Program Verification: Lecture 26

José Meseguer
University of Illinois at Urbana-Champaign

Narrowing-Based Symbolic LTL Model Checking

We can verify invariants of a topmost rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ when $E \cup B$ is FVP by narrowing search with $\leadsto_{R/(E \cup B)}$ from a symbolic initial state u.

Narrowing-Based Symbolic LTL Model Checking

We can verify invariants of a topmost rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ when $E \cup B$ is FVP by narrowing search with $\leadsto_{R/(E \cup B)}$ from a symbolic initial state u. Can this be generalized to narrowing-based symbolic LTL model checking for such an $\mathcal{R}$?

Narrowing-Based Symbolic LTL Model Checking

We can verify invariants of a topmost rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ when $E \cup B$ is FVP by narrowing search with $\rightsquigarrow_{R/(E \cup B)}$ from a symbolic initial state u. Can this be generalized to narrowing-based symbolic LTL model checking for such an $\mathcal{R}$?

The main problem is that, in general, it is meaningless to say which state predicates $p \in \Pi$ are satisfied in a symbolic state u, since some ground instance $u\rho$ may satisfy some predicates in Π, and another ground instance $u\tau$ may satisfy a different set of predicates in Π.

Narrowing-Based Symbolic LTL Model Checking

We can verify invariants of a topmost rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ when $E \cup B$ is FVP by narrowing search with $\rightsquigarrow_{R/(E \cup B)}$ from a symbolic initial state u. Can this be generalized to narrowing-based symbolic LTL model checking for such an $\mathcal{R}$?

The main problem is that, in general, it is meaningless to say which state predicates $p \in \Pi$ are satisfied in a symbolic state u, since some ground instance $u\rho$ may satisfy some predicates in Π, and another ground instance $u\tau$ may satisfy a different set of predicates in Π.

However, if $\mathcal{R}$ is deadlock-free, and the equations D defining the satisfaction relation $u \models p$ between terms of top sort $State$ and state predicates Π are such that $E \cup D \cup B$ is FVP modulo B, LTL symbolic model checking of $\mathcal{R}$ from a symbolic initial state u becomes possible in a symbolic Kripke structure $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$, whose symbolic transitions are performed by a Π-aware narrowing relation $\rightsquigarrow_\Pi$ explained in what follows.

The Narrowing Relation $\rightsquigarrow_{\Pi}$

Given a deadlock-free topmost rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ with rules $(l \to r) \in R$ s.t. $l, r \in T_{\Sigma}(X) \setminus X$, topmost sort $State$, and a set $\Pi = \{p_1, \ldots, p_k\}$ of state predicates whose satisfaction in $\mathcal{R}$ is defined by equations D such that $E \cup D \cup B$ is FVP modulo axioms B, the Π-aware narrowing relation between terms $u, w \in T_{\Sigma,State}(X)$ is defined as follows:

The Narrowing Relation $\rightsquigarrow_\Pi$

Given a deadlock-free topmost rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ with rules $(l \to r) \in R$ s.t. $l, r \in T_\Sigma(X) \setminus X$, topmost sort $State$, and a set $\Pi = \{p_1, \ldots, p_k\}$ of state predicates whose satisfaction in $\mathcal{R}$ is defined by equations D such that $E \cup D \cup B$ is FVP modulo axioms B, the Π-aware narrowing relation between terms $u, w \in T_{\Sigma,State}(X)$ is defined as follows:

$$u \overset{\alpha\gamma}{\rightsquigarrow}_\Pi w$$

holds iff (by definition)

The Narrowing Relation $\rightsquigarrow_{\Pi}$

Given a deadlock-free topmost rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ with rules $(l \to r) \in R$ s.t. $l, r \in T_{\Sigma}(X) \setminus X$, topmost sort *State*, and a set $\Pi = \{p_1, \ldots, p_k\}$ of state predicates whose satisfaction in $\mathcal{R}$ is defined by equations D such that $E \cup D \cup B$ is FVP modulo axioms B, the Π-aware narrowing relation between terms $u, w \in T_{\Sigma,State}(X)$ is defined as follows:

$$u \overset{\alpha\gamma}{\rightsquigarrow}_{\Pi} w$$

holds iff (by definition)

