Symbolic Exploration

Azadeh Farzan
CS410 - Fall 2019
Reachability

One of the *simplest* verification problems:

- Given a *set of bad states*, can a program/system *reach* one of these states during its execution?
- It is a *decision problem*.
Propose a trivial algorithm for reachability...
State Space Exploration

A program with 100 Boolean variables can have up to $2^{100}$ different reachable states.

Representing these individually is not feasible.

Symbolic representation accommodates representing sets of states more compactly.

\[ \begin{align*}
\text{p, q, r:} & \quad \begin{array}{c}
\text{p} \\
\hline
\text{q states}
\end{array} & \quad \begin{array}{c}
q \land \lnot r \\
\hline
\text{2 states}
\end{array}
\end{align*} \]
Formally ... 

A boolean formula $F$ represents the set of all states $s$ where $s \models F$

$\text{states}(F) = \{ s \mid s \models F \}$
How do we solve reachability symbolically?
First Attempt

\[ R_0 = I \quad \rightarrow \quad \text{set of initial states} \]

\[ R_1 (\vec{v}) = [R_0 (\vec{v'}) \land \text{step}(\vec{v'}, \vec{v})] \lor R_0 (\vec{v}) \]

reachable by one step from \( R_0 \)

union \( R_0 \)

\[ R_2 (\vec{v}) = [R_1 (\vec{v'}) \land \text{step}(\vec{v'}, \vec{v})] \lor R_1 (\vec{v}) \]

\[ \vdots \]
Let $E(\mathcal{U})$ represent all error states.

At each step $j$, if $R_j(\mathcal{V}) \land E(\mathcal{U})$ is satisfiable, then an error state is reachable.
If the system is finite-state then

\[ \exists j : R_j = R_{j+1} \]

We either find a \( j \) s.t. \( R_j \notin \mathcal{E} \) is satisfiable or a \( j \) s.t. \( R_j = R_{j+1} \).
Let's make this better!
Property-Directed Reachability
A Slight Perspective Shift
Setup

- **Clause**: disjunction of literals
- **Cube**: conjunction of literals
- Each frame $R_j$ is a **CNF** formula.
- But now, it is an over-approximation of the set of reachable states in $j$ steps.
Invariants

- \( R_0 = I \)
- \( R_j \subseteq R_{j+1} \)
- \( \text{CL}(R_{j+1}) \subseteq \text{CL}(R_j) \) \( (j > 0) \)
- \( T(R_j) \subseteq R_{j+1} \) \( \implies T: \text{step} \)
- \( R_j \subseteq \neg E \) \( \implies \text{except the last frame } N \)
set of states represented by $c$

$\text{CAR}_{\text{R\text{N\text{A}\text{T}}}}$

not SAT

every state visited in a step after $R_n$
$\text{RNAT}$

$\overline{7C_2AR\text{RNAT}}$
$\text{not SAT}$

$\overline{7C_3\text{RNAT}}$
$\text{not SAT}$
$c \land c_2 \land c_3 \rightarrow $ the CNF formula for $\text{RnAT}$

guaranteed to include $(\text{RnAT})_v\text{Rn}$
The Algorithm

Check if $R_n \land E$ is SAT.

- No? $R_n \land \neg E$
  - new empty frame $R_{n+1}$
  - $\forall j > 0$, push clauses from $R_j$ to $R_{j+1}$
    - clause $c \in CL(R_j)$ can be pushed if $R_j \land \neg \neg c$ is not SAT.
  - Terminate if two equal frames found
• Is $s$ truly reachable?
• Can it be reached from $R_{N-1}$?
\[ R_{N-1} \rightarrow \text{AT AS S is SAT?} \]

• S is a cube \( \Rightarrow \) S is a clause
$R_{n-1}$ has to be made more precise!

$R_{n-1} \land T \land \exists s$ is SAT?

- $s$ is a satisfiable assignment.
- Try blocking $s$ from $R_{n-1}$!
The Algorithm

- **Check if** $R_n \land E$ **is SAT.**

  - **Yes?** → careful: $R_n$ was overshooting!
    - There is a satisfying assignment $S$.
  - **Check if** $R_{n-1} \land \neg \neg S$ **is SAT**
    - **No?** Add $S$ to $R_n$ and start over
    - **Yes?** get assignment $t$
      repeat step (**) with $(R_{n-2}, t)$
If we keep trying to block in earlier frames and reach the first frame, then the error is truly reachable.