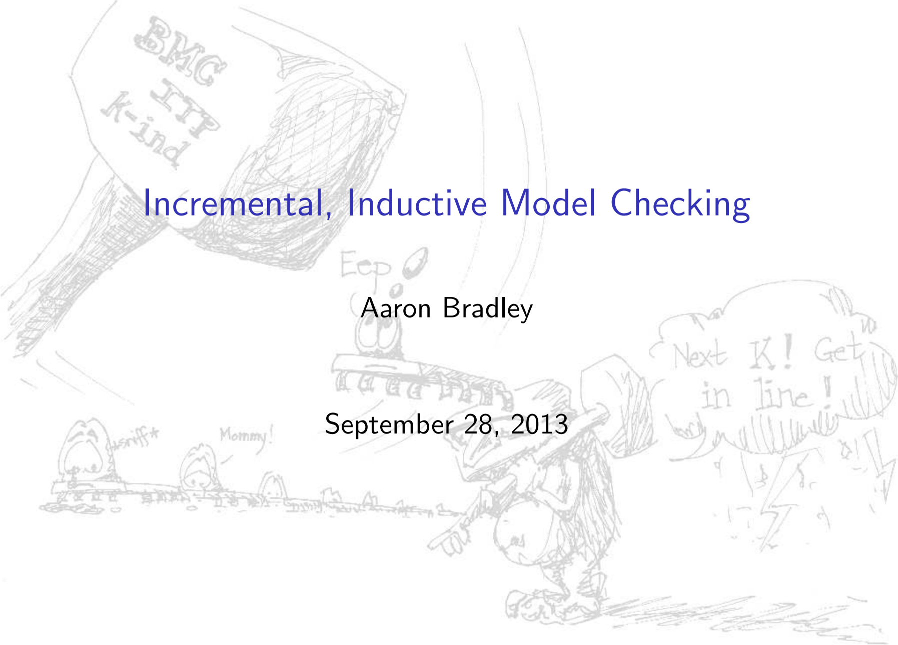


Incremental, Inductive Model Checking

Aaron Bradley

September 28, 2013

Notation

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

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

Problem: Show all reachable states satisfy P

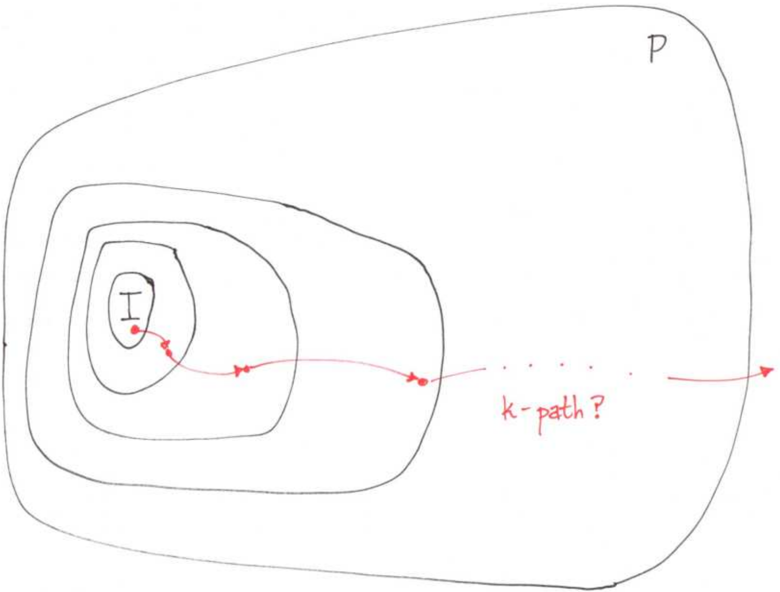
SAT-Based Model Checking:
Just Unroll

Bounded Model Checking (BMC)

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

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

until an error is found or the diameter is reached.



