Abstraction and Interpolation

EECS 219C: Formal Methods

Pramod Subramanyan

March 21, 2018

Reviewing BMC

Given M = (S, I, R, L) and LTL property ϕ

We create a satisfiability instance that encodes

- System state transitions:
  - $I(s_0) \wedge R(s_0, s_1) \wedge \ldots \wedge R(s_l, s_{l+1}) \wedge \ldots \wedge R(s_{k-1}, s_k)$
- Set of loop-conditions: $L(l, k) = R(s_k, s_l)$
- Inductively constructed VC for **negation** of ϕ
  - Fϕ: $\pi_0^k \models F\phi = s_0 \Rightarrow \phi \vee \pi_1^k \models F\phi$
  - Gϕ: $\pi_0^k \models G\phi = s_0 \Rightarrow \phi \wedge \pi_1^k \models G\phi$; s_k must loop back to s_l

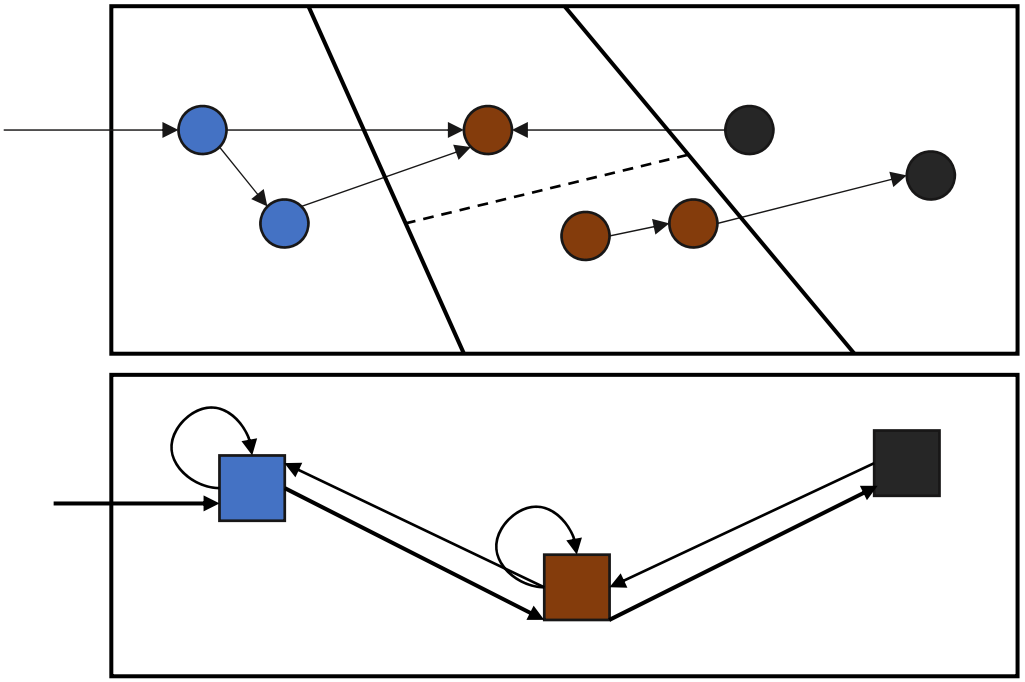
Reviewing Abstraction

Given $M = (S, I, R, L)$, $h: S \to \hat{S}$

Suppose $\widehat{M} = (\hat{S}, \hat{I}, \hat{R}, \hat{L})$ where

- $\hat{I}(\hat{d})$ iff
  $\exists d. \left(h(d) = \hat{d} \wedge I(d)\right)$
- $\hat{R}(\widehat{d_1}, \widehat{d_2})$ iff
  $\exists d_1 d_2. \left(h(d_1) = \widehat{d_1} \wedge h(d_2) = \widehat{d_2} \wedge R(d_1, d_2)\right)$
- $\hat{L}(\hat{d}) = \bigcup_{h(d)=\hat{d}} L(d)$