- $\exists v \;\; s.t. \;\; u \rightsquigarrow^{\alpha}_{R/(E \cup B)} v$

The Narrowing Relation $\rightsquigarrow_{\Pi}$

Given a deadlock-free topmost rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ with rules $(l \to r) \in R$ s.t. $l, r \in T_{\Sigma}(X) \setminus X$, topmost sort $State$, and a set $\Pi = \{p_1, \ldots, p_k\}$ of state predicates whose satisfaction in $\mathcal{R}$ is defined by equations D such that $E \cup D \cup B$ is FVP modulo axioms B, the Π-aware narrowing relation between terms $u, w \in T_{\Sigma,State}(X)$ is defined as follows:

$$u \overset{\alpha\gamma}{\rightsquigarrow}_{\Pi} w$$

holds iff (by definition)

- $\exists v \;\; s.t. \;\; u \overset{\alpha}{\rightsquigarrow}_{R/(E \cup B)} v$
- $\exists (b_1, \ldots, b_k) \in \{true, false\}^k$

The Narrowing Relation $\rightsquigarrow_{\Pi}$

Given a deadlock-free topmost rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ with rules $(l \to r) \in R$ s.t. $l, r \in T_{\Sigma}(X) \setminus X$, topmost sort $State$, and a set $\Pi = \{p_1, \ldots, p_k\}$ of state predicates whose satisfaction in $\mathcal{R}$ is defined by equations D such that $E \cup D \cup B$ is FVP modulo axioms B, the Π-aware narrowing relation between terms $u, w \in T_{\Sigma, State}(X)$ is defined as follows:

$$u \overset{\alpha\gamma}{\rightsquigarrow}_{\Pi} w$$

holds iff (by definition)

- $\exists v \;\; s.t. \;\; u \rightsquigarrow^{\alpha}_{R/(E \cup B)} v$
- $\exists (b_1, \ldots, b_k) \in \{true, false\}^k$
- $\exists \gamma \in Unif_{E \cup D \cup B}(v \models p_1 = b_1 \wedge \ldots \wedge v \models p_k = b_k)$

The Narrowing Relation $\rightsquigarrow_{\Pi}$

Given a deadlock-free topmost rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ with rules $(l \rightarrow r) \in R$ s.t. $l, r \in T_{\Sigma}(X) \setminus X$, topmost sort $State$, and a set $\Pi = \{p_1, \ldots, p_k\}$ of state predicates whose satisfaction in $\mathcal{R}$ is defined by equations D such that $E \cup D \cup B$ is FVP modulo axioms B, the Π-aware narrowing relation between terms $u, w \in T_{\Sigma,State}(X)$ is defined as follows:

$$u \overset{\alpha\gamma}{\rightsquigarrow}_{\Pi} w$$

holds iff (by definition)

- $\exists v \;\; s.t. \;\; u \rightsquigarrow^{\alpha}_{R/(E \cup B)} v$
- $\exists (b_1, \ldots, b_k) \in \{true, false\}^k$
- $\exists \gamma \in Unif_{E \cup D \cup B}(v \models p_1 = b_1 \wedge \ldots \wedge v \models p_k = b_k)$

such that $w = v\gamma$.

