Lecture 18

Monday, March 30, 2020 4:53 PM

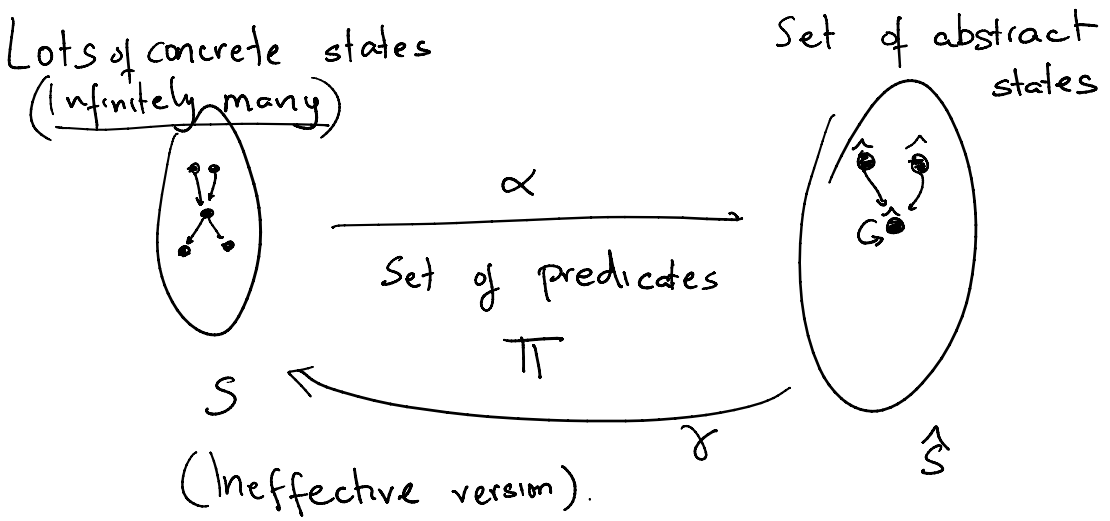
- Link to shared Google Doc:

https://docs.google.com/document/d/1bxvGf1F33HZPLlqkSmu5vh5NadFLaJRBypzr43dOLQY/edit?usp=sharing

- Hello!

- Continue discussion of predicate abstraction, and of CEGAR.
- (If time permits) Discuss dataflow analysis.

- Predicate abstraction



(Ineffective version).

Edge drawing rule: ∀ q $q' \in S$, if $q \to q'$ is a concrete transition, then add the abstract transition $\alpha(q) \to \alpha(q')$.

```
for q in S:
    for q' in S:
```

for q' in S:

If exists Conc.Transition (q, q'):

add AbsTransition $(\alpha(q)\ \alpha(q'))$

Soundness theorem: If all reachable abstract states are safe, then all reachable concrete states are also safe.

Minimal existential abstraction

Edge drawing rule ②

For all $\hat{q}$ $\hat{q}'$ $\in \hat{S}$

- if $\exists$ concrete transition $q \rightarrow q'$
  s.t $q \in \gamma(\hat{q})$ — $\alpha(q) = \hat{q}$
  $q' \in \gamma(\hat{q}')$ — $\alpha(q') = \hat{q}'$

(declare-var q —)
(declare-var q'—)
(assert (= α(q) q̂))
(assert (= α(q') q̂'))
(assert (exists Trans q q'))

- then add the abstract transition $\hat{q} \rightarrow \hat{q}'$.

- Algorithmically effective, given an SMT solver.
- For n predicates, requires 2^{2n} queries to the SMT solver.

Minimal existential abstraction (EDR ②)

- Draws exactly the transitions that the

- Draws exactly the ... abstraction is forced to draw.