BMC

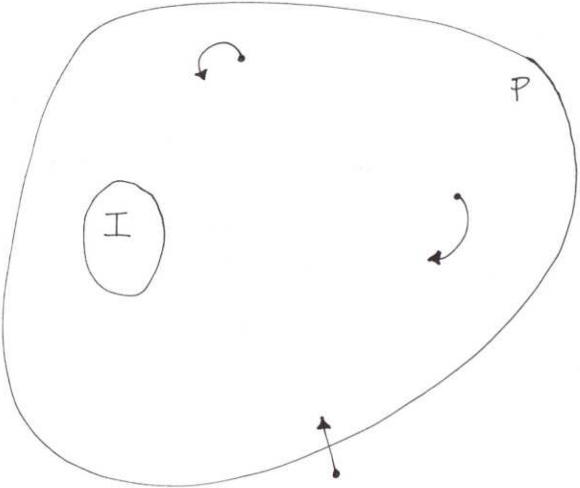
Induction

Mathematical induction over S:

- $I(\overline{x}) \Rightarrow P(\overline{x})$ (*initiation*)
- $P(\overline{x}) \wedge T(\overline{x}, \overline{i}, \overline{x}') \Rightarrow P(\overline{x}')$ (*consecution*)

Failure does not imply that P does not hold.

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



P is inductive

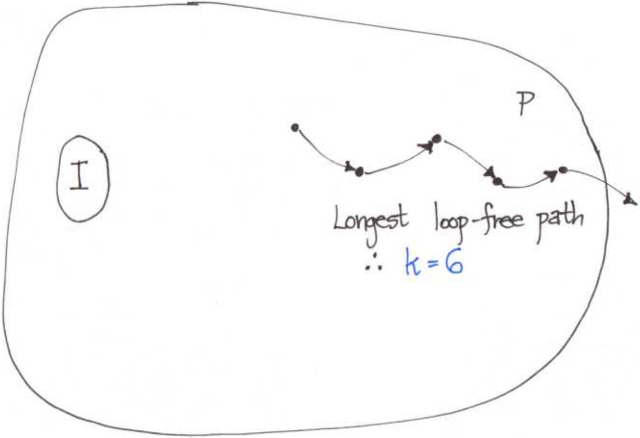
k-Induction

Initiation: (BMC)

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

Consecution:

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



k-Induction

Interpolant-based Model Checking (ITP)

Post-condition operator:

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

Abstract post-condition operator:

$$\mathrm{post}(F)(\overline{x}) \Rightarrow \widehat{\mathrm{post}}(F)(\overline{x})$$

Interpolant-based Model Checking (ITP)

If this query is UNSAT

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

then extract G such that

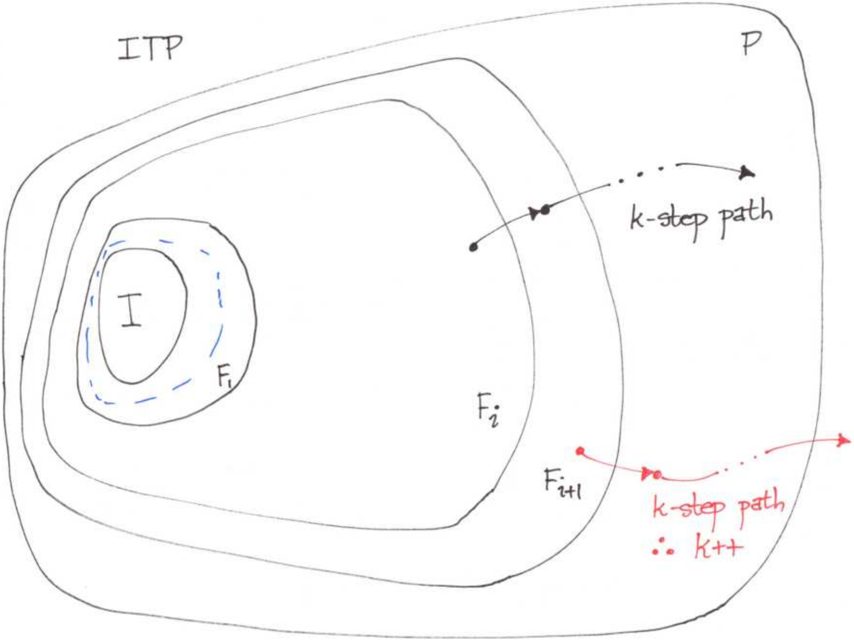
$$F(\overline{x}_0) \;\wedge\; T(\overline{x}_0, \overline{i}_0, \overline{x}_1) \;\Rightarrow\; G(\overline{x}_1)$$