The Kripke Structure $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$

For a symbolic state $u \in T_{\Sigma,State}(X)$ s.t. $\exists(b_1,\ldots,b_k) \in \{true, false\}^k$ with $(u \models p_i)!_{\vec{E\cup D},B} = b_i$, $1 \leq i \leq k$, $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$ is a Kripke structure with set of states

The Kripke Structure $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$

For a symbolic state $u \in T_{\Sigma,State}(X)$ s.t. $\exists(b_1, \ldots, b_k) \in \{true, false\}^k$ with $(u \models p_i)!_{\vec{E \cup D},B} = b_i$, $1 \leq i \leq k$, $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$ is a Kripke structure with set of states $NK(u) = \{w \in T_{\Sigma,State}(X) \mid u \rightsquigarrow^{*}_{\Pi} w\}$,

The Kripke Structure $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$

For a symbolic state $u \in T_{\Sigma,State}(X)$ s.t. $\exists(b_1,\ldots,b_k) \in \{true, false\}^k$ with $(u \models p_i)!_{\vec{E \cup D},B} = b_i$, $1 \leq i \leq k$, $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$ is a Kripke structure with set of states $NK(u) = \{w \in T_{\Sigma,State}(X) \mid u \rightsquigarrow_\Pi^* w\}$, transition relation $\rightsquigarrow_\Pi$,

The Kripke Structure $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$

For a symbolic state $u \in T_{\Sigma,State}(X)$ s.t. $\exists(b_1, \ldots, b_k) \in \{true, false\}^k$ with $(u \models p_i)!_{E \cup \vec{D},B} = b_i$, $1 \leq i \leq k$, $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$ is a Kripke structure with set of states $NK(u) = \{w \in T_{\Sigma,State}(X) \mid u \leadsto^{*}_{\Pi} w\}$, transition relation $\leadsto_{\Pi}$, and satisfaction relation $w \models_{\mathcal{NK}(\mathcal{R},State)_{\Pi}(u)} p_i$ defined for each $w \in NK(u)$ and $p_i \in \Pi$ by the unique $b'_i \in \{true, false\}^k$ such that $(w \models p_i)!_{E \cup \vec{D},B} = b'_i$, $1 \leq i \leq k$.

The Kripke Structure $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$

For a symbolic state $u \in T_{\Sigma,State}(X)$ s.t. $\exists(b_1, \ldots, b_k) \in \{true, false\}^k$ with $(u \models p_i)!_{E \cup \vec{D},B} = b_i$, $1 \leq i \leq k$, $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$ is a Kripke structure with set of states $NK(u) = \{w \in T_{\Sigma,State}(X) \mid u \rightsquigarrow^*_\Pi w\}$, transition relation $\rightsquigarrow_\Pi$, and satisfaction relation $w \models_{\mathcal{NK}(\mathcal{R},State)_\Pi(u)} p_i$ defined for each $w \in NK(u)$ and $p_i \in \Pi$ by the unique $b'_i \in \{true, false\}^k$ such that $(w \models p_i)!_{E \cup \vec{D},B} = b'_i$, $1 \leq i \leq k$.

The following theorem about $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$ (whose proof is given in the Appendix 1) shows that any LTL formula φ which holds for a symbolic initial state u also holds for all its ground instance states.

The Kripke Structure $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$

For a symbolic state $u \in T_{\Sigma,State}(X)$ s.t. $\exists(b_1, \ldots, b_k) \in \{true, false\}^k$ with $(u \models p_i)!_{\vec{E \cup D},B} = b_i$, $1 \leq i \leq k$, $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$ is a Kripke structure with set of states $NK(u) = \{w \in T_{\Sigma,State}(X) \mid u \rightsquigarrow^*_\Pi w\}$, transition relation $\rightsquigarrow_\Pi$, and satisfaction relation $w \models_{\mathcal{NK}(\mathcal{R},State)_\Pi(u)} p_i$ defined for each $w \in NK(u)$ and $p_i \in \Pi$ by the unique $b'_i \in \{true, false\}^k$ such that $(w \models p_i)!_{\vec{E \cup D},B} = b'_i$, $1 \leq i \leq k$.

The following theorem about $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$ (whose proof is given in the Appendix 1) shows that any LTL formula φ which holds for a symbolic initial state u also holds for all its ground instance states.

Theorem

For each $\varphi \in LTL(\Pi)$ and u as above, if $\mathcal{NK}(\mathcal{R}, State)_\Pi(u), u \models \varphi$, then $\forall \rho \in [vars(u) \to T_\Sigma]$, $\mathcal{K}(\mathcal{R}, State)_\Pi, [u\rho] \models \varphi$.