Abstraction Visualized



Cone-of-Influence Reduction

$$I \doteq x = 0 \wedge y = 0 \wedge z = 0 \wedge u = 0$$
$$R \doteq x' = x + y \wedge y' = y + z \wedge z' = z + 1 \wedge u' = u + 2$$
$$\phi \doteq G(x \geq 0)$$

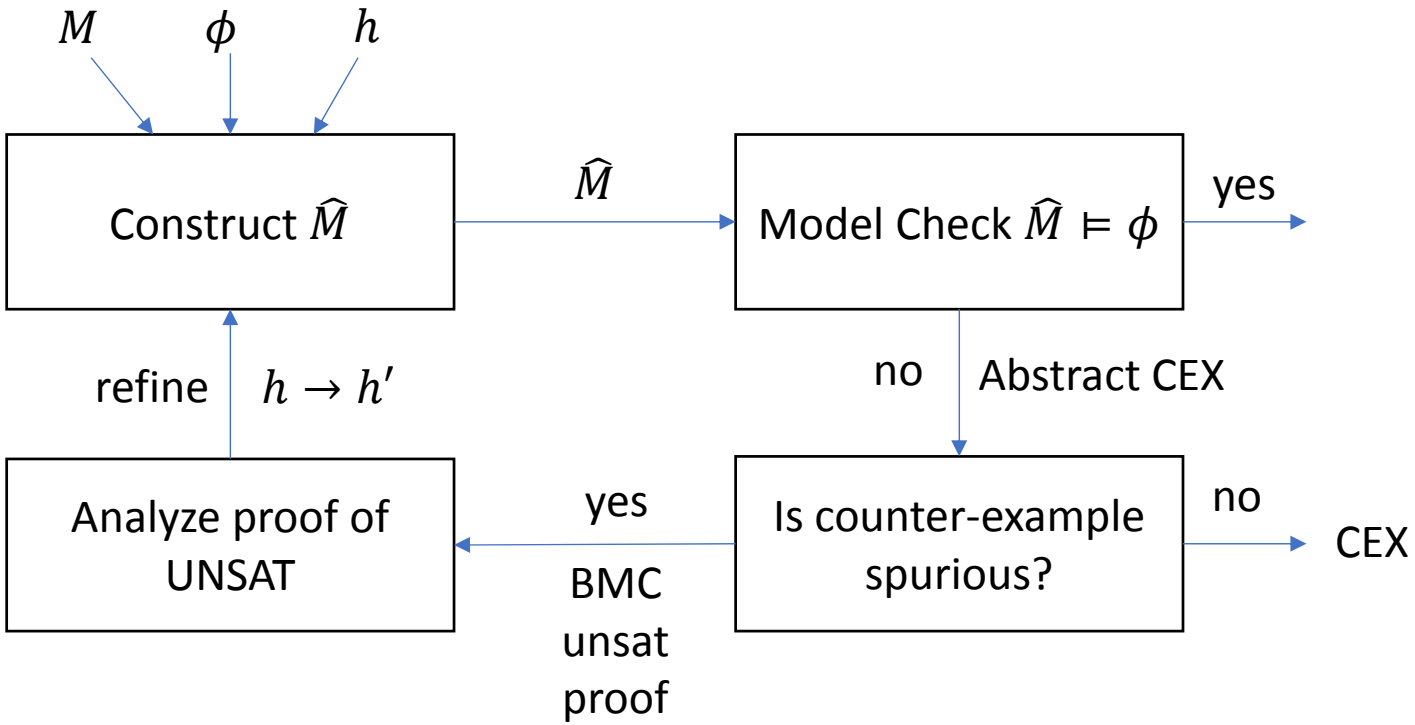
Computing CoI:

$$C_0 = \{x\}, C_1 = \{x, y\}, C_2 = \{x, y, z\}, C_3 = C_2$$

The abstraction is:

$$\hat{I} \doteq x = 0 \wedge y = 0 \wedge z = 0$$
$$\hat{R} \doteq x' = x + y \wedge y' = y + z \wedge z' = z + 1$$