and

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

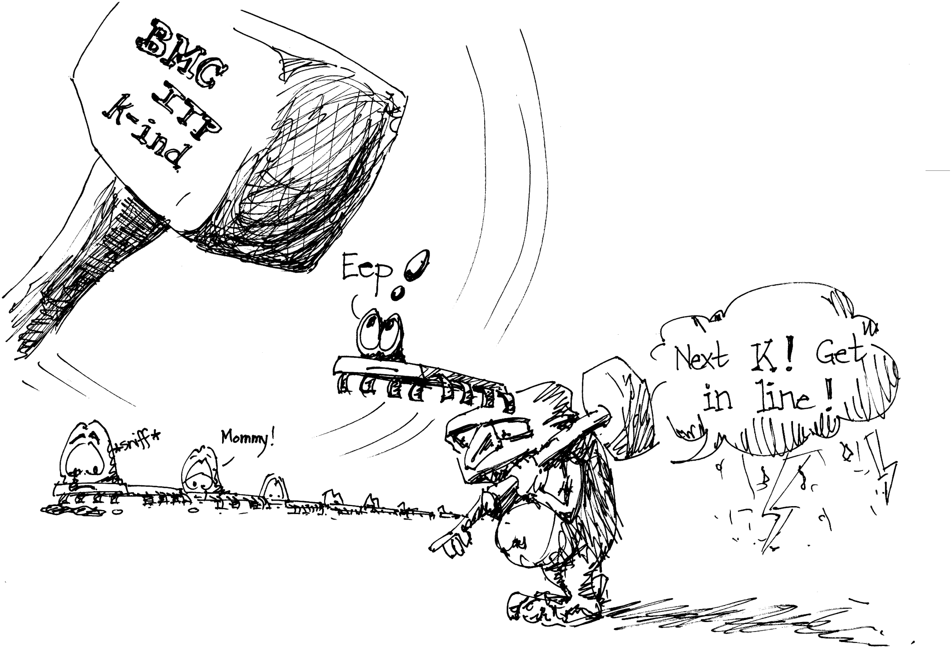
Then

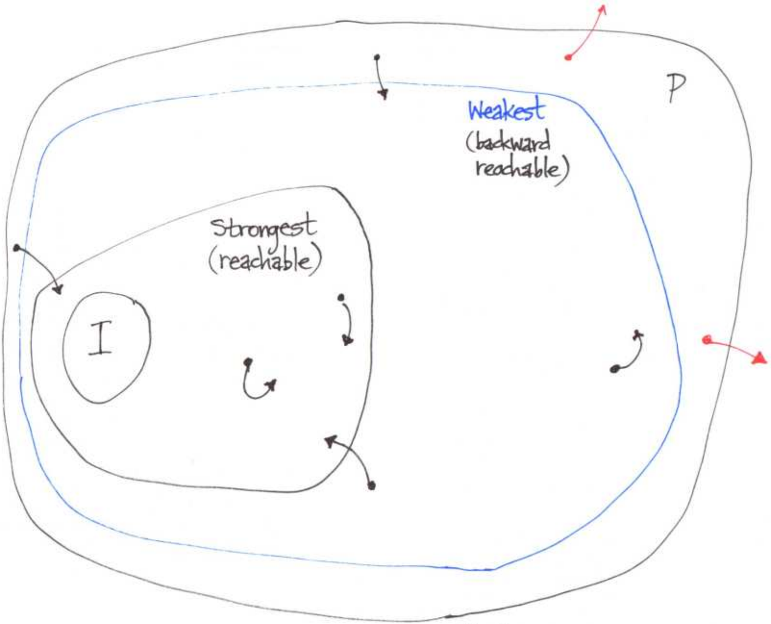
$$\widehat{\text{post}}(F)(\overline{x}) := G(\overline{x})$$