The Kripke Structure $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$

For a symbolic state $u \in T_{\Sigma,State}(X)$ s.t. $\exists(b_1, \ldots, b_k) \in \{true, false\}^k$ with $(u \models p_i)!_{\vec{E \cup D},B} = b_i$, $1 \le i \le k$, $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$ is a Kripke structure with set of states $NK(u) = \{w \in T_{\Sigma,State}(X) \mid u \rightsquigarrow^*_\Pi w\}$, transition relation $\rightsquigarrow_\Pi$, and satisfaction relation $w \models_{\mathcal{NK}(\mathcal{R},State)_\Pi(u)} p_i$ defined for each $w \in NK(u)$ and $p_i \in \Pi$ by the unique $b'_i \in \{true, false\}^k$ such that $(w \models p_i)!_{\vec{E \cup D},B} = b'_i$, $1 \le i \le k$.

The following theorem about $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$ (whose proof is given in the Appendix 1) shows that any LTL formula φ which holds for a symbolic initial state u also holds for all its ground instance states.

Theorem

For each $\varphi \in LTL(\Pi)$ and u as above, if $\mathcal{NK}(\mathcal{R}, State)_\Pi(u), u \models \varphi$, then $\forall \rho \in [vars(u) \to T_\Sigma]$, $\mathcal{K}(\mathcal{R}, State)_\Pi, [u\rho] \models \varphi$.

Note that we can always split any $v \in T_{\Sigma,State}(X) \setminus X$ into a finite set of instances by unifiers that satisfy Π. In this way, the assumption that the satisfaction of Π-predicates is defined in u can be weakened.

State Space Reduction in $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$

By the above Theorem, the Kripke structure $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$ supports LTL model checking for all ground instances of u using the decision procedure for LTL model checking described in Appendix 1 to Lecture 22.

State Space Reduction in $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$

By the above Theorem, the Kripke structure $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$ supports LTL model checking for all ground instances of u using the decision procedure for LTL model checking described in Appendix 1 to Lecture 22. However, this requires that the set $NK(u)$ is finite. When $NK(u)$ is infinite, we can try one of the following four possibilities to reduce the state space of $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$ to a finite state space:

State Space Reduction in $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$

By the above Theorem, the Kripke structure $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$ supports LTL model checking for all ground instances of u using the decision procedure for LTL model checking described in Appendix 1 to Lecture 22. However, this requires that the set $NK(u)$ is finite. When $NK(u)$ is infinite, we can try one of the following four possibilities to reduce the state space of $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$ to a finite state space:

1. Perform LTL model checking by folding variant narrowing, provided the folding $\leadsto_{\Pi}$-narrowing graph from u is finite.

State Space Reduction in $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$

By the above Theorem, the Kripke structure $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$ supports LTL model checking for all ground instances of u using the decision procedure for LTL model checking described in Appendix 1 to Lecture 22. However, this requires that the set $NK(u)$ is finite. When $NK(u)$ is infinite, we can try one of the following four possibilities to reduce the state space of $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$ to a finite state space:

1. Perform LTL model checking by folding variant narrowing, provided the folding $\leadsto_{\Pi}$-narrowing graph from u is finite.
2. Define an equational abstraction $\mathcal{R}/G$ such that: (i) $E \cup D \cup G \cup B$ is FVP and protects the Booleans, and (ii) the folding $\leadsto_{\Pi}$-narrowing graph from u is finite for $\mathcal{R}/G$.