Counter-example Guided Abstraction Refinement



CEGAR: Example

$I \doteq x = 0 \wedge y = 0 \wedge z = 0 \wedge u = 0$
$R \doteq x' = x + y \wedge y' = y + ite(z \geq 0, z, -z) \wedge z' = z + 1$
$\phi \doteq G(x \geq 0)$

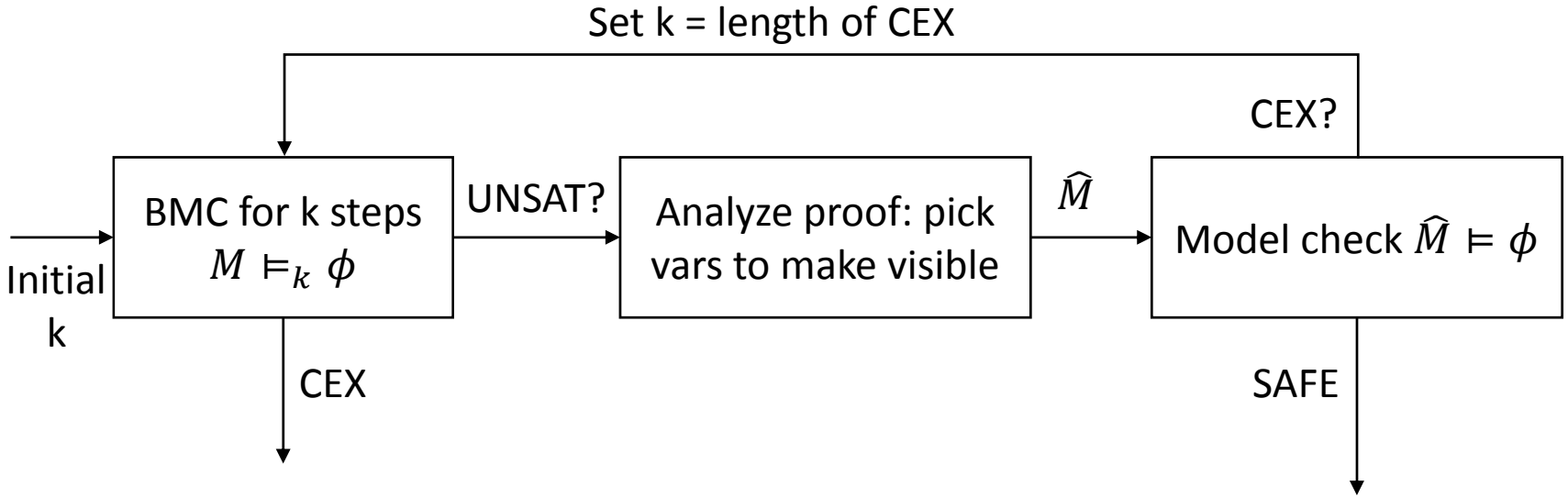
$\hat{I} \doteq x = 0$
$\hat{R} \doteq x' = x + y$

Abstract CEX: (0, 0) → (0,-1)→ (-1,10)

CEX is spurious, as y=0, y’ = -1 is not possible, so we refine

$\hat{I} \doteq x = 0 \wedge y = 0$
$\hat{R} \doteq x' = x + y \wedge y' = y + ite(z \geq 0, z, -z)$

Proof-Based Abstraction



PBA: Example

$I \doteq x = 0 \wedge y = 0 \wedge z = 0 \wedge u = 0$
$R \doteq x' = x + y \wedge y' = y + ite(z \geq 0, z, -z) \wedge z' = z + 1 \wedge u' = u + 2$
$\phi \doteq G(x \geq 0)$