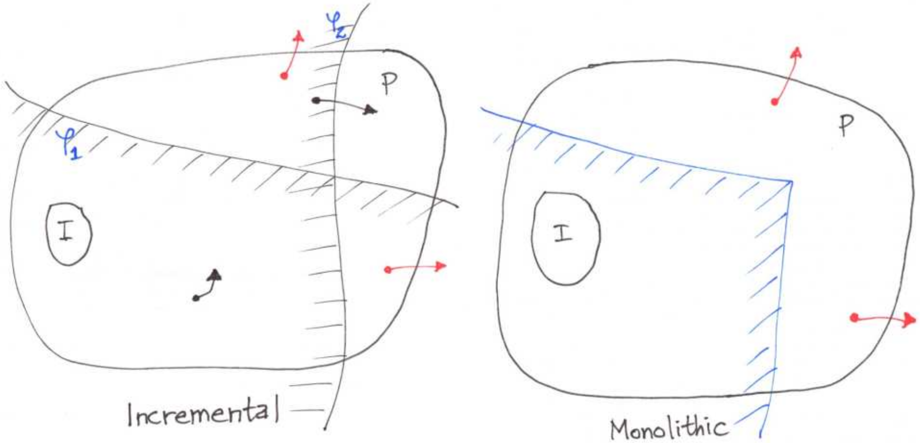
ITP
P
I
F_1
F_i
F_{i+1}
k-step path
k-step path
∴ k++

SAT-Based Model Checking:
Don't Unroll!

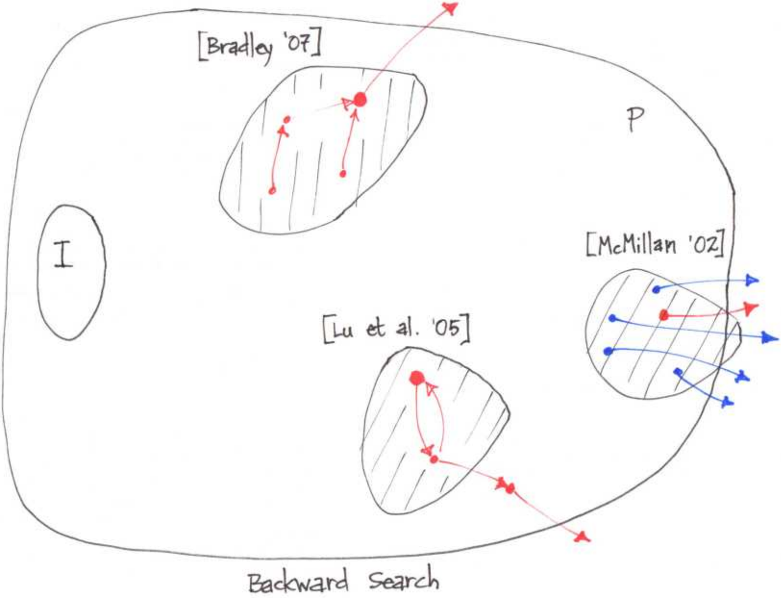

BMC
ITP
K-ind
Eep
sniff
Mommy!
Next K! Get in line!



Inductive Strengthening



φ_2
P
φ_1
I
Incremental
P
I
Monolithic



[Bradley '07]
P
I
[McMillan '02]
[Lu et al. '05]
Backward Search

FSIS (Finite State Inductive Strengthening)