State Space Reduction in $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$

By the above Theorem, the Kripke structure $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$ supports LTL model checking for all ground instances of u using the decision procedure for LTL model checking described in Appendix 1 to Lecture 22. However, this requires that the set $NK(u)$ is finite. When $NK(u)$ is infinite, we can try one of the following four possibilities to reduce the state space of $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$ to a finite state space:

1. Perform LTL model checking by folding variant narrowing, provided the folding $\rightsquigarrow_\Pi$-narrowing graph from u is finite.
2. Define an equational abstraction $\mathcal{R}/G$ such that: (i) $E \cup D \cup G \cup B$ is FVP and protects the Booleans, and (ii) the folding $\rightsquigarrow_\Pi$-narrowing graph from u is finite for $\mathcal{R}/G$.
3. Define a bisimilar equational abstraction $\mathcal{R}/G$ such that: (i) $E \cup D \cup G \cup B$ is FVP and protects the Booleans, and (ii) the folding $\rightsquigarrow_\Pi$-narrowing graph from u is finite for $\mathcal{R}/G$.

State Space Reduction in $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$

By the above Theorem, the Kripke structure $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$ supports LTL model checking for all ground instances of u using the decision procedure for LTL model checking described in Appendix 1 to Lecture 22. However, this requires that the set $NK(u)$ is finite. When $NK(u)$ is infinite, we can try one of the following four possibilities to reduce the state space of $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$ to a finite state space:

1. Perform LTL model checking by folding variant narrowing, provided the folding $\rightsquigarrow_{\Pi}$-narrowing graph from u is finite.
2. Define an equational abstraction $\mathcal{R}/G$ such that: (i) $E \cup D \cup G \cup B$ is FVP and protects the Booleans, and (ii) the folding $\rightsquigarrow_{\Pi}$-narrowing graph from u is finite for $\mathcal{R}/G$.
3. Define a bisimilar equational abstraction $\mathcal{R}/G$ such that: (i) $E \cup D \cup G \cup B$ is FVP and protects the Booleans, and (ii) the folding $\rightsquigarrow_{\Pi}$-narrowing graph from u is finite for $\mathcal{R}/G$.
4. Perform bounded LTL symbolic model checking.

Let us explore these possibilities in more detail.

The Folding $\rightsquigarrow_{\Pi}$-narrowing graph from u

Replacing $\rightsquigarrow_{R/(E \cup B)}$ by $\rightsquigarrow_{\Pi}$, $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$ is entirely similar to the narrowing tree from u.

The Folding $\rightsquigarrow_{\Pi}$-narrowing graph from u

Replacing $\rightsquigarrow_{R/(E \cup B)}$ by $\rightsquigarrow_{\Pi}$, $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$ is entirely similar to the narrowing tree from u. Just as we have a folding narrowing graph $FNG_{\mathcal{R}}(u)$ for the $\rightsquigarrow_{R/(E \cup B)}$-narrowing tree, we also have a folding narrowing graph (a Kripke structure!) $FNG^{\Pi}_{\mathcal{R}}(u)$ for $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$.

The Folding $\leadsto_{\Pi}$-narrowing graph from u

Replacing $\leadsto_{R/(E \cup B)}$ by $\leadsto_{\Pi}$, $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$ is entirely similar to the narrowing tree from u. Just as we have a folding narrowing graph $FNG_{\mathcal{R}}(u)$ for the $\leadsto_{R/(E \cup B)}$-narrowing tree, we also have a folding narrowing graph (a Kripke structure!) $FNG^{\Pi}_{\mathcal{R}}(u)$ for $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$.

The construction of $FNG^{\Pi}_{\mathcal{R}}(u)$ is entirely similar to that of $FNG_{\mathcal{R}}(u)$ in Lecture 24, just replacing the folding relation $v \preccurlyeq_{E \cup B} w$ by the folding relation $v \preccurlyeq^{\Pi}_{E \cup D \cup B} w$ defined by the equivalence:

The Folding $\rightsquigarrow_{\Pi}$-narrowing graph from u

Replacing $\rightsquigarrow_{R/(E \cup B)}$ by $\rightsquigarrow_{\Pi}$, $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$ is entirely similar to the narrowing tree from u. Just as we have a folding narrowing graph $FNG_{\mathcal{R}}(u)$ for the $\rightsquigarrow_{R/(E \cup B)}$-narrowing tree, we also have a folding narrowing graph (a Kripke structure!) $FNG^{\Pi}_{\mathcal{R}}(u)$ for $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$.