BMC for 2 steps gives us a proof involving x,y,z

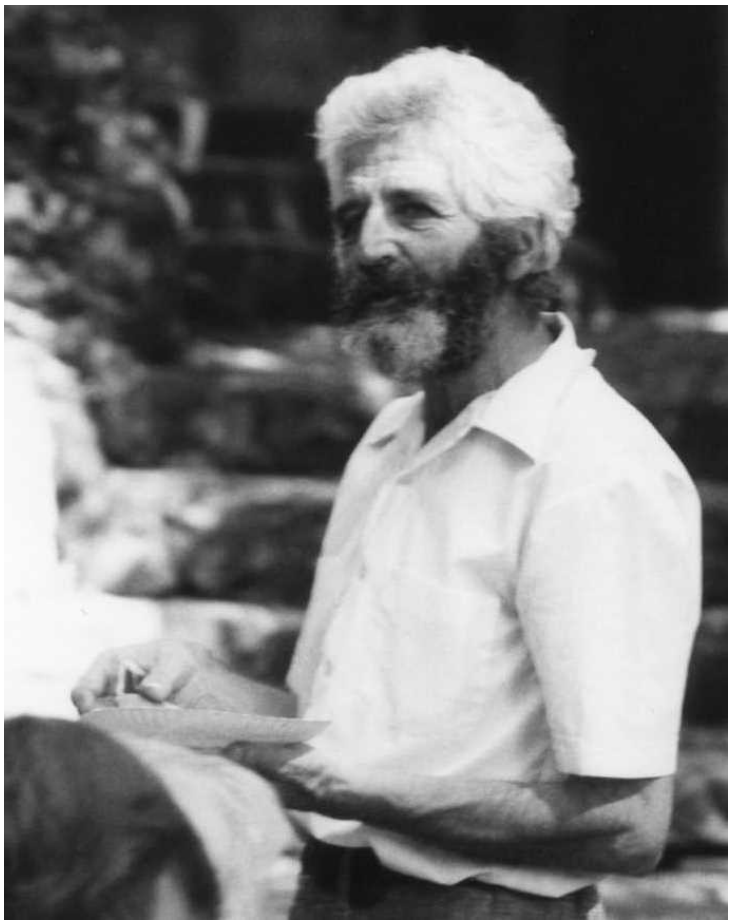
So we refine the abstraction to include these vars

$\hat{I} \doteq x = 0 \wedge y = 0 \wedge z = 0$
$\hat{R} \doteq x' = x + y \wedge y' = y + ite(z \geq 0, z, -z) \wedge z' = z + 1$

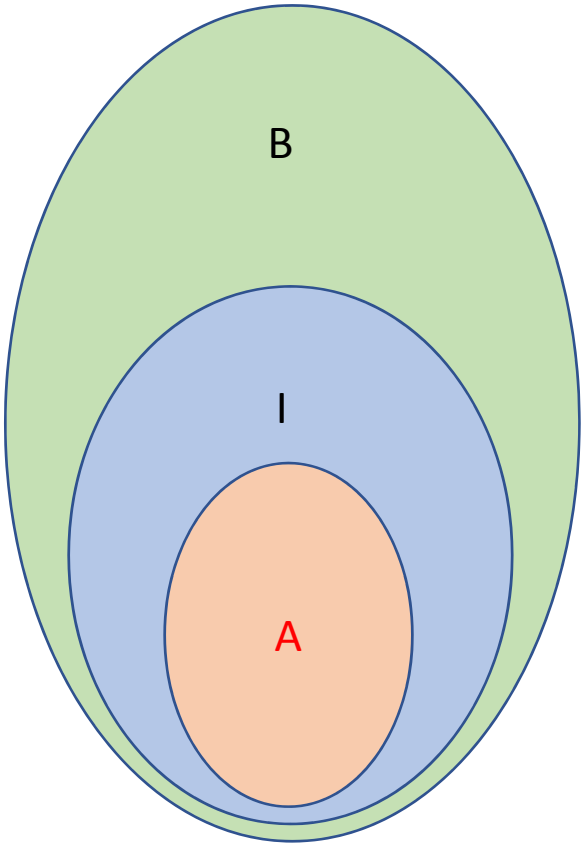
Complete model checking procedure proves ϕ on $\widehat{M}$

