Program Verification: Lecture 24

José Meseguer
University of Illinois at Urbana-Champaign
Symbolic Model Checking Modulo an FVP Theory $E \cup B$

- In Lecture 23, narrowing-based symbolic model checking was extended from topmost rewrite theories $\mathcal{R} = (\Sigma, B, R)$ to topmost theories $\mathcal{R} = (\Sigma, E \cup B, R)$, with $E \cup B$ FVP.

- The extension was very smooth:
  - Instead or narrowing with $R$ modulo axioms $B$ by performing $B$-unification, one narrows with $R$ modulo axioms $E \cup B$ by performing $E \cup B$-variant unification.
  - To try to make the narrowing symbolic search space finite, instead of folding symbolic states that are instances modulo axioms $B$ of more general states, we fold them into more general states symbolic states of which they are instances modulo $E \cup B$.
  - In both cases, the `fvu-narrow` command in Maude supports symbolic model checking with narrowing.

In this lecture I will: (1) illustrate this kind of symbolic reachability analysis with folding modulo an FVP theory $E \cup B$ with two infinite-state system examples, and (2) will show how the folding narrowing graph $FG_{\mathcal{R}}(u)$ from a symbolic initial state $u$ faithfully characterizes the satisfaction (resp. violation) of invariants in $\mathcal{R}$. 
The following vending machine allows buying cakes or cookies with either dollars or quarters thanks to the FVP equation: $q q q q = \$. 

mod VENDING-MACHINE is
  sorts Coin Item Marking Money State .
  subsort Coin < Money .
  op empty : -> Money .
  subsort Money Item < Marking .
  op <> : Marking -> State .
  ops $ q : -> Coin .
  ops cookie cake : -> Item .
  var M : Marking .
  rl [add-q] : < M > => < M q > .
  rl [buy-ca] : < M $ > => < M cake > .
  rl [buy-co] : < M $ > => < M cookie q > .
  eq [change]: q q q q = $ [variant] .
endm
The following vending machine allows buying cakes or cookies with either dollars or quarters thanks to the FVP equation: $q \cdot q \cdot q \cdot q = \$$. 

mod VENDING-MACHINE is 
  sorts Coin Item Marking Money State .
  subsort Coin < Money .
  op empty : -> Money .
  subsort Money Item < Marking .
  op < > : Marking -> State .
  ops $ q : -> Coin .
  ops cookie cake : -> Item .
  var M : Marking .
  rl [add-q] : < M > => < M q > .
  rl [buy-ca] : < M $ > => < M cake > .
  rl [buy-co] : < M $ > => < M cookie q > .
  eq [change]: q q q q = $ [variant] .

endm

\[
< $ $ > \rightarrow < $ $ $ > \rightarrow < $ $ $ $ > \rightarrow \infty \\
< $ $ q > \rightarrow < $ $ q q > \rightarrow \infty \\
(\text{one initial state - infinite space})
Narrowing-Based Symbolic Model Checking

- We can consider, for example, the most general symbolic initial state possible in VENDING-MACHINE, namely, \(< M >\) and its symbolic transitions by the [buy-ca] rule.
- The vertical lines in the figure below describe the narrowing steps and unifiers for the narrowing path:

\[
\langle M \rangle \leadsto \langle \text{cake } M' \rangle \leadsto \langle \text{cake cake } M'' \rangle \leadsto \ldots
\]

\[
\begin{align*}
< & M > \\
\{M \rightarrow$ & $M'\} \\
< & \text{cake } M' > \\
\{M' \rightarrow$ & $M''\} \\
< & \text{cake cake } M'' > \\
\infty
\end{align*}
\]

(Infinite Symbolic Search Space)
Narrowing-Based Symbolic Model Checking

- We can consider, for example, the most general symbolic initial state possible in VENDING-MACHINE, namely, $< M >$ and its symbolic transitions by the [buy-ca] rule.
- The vertical lines in the figure below describe the narrowing steps and unifiers for the narrowing path:

$$
< M > \leadsto < \text{cake } M' > \leadsto < \text{cake cake } M'' > \leadsto \ldots
$$

(Infinite Symbolic Search Space)
Folding Infinite Symbolic State Spaces into a Finite Graph

- **Narrowing-Based Symbolic Model Checking** can model check **infinite state systems** by representing **infinite sets of states** by terms with variables.

- The symbolic state space (narrowing tree) **can still be infinite**; but its states can be **over-approximated** by folding in the folding graph, which sometimes can be **finite**.

- The transition system of the **folding graph** is an **abstraction** (it identifies many symbolic states) that **over-approximates** the states and transitions of the narrowing tree.
Folding Infinite Symbolic State Spaces into a Finite Graph

- Narrowing-Based Symbolic Model Checking can model check infinite state systems by representing infinite sets of states by terms with variables.
- The symbolic state space (narrowing tree) can still be infinite; but its states can be over-approximated by folding in the folding graph, which sometimes can be finite.
- The transition system of the folding graph is an abstraction (it identifies many symbolic states) that over-approximates the states and transitions of the narrowing tree.