- FMCAD 2007
- From backward reachable state s to

$$C \subseteq \neg s \quad \text{s.t.}$$

$$F \wedge C \wedge T \Rightarrow C' \quad (\text{and } I \Rightarrow C)$$

- 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

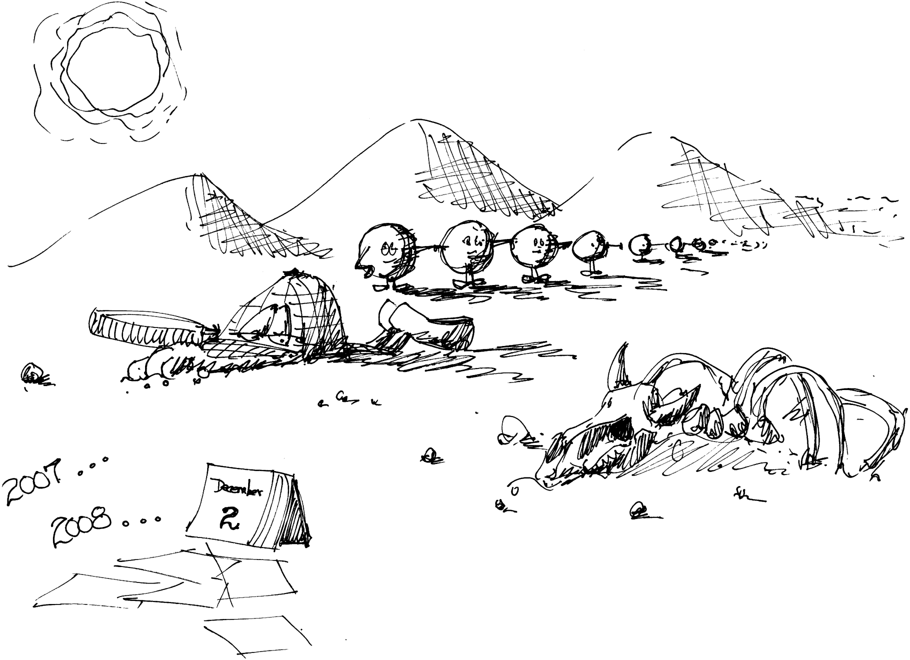
- Initiation:

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

- Consecution (relative to information G):

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

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


2007 . . .
2008 . . .
December
2

IC3[1]

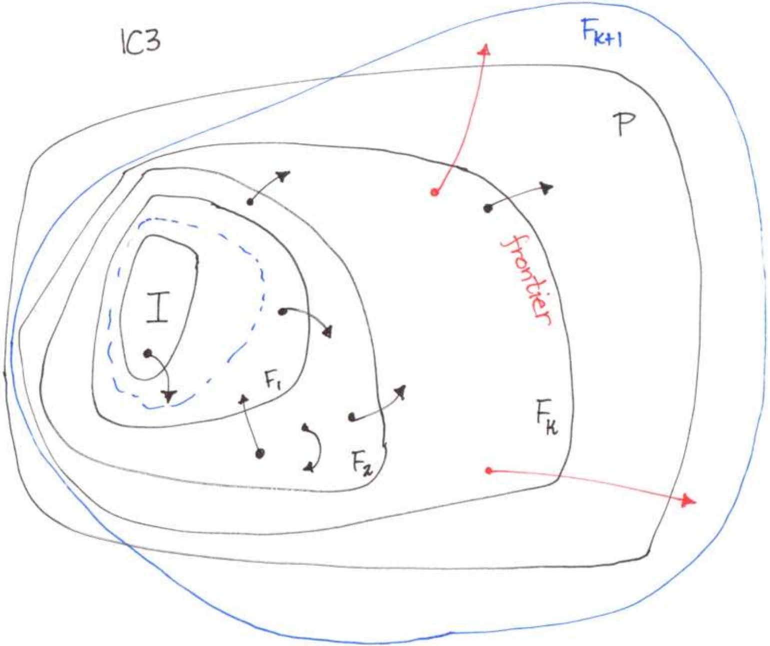
Use inductive generalization to incrementally construct over-approximating sets.

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

Four invariants:

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

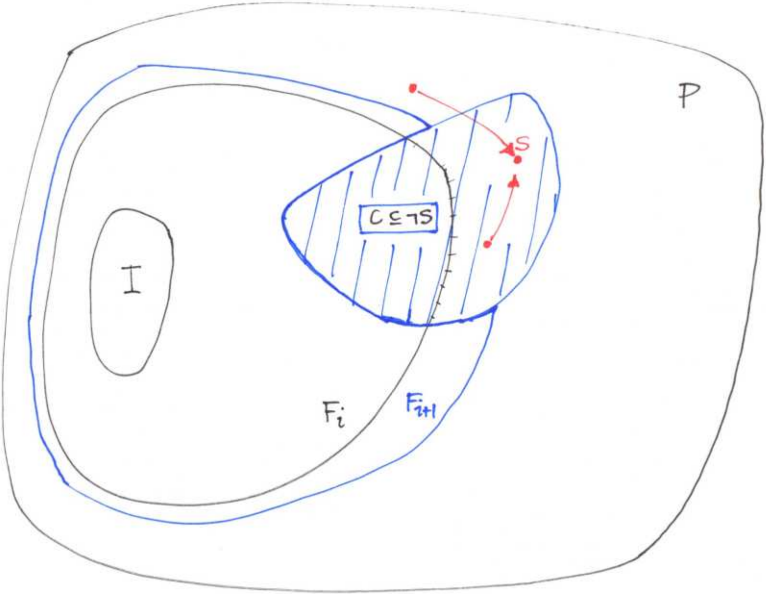
[1] *Incremental Construction of Inductive Clauses for Indubitable Correctness*
Sometimes called *Property Directed Reachability* (PDR)


IC3
Fk+1
P
frontier
I
F1
F2
FK

IC3

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 $F_1, \ldots, F_{j+1}$
- Failure:
  - Predecessor t
  - Enqueue new obligation $\langle t, \ j-1 \rangle$



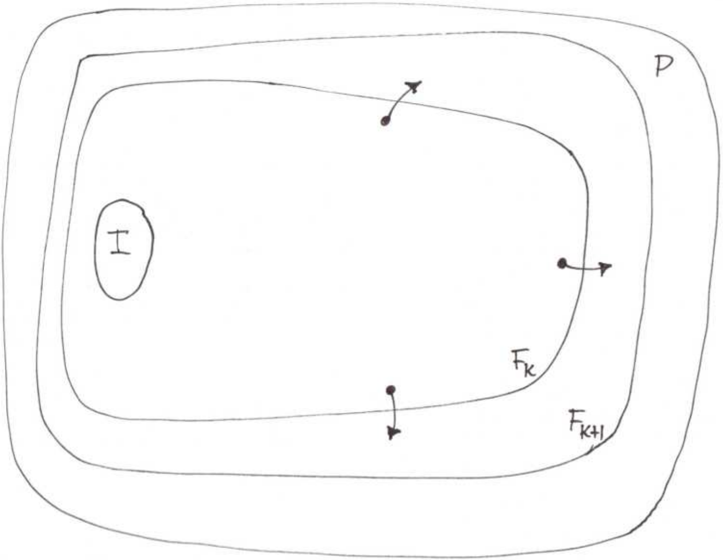
Inductive Generalization

IC3

When

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

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



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

IC3

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

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

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

Research Inspired by IC3:
Incremental, Inductive Model Checking

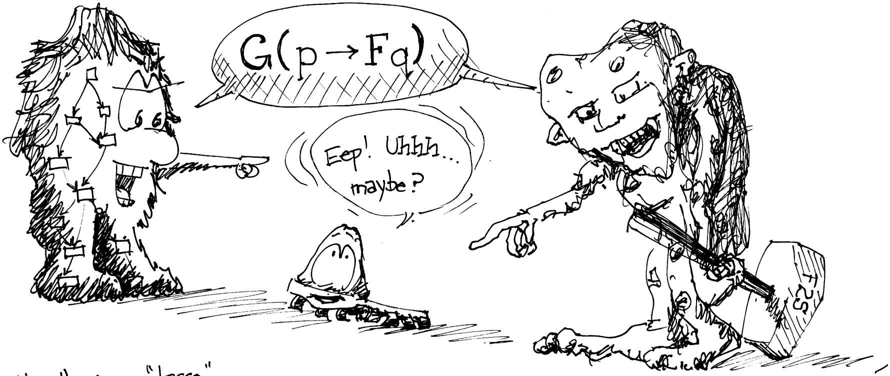
Improvements/Extensions to IC3

- 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 $U_0 \subseteq U_1 \subseteq \cdots \subseteq U_k$
  - 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 ⟹ IIV (Incremental, Inductive Verification)



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

IICTL ←

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

FAIR

IC3

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?