Unbounded Model Checking Using Craig Interpolants

William Craig (1918-2016)



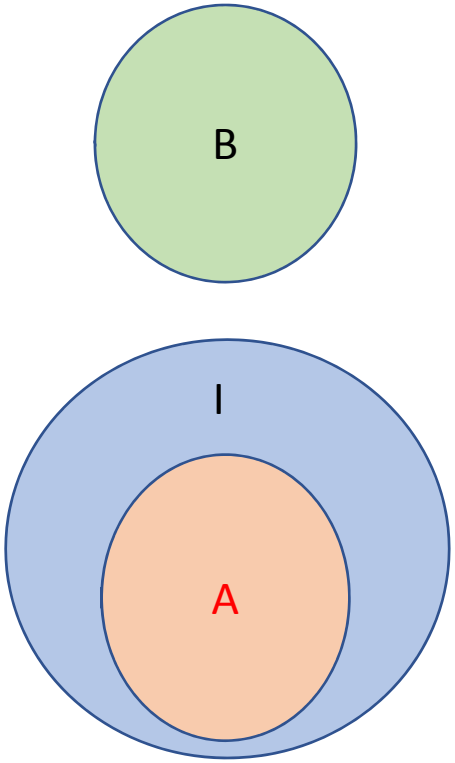
Craig Interpolants



For two formulas A and B such that $A \Rightarrow B$, a Craig interpolant is a formula I such that:

1. $A \Rightarrow I$
2. $I \Rightarrow B$
3. *The nonlogical symbols in* I occur in both A, B

Reverse Interpolants

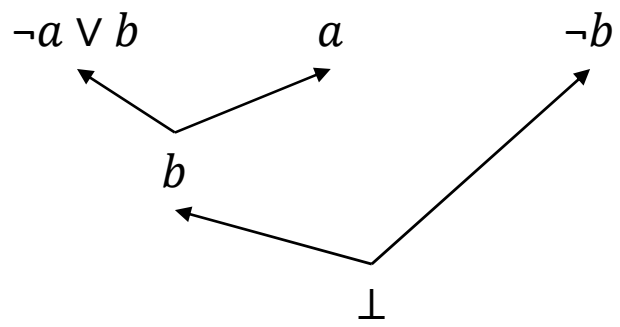


For two formulas A and B such that $A \wedge B$ is UNSAT, a reverse interpolant is a formula I such that:

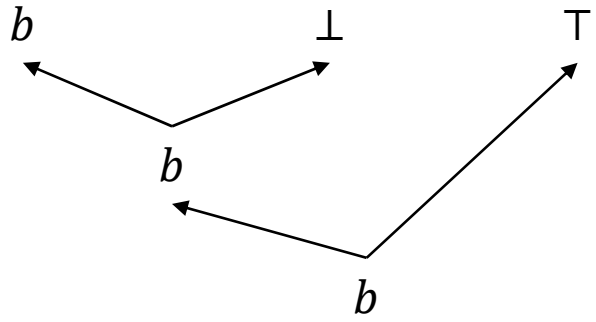
1. $A \Rightarrow I$
2. $I \wedge B$ is UNSAT
3. *The nonlogical symbols in* I occur in both A, B

Computing Interpolants from Resolution Proofs

Resolution proof for: $(\neg a \vee b), a, \neg b$



Let $\mathrm{A} = \{(\neg a \vee b), a\}$ $\mathrm{B} = \{\neg b\}$



Given a proof of UNSAT Π of $A \cup B$, for all vertices $c \in V_\Pi$ let $p(c)$ be:

- If c is a root then:
  - If $c \in A$ then $p(c) = g(c)$
  - Else $p(c) = \top$
- Else, let c_1, c_2 be preds of c and let v be their pivot variable
  - If v is local to A, then $p(c) = p(c_1) \vee p(c_2)$
  - Else, $p(c) = p(c_1) \wedge p(c_2)$