Narrowing + folding relation \(\Rightarrow\) (symbolic initial states and (hopefully) finite state space) (instantiation relation \(\preceq_{E\cup B}\))
E ∪ B-Unification Command in Maude

Maude provides a \((E ∪ B)\)-unification command for any equational theory \((\Sigma, E ∪ B)\) that is convergent modulo \(B\). The complete set of \((E ∪ B)\)-unifiers will always be finite if \(E ∪ B\) is FVP.

\[
\text{variant unify [ in } \langle \text{ModId} \rangle : ] \langle \text{Term1} \rangle =? \langle \text{Term2} \rangle .
\]

- \text{ModId} is the name of the module
- A complete set of \(E ∪ B\)-unifiers are returned.
- Folding variant narrowing is used internally to compute \(E ∪ B\)-unifiers.
(E \cup B)-Unification Command in Maude (II)

Maude> (variant unify in NARROWING-VENDING-MACHINE :  
    < q q X:Marking > =? < $ Y:Marking > .)  
Solution 1  
X:Marking --> q q Y:Marking  
Solution 2  
X:Marking --> $ #12:Marking ; Y:Marking --> q q #12:Marking
Bakery Algorithm: Transition System

Token to give ; Token serving ; Set of Processes
Nat       Nat           [{ idle, wait(Nat), crit(Nat) }]

\[
\begin{align*}
        r | \ N ; M ; [idle]\ PS & \Rightarrow (s\ N) ; M ; [wait(N)]\ PS . \\
r | \ N ; M ; [wait(M)]\ PS & \Rightarrow N ; M ; [crit(M)]\ PS . \\
r | \ N ; M ; [crit(M)]\ PS & \Rightarrow N ; (s\ M) ; [idle]\ PS . 
\end{align*}
\]
Bakery Algorithm: Transition System

Token to give ; Token serving ; Set of Processes
Nat Nat \{ idle, wait(Nat), crit(Nat) \]

\[\begin{align*}
    \text{rl } \text{N} ; \text{M} ; \text{[idle]} \text{ PS } & \Rightarrow (\text{s N}) ; \text{M} ; \text{[wait}}(\text{N})]\text{ PS} . \\
    \text{rl } \text{N} ; \text{M} ; \text{[wait(M)] PS } & \Rightarrow \text{N} ; \text{M} ; \text{[crit(M)] PS} . \\
    \text{rl } \text{N} ; \text{M} ; \text{[crit(M)] PS } & \Rightarrow \text{N} ; (\text{s M}) ; \text{[idle]} \text{ PS} .
\end{align*}\]

\[\begin{align*}
    0 ; 0 ; \text{[idle]} & \rightarrow s ; s ; \text{[idle]} \rightarrow ss ; ss ; \text{[idle]} \\
    s ; 0 ; \text{[wait(0)]} & \rightarrow s ; s ; \text{[wait(s)]} \rightarrow sss ; ss ; \text{[wait(ss)]} \\
    s ; 0 ; \text{[crit(s)]} & \rightarrow ss ; s ; \text{[crit(s)]} \rightarrow \infty
\end{align*}\]

(Transition System: one initial state - infinite space)
Bakery Algorithm: Symbolic Transition System

\[
\begin{align*}
0 & ; 0 & ; \text{[idle]} & \quad s & ; s & ; \text{[idle]} & \quad ss & ; ss & ; \text{[idle]} \\
\downarrow & & \downarrow & & \downarrow \\
\text{s} & ; 0 & ; \text{[wait(0)]} & \quad \text{s} & ; s & ; \text{[wait(s)]} & \quad \text{sss} & ; ss & ; \text{[wait(ss)]} \\
\downarrow & & \downarrow & & \downarrow \\
\text{s} & ; 0 & ; \text{[crit(s)]} & \quad ss & ; s & ; \text{[crit(s)]} & \quad \infty
\end{align*}
\]

(Transition System: one initial state - infinite state space)
Bakery Algorithm: Symbolic Transition System

\[
\begin{align*}
0 ; 0 ; \text{[idle]} & \rightarrow s ; s ; \text{[idle]} & \rightarrow ss ; ss ; \text{[idle]} \\
\downarrow & & \downarrow & & \downarrow \\
s ; 0 ; \text{[wait(0)]} & \rightarrow s ; s ; \text{[wait(s)]} & \rightarrow sss ; ss ; \text{[wait(ss)]} \\
\downarrow & & \downarrow & & \downarrow \\
s ; 0 ; \text{[crit(s)]} & \rightarrow ss ; s ; \text{[crit(s)]} & \rightarrow \text{[crit(s)]} \\
\end{align*}
\]

(Transition System: one initial state - infinite state space)

\[
\begin{align*}
N ; N ; \text{[idle]} & \rightarrow sN ; sN ; \text{[idle]} & \rightarrow ssN ; ssN ; \text{[idle]} \\
\downarrow & & \downarrow & & \downarrow \\
sN ; N ; \text{[wait(N)]} & \rightarrow ssN ; sN ; \text{[wait(sN)]} & \rightarrow sssN ; ssN ; \text{[wait(ssN)]} \\
\downarrow & & \downarrow & & \downarrow \\
sN ; N ; \text{[crit(N)]} & \rightarrow ssN ; sN ; \text{[crit(sN)]} & \rightarrow \text{[crit(sN)]} \\
\end{align*}
\]