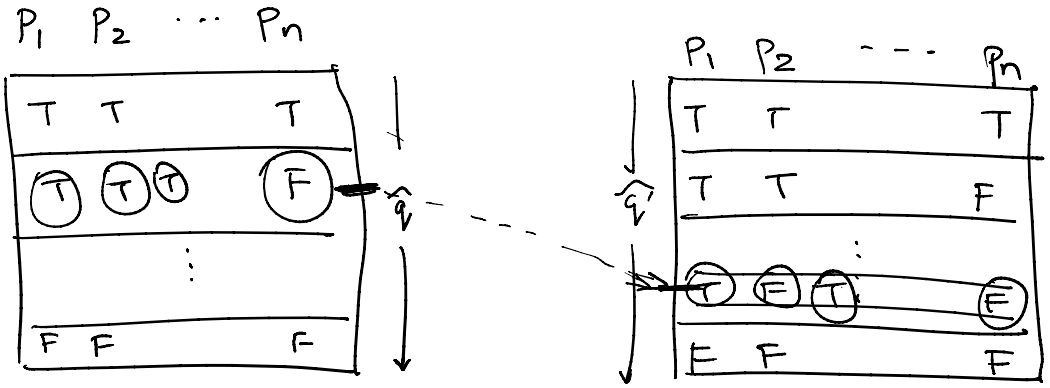
- If you skip drawing any abstract transition then the soundness theorem will fail.

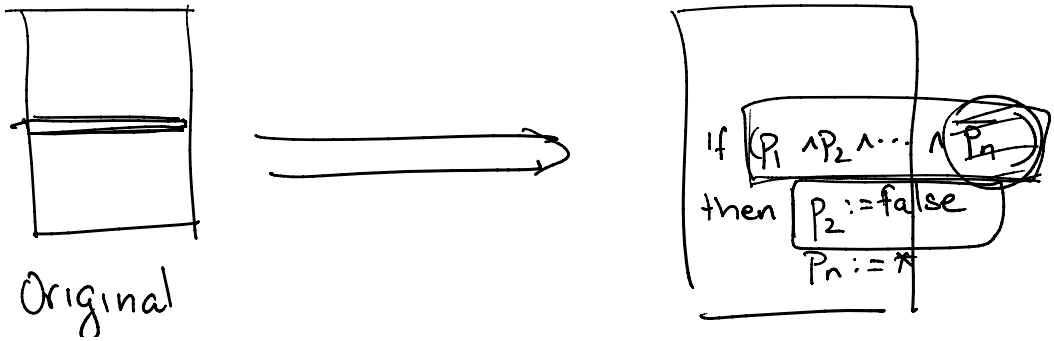
Proposal ①: Apply EDR② only to reachable abstract states. ⟸ Should always be there, unless somehow difficult to implement.

Proposal ②: Draw more edges than EDR② requires.

- Soundness theorem will continue to hold.
- Skip some queries to the SMT solver.

Existential abstractions





Original program

| Pn := * |

Boolean approx.

Challenge of existential abstraction

Path in the abstract transition graph (perhaps to an error state)

need not correspond to a path in the concrete transition graph

① – Individual transitions may be spurious

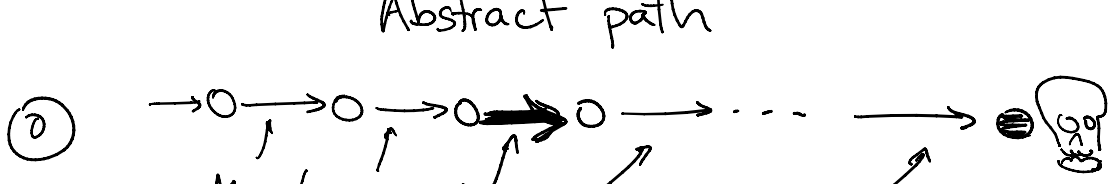
(Abstract transition graph contains more edges than EDR ②)

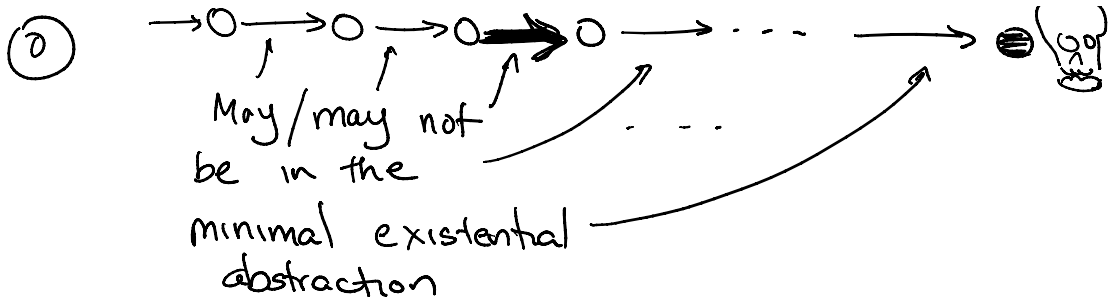
Lazy abstraction

BLAST

② – Paths in the abstract transition graph may be infeasible.