Interpolant is $p(\bot)$

Interpolation based MC

$k > 0; M = (Init, R, bad);\ P = Init$

If $Init \wedge bad$ is SAT then return UNSAFE

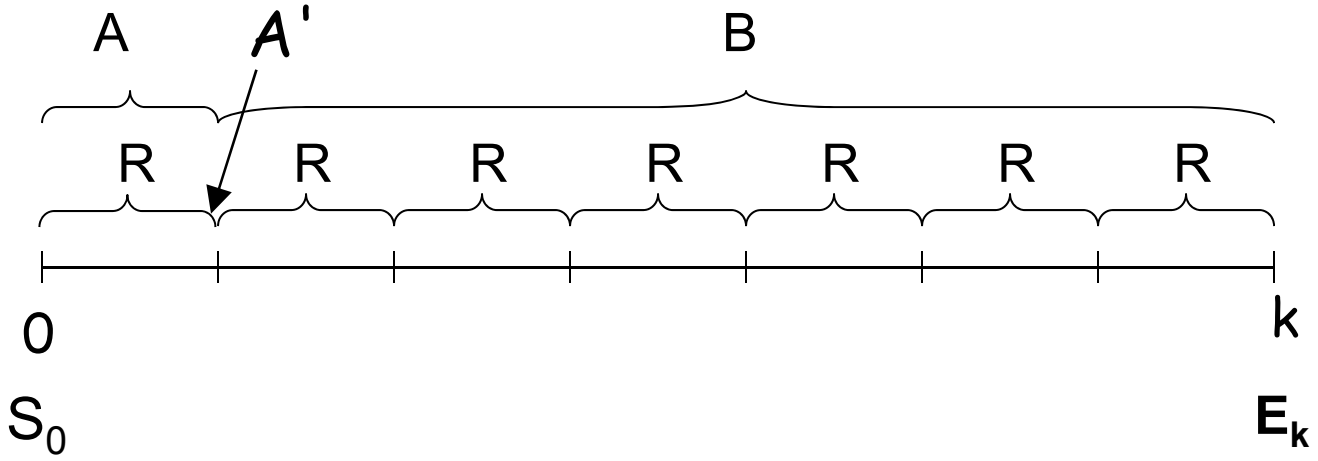
repeat

- $M' = (P, R, bad)$
- $A = PREF_1(M')$
- $B = SUFF_0^k(M')$
- If $A \wedge B$ is SAT:
  - If $P = Init$ return CEX
  - Else increase k; $P = Init$; continue
- Let Q be an interpolant for $A \wedge B$
- If $Q \Rightarrow P$ then **return** SAFE
- $P = P \vee Q$

$$PREF_h(M) = Init(s_{-l}) \wedge \bigwedge_{-h \le i < 0} R(s_i, s_{i+1})$$

$$SUFF_j^k(M) = \bigwedge_{0 \le i < k} R(s_i, s_{i+1}) \wedge \bigwedge_{j \le i \le k} bad(s_i)$$

Intuition



- A' tells us everything the prover deduced about the image of S_0 in proving it can't reach an error in k steps.
- Hence, A' is in some sense an abstraction of the image relative to the property *and* the bound k

The fixed point P is an inductive invariant

Refinement

- The procedure may be inconclusive for a fixed k
  - May add a state that reaches error in k steps (getting SAT in step 2 with Z != S_0)
- Refinement is just increasing k
  - How big can k get?

Interpolation based MC

For a fixed k:

1. Set Z initially to S_0
2. Do BMC starting from Z for k steps
   - If SAT: have we found a counterexample?
   - If UNSAT, continue
3. Use interpolation to compute over-approximation of next states of Z and add them back into Z
   – Can newly added states lead to error states in k-1 steps? In k steps?
4. If Z does not increase
   – We've reached a fixed point Z=P. Is the property true?
5. Otherwise, back to step 2