(Symbolic Transition System: infinite initial state set - infinite state space)
Bakery Algorithm: Folding the Symbolic Transition System

(Folding Symbolic Transition System: infinite initial state set - finite state space)
The Faithfulness of Folding Symbolic Transition Systems

Suppose that we wish to verify an invariant $I$ for a topmost $\mathcal{R}$ using the folding graph $FG_{\mathcal{R}}(u)$ generated by a symbolic initial state $u$. Since $FG_{\mathcal{R}}(u)$ over-approximates the narrowing tree from $u$, if no violation of invariant $I$ (i.e., an instance of $u$ reaching its complement) is found exploring $FG_{\mathcal{R}}(u)$, a fortiori no such violation can be found in the narrowing tree. But by the Completeness of Narrowing Search Theorem (Lecture 23, pg. 8), this means that $I$ holds for all ground instances of $u$.

But what happens if we find a counterexample, that is, a path from $u$ in $FG_{\mathcal{R}}(u)$ violating $I$? Does it mean that invariant $I$ is violated for some ground instance of $u$? Or could such a path be a spurious counterexample not corresponding to any real violation of $I$?

We shall call $FG_{\mathcal{R}}(u)$ a faithful abstraction of $\mathcal{R}$ from the set of initial states symbolically specified by $u$ iff $FG_{\mathcal{R}}(u)$ has no spurious counterexamples for any pattern-specified invariant $I$. To show that $FG_{\mathcal{R}}(u)$ is faithful, we need to look at it more carefully.
The Folding Narrowing Graph $\text{FNG}_R(u)$

Given a topmost $R = (\Sigma, E \cup B, R)$ with $E \cup B$ FVP, and a symbolic initial state $u$, the folding narrowing graph $\text{FNG}_R(u)$ is generated in a breadth first manner by paths of increasing length from $u$ as follows:

- $u \xrightarrow{R,(E \cup B)} u$ is the only path at depth 0.
- The paths of length $n + 1$ (and the depth of their ending nodes) are:
  - either narrowing paths $u \xrightarrow{n} R,(E \cup B) v_n \xrightarrow{n} R,(E \cup B) v$ such that (i) $u \xrightarrow{n} R,(E \cup B) v_n$ in $\text{FNG}_R(u)$, (so $v$ has narrowing depth $n+1$ in $u$’s narrowing tree), and it is not the case that (ii): either exists a narrowing path $u \xrightarrow{k} R,(E \cup B) w$, $k \leq n$, in $\text{FNG}_R(u)$, or a different narrowing path $u \xrightarrow{n} R,(E \cup B) w_n \xrightarrow{n} R,(E \cup B) w$ with $u \xrightarrow{n} R,(E \cup B) w_n$ in $\text{FNG}_R(u)$, such that $v \leq_{E \cup B} w$ (read, $v$ is an instance modulo $E \cup B$ of $w$), where,

$$v \leq_{E \cup B} w \iff \exists \gamma \text{ s.t. } w\gamma = E \cup B v;$$

- otherwise, they are paths of the form $u \xrightarrow{n} R,(E \cup B) v_n \leq_{E \cup B} w$ associated to a narrowing path $u \xrightarrow{n} R,(E \cup B) v_n \xrightarrow{n} R,(E \cup B) v$ s.t. (i)–(ii) above hold with $v \leq_{E \cup B} w$. Therefore, $w$ has narrowing depth $d \leq n + 1$ in $u$’s narrowing tree.
Faithfulness of $FNG_\mathcal{R}(u)$ (proofs in Appendix)

**Theorem**

*(Over-Approximation Theorem).* Given a topmost $\mathcal{R} = (\Sigma, E \cup B, R)$ with $E \cup B$ FVP and a symbolic initial state $u$, for every narrowing path from $u$, $u \rightsquigarrow^*_{\mathcal{R},(E\cup B)} v$ there is a node $w$ in the folding narrowing path of $u$ $FNG_\mathcal{R}(u)$ such that $v \preceq_{E\cup B} w$.

**Theorem**

*(Faithfulness Theorem).* For $\mathcal{R} = (\Sigma, E \cup B, R)$ and $u$ as above, $FNG_\mathcal{R}(u)$ is a faithful over-approximation of the narrowing tree of $u$ in the sense that for any set of states of $\mathcal{R}$ described by a pattern term $p$, an instance of $p$ can be reached by a narrowing path $u \rightsquigarrow^*_{\mathcal{R},(E\cup B)} v$ such that $\text{Unif}_{E\cup B}(v = p) \neq \emptyset$ iff there is a node $w$ in $FNG_\mathcal{R}(u)$ such that $\text{Unif}_{E\cup B}(w = p) \neq \emptyset$.

In particular, if $p$ is the negation of an invariant, any counterexample found in $FNG_\mathcal{R}(u)$ is a true counterexample and therefore proves the invariant’s violation (i.e., $FNG_\mathcal{R}(u)$ has no spurious counterexamples).