(Incompleteness of predicate abstraction as a verification technique; CEGAR)





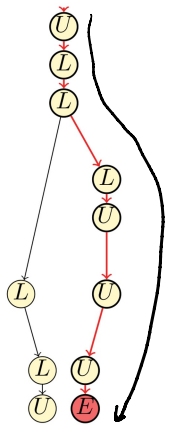
① Confirm that each transition is required by EDR②.

If any transition is spurious:

fix the Boolean program to eliminate extra edges; Go back to the beginning

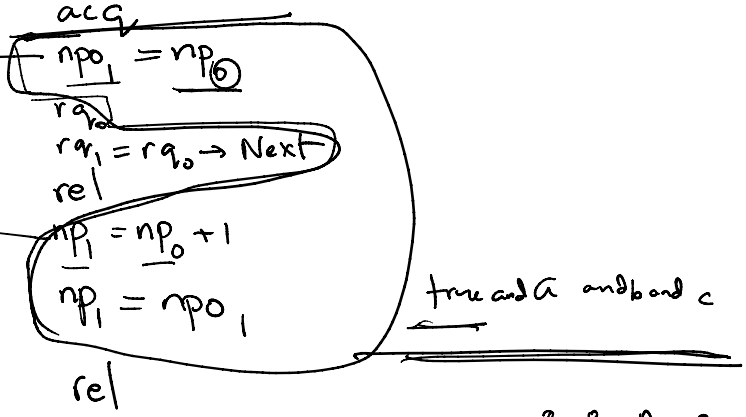
② Every transition in the abstract path was required by EDR②.

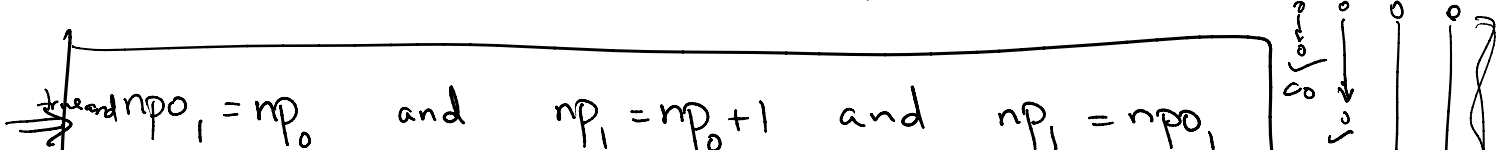
Still, the whole path might be infeasible.

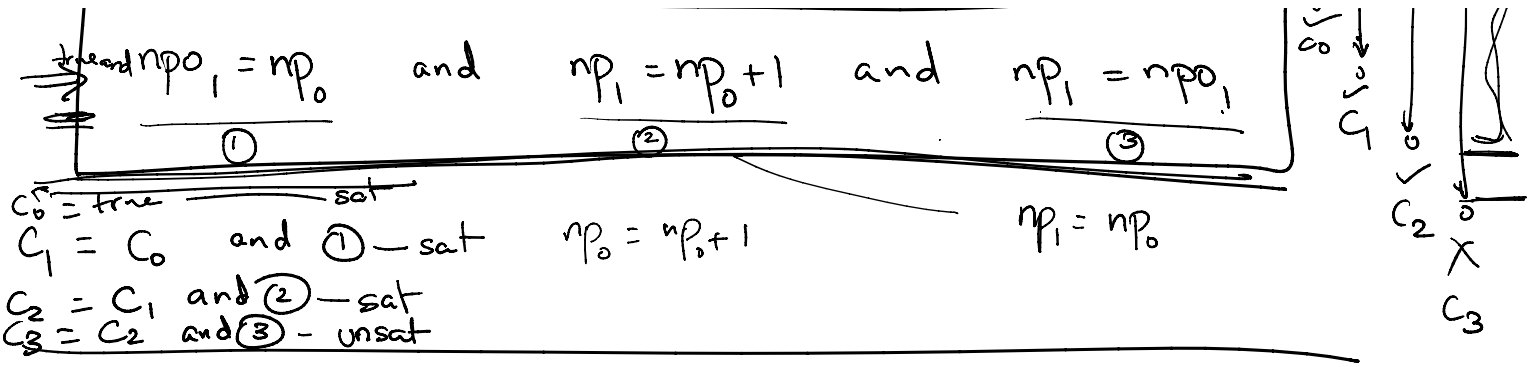


```
do {
    KeAcquireSpinLock();
    nPacketsOld = nPackets;
    if (request) {
        request = request->Next;
        KeReleaseSpinLock();
        nPackets++;
    }
} while(nPackets != nPacketsOld);

KeReleaseSpinLock();
```







