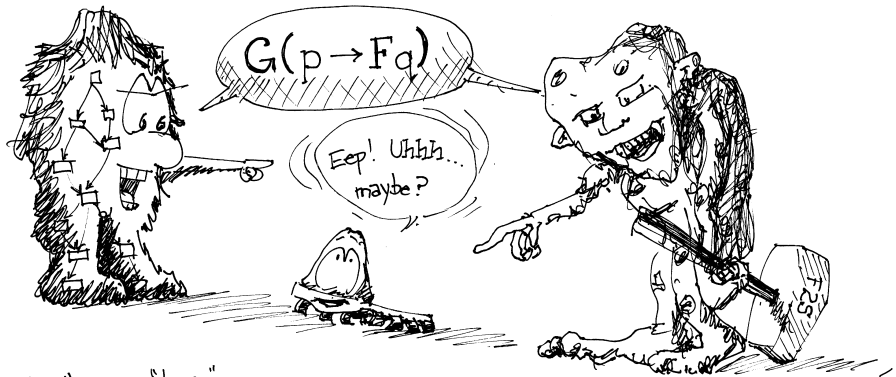
Improvements/Extensions to IC3

- ▶ Lift predecessor state s to set of predecessors \bar{s} :
 - ▶ with kCOI, statically (original paper)
 - ▶ with ternary simulation [Een et al. '11]
 - ▶ with SAT [Chockler et al. '11]
- ▶ Improve proofs [Bradley et al. '11]
 - ▶ Strengthen, weaken, shrink
 - ▶ Used in FAIR, IICTL
- ▶ Apply IC3 in design/verify cycle [Chockler et al. '11]
 - ▶ Extract inductive core from previous run
 - ▶ Accelerate analysis of mutated design or similar property
- ▶ Improve generalization [Hassan et al. '13]
 - ▶ Apply inductive generalization to counterexamples to generalization (CTGs)
 - ▶ Not just explicitly discovered backward reachable states
 - ▶ Essentially uniform improvement \therefore better

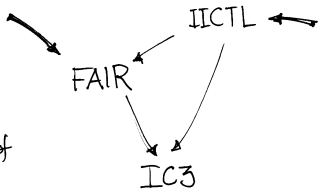
Localization Reduction

- ▶ Extract information from incomplete concrete run to guide refinements [Baumgartner et al. '12]
 - ▶ Level at which variable is first used
 - ▶ Reduction in abstract model size in practice
- ▶ Lazy abstraction [Vizel et al. '12]
 - ▶ Visible variables abstraction $U_0 \subseteq U_1 \subseteq \dots \subseteq U_k$
 - ▶ Refinement: run IC3 on concrete model at k
 - ▶ Then use unsat core of $F_i \wedge T \Rightarrow F'_{i+1}$ to derive new U_i

IC3 \Rightarrow IIV (Incremental, Inductive Verification)



- Hypothesize "lasso"
"skeleton"
- Attempt to "flesh out"
- Failure explained by inductive proof...
- ... that refines space of hypotheses



- "Local" style
- Proof-based generalization
- Lifting-based generalization

LTL (ω -regular) Model Checking [Bradley et al. '11]

- ▶ Search for lasso as usual
- ▶ Top-level SAT query:
 - ▶ Find set of states in one “arena” that satisfy all Büchi conditions
 - ▶ If UNSAT, property holds
- ▶ Reachability queries to connect states:
 - ▶ **Stem**: From initial state to one of states
 - ▶ **Cycle**: From state to state
- ▶ Refinement from inductive strengthenings:
 - ▶ **Stem**: Global reachability
 - ▶ **Cycle**: Transection of state space—**loop must be on one side**

CTL Model Checking [Hassan et al. '12]

- ▶ “Local” method + generalization
- ▶ Incrementally refine lower/upper bounds on subformulas
- ▶ Generalize from queries involving explicit states:
 - ▶ $EX\psi$: SAT (unsat core)
 - ▶ $EF\psi$: reachability, e.g., IC3 (inductive strengthening)
 - ▶ $EG\psi$: constrained cycle, e.g., FAIR (inductive strengthening)
- ▶ Generalize traces with aggressive lifting

Other decidable domains

- ▶ Timed systems [Hoder et al. '12, Kindermann et al. '12]
- ▶ Petri nets (and more general) [Kloos et al. '13]
- ▶ Finite-state safety games [Morgenstern '13]

IC3 with SMT

- ▶ Combination with lazy abstraction [Cimatti et al. '12]
- ▶ Constrained Horn Clauses [Hoder et al. '12]
- ▶ Polyhedra [Welp et al. '13]

Combination with ITP

Use inductive generalization to locally construct interpolant
[Vizel et al. '13]

Conclusion

Main ideas:

- ▶ Induction as a mechanism for generalization
- ▶ Incremental, local (state-triggered) reasoning

Complements monolithic reasoning, which sometimes wins



Thanks! Questions?