Goal: Refine the abstraction by introducing new predicates.

Question: How?

Something failed between C_2 & C_3.

Question: Why is C_3 infeasible given that C_2 is feasible?

$$C_3 = C_2 \wedge B$$

Craig's Interpolation Theorem

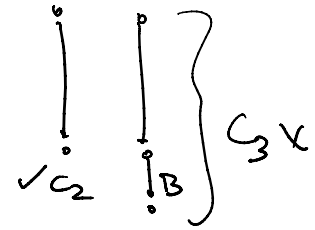
If C_3 is unsat and C_2 is sat

(& C_2 & C_3 are both formulas in FOL)

then $\exists$ a logical formula I (for interpolant)

s.t. $C_2 \Rightarrow I$

$I \wedge B$ is unsat



and where symbols(I) $\subseteq$ symbols(C_2) $\cap$ symbols(B)

Interpolating procedures exist for various theories

- Propositional logic
- Linear real arithmetic
- Difference logic with UF.

I is the new predicate

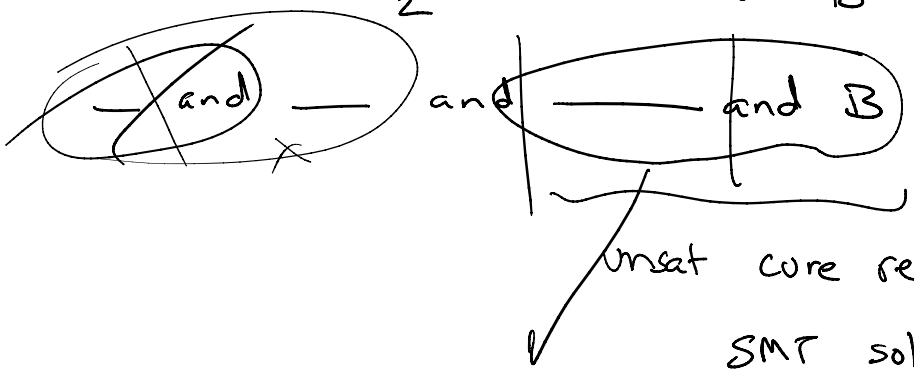
$\Pi = \{P_1\ P_2\ \ldots\ P_n\}$

$\Downarrow$

$\Pi' = \Pi \cup \{I\}$

Ex: C_2 = — and (— and —) (sat)

C_3 = (C_2) and B (unsat)



unsat core returned by SMT solver

new predicate / interpolant I.

Quantifier elimination for LRA

$(\exists z \quad x+y \le 2z \quad \text{and} \quad z \le y-3 \text{ and } x \le y)\, \varphi(x,y)$

$\frac{x+y}{2} \le z$ and $z \le y-3$ and $x \le y$

$x \le 2y-6$

$\frac{x+y}{2} \le y-3$ and $x \le y$

$x+y \le 2y-6$

$x \le y-6$ and $x \le y$

Quantifier elim for LIA

$\varphi(x) = \exists z \cdot x = z+z$

$= x$ is even.

$= (x \bmod 2 = 0)$

$\exists z \cdot x = z+z+z$

$(x \bmod 3 = 0)$