The construction of $FNG^{\Pi}_{\mathcal{R}}(u)$ is entirely similar to that of $FNG_{\mathcal{R}}(u)$ in Lecture 24, just replacing the folding relation $v \preccurlyeq_{E \cup B} w$ by the folding relation $v \preccurlyeq^{\Pi}_{E \cup D \cup B} w$ defined by the equivalence:

$$v \preccurlyeq^{\Pi}_{E \cup D \cup B} w \quad \Leftrightarrow_{def} \quad v \preccurlyeq_{E \cup B} w \wedge \forall p \in \Pi,\ (v \models p)!_{\vec{E \cup D},B} = (w \models p)!_{\vec{E \cup D},B}.$$

The Folding $\rightsquigarrow_\Pi$-narrowing graph from u

Replacing $\rightsquigarrow_{R/(E \cup B)}$ by $\rightsquigarrow_\Pi$, $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$ is entirely similar to the narrowing tree from u. Just as we have a folding narrowing graph $FNG_\mathcal{R}(u)$ for the $\rightsquigarrow_{R/(E \cup B)}$-narrowing tree, we also have a folding narrowing graph (a Kripke structure!) $FNG^\Pi_\mathcal{R}(u)$ for $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$.

The construction of $FNG^\Pi_\mathcal{R}(u)$ is entirely similar to that of $FNG_\mathcal{R}(u)$ in Lecture 24, just replacing the folding relation $v \preccurlyeq_{E \cup B} w$ by the folding relation $v \preccurlyeq^\Pi_{E \cup D \cup B} w$ defined by the equivalence:

$$v \preccurlyeq^\Pi_{E \cup D \cup B} w \quad \Leftrightarrow_{def} \quad v \preccurlyeq_{E \cup B} w \wedge \forall p \in \Pi,\ (v \models p)!_{E \vec{\cup} D, B} = (w \models p)!_{E \vec{\cup} D, B}.$$

The Faithfulness Theorem for $FNG_\mathcal{R}(u)$ in Lecture 24, pg. 13, generalizes to (see Theorems 8 and 12 in Appendix 2):

Theorem

For $\varphi \in LTL(\Pi)$ (resp. φ a safety formula) we have:

$$FNG^\Pi_\mathcal{R}(u), u \models \varphi \quad \Rightarrow \quad (resp.\ \Leftrightarrow)\ \mathcal{NK}(\mathcal{R}, State)_\Pi(u), u \models \varphi.$$

Π-(Bi)Simulation maps of Kripke Structures

A (Π-)simulation (resp. (Π-)bisimulation) map $f : \mathcal{A} \to \mathcal{B}$ of Kripke structures over Π is, by definition, a simulation (resp. bisimulation) map of the underlying transition systems (see Lecture 25) s.t. for each $p \in \Pi$, and $a \in A$ we have $a \models_{\mathcal{A}} p \Leftrightarrow f(a) \models_{\mathcal{B}} p$.

Π-(Bi)Simulation maps of Kripke Structures

A (Π-)simulation (resp. (Π-)bisimulation) map $f : \mathcal{A} \rightarrow \mathcal{B}$ of Kripke structures over Π is, by definition, a simulation (resp. bisimulation) map of the underlying transition systems (see Lecture 25) s.t. for each $p \in \Pi$, and $a \in A$ we have $a \models_{\mathcal{A}} p \Leftrightarrow f(a) \models_{\mathcal{B}} p$. The following theorem holds for a Π-(bi)simulation map between Kripke structures (see Appendix 1):

Π-(Bi)Simulation maps of Kripke Structures

A (Π-)simulation (resp. (Π-)bisimulation) map $f : \mathcal{A} \to \mathcal{B}$ of Kripke structures over Π is, by definition, a simulation (resp. bisimulation) map of the underlying transition systems (see Lecture 25) s.t. for each $p \in \Pi$, and $a \in A$ we have $a \models_{\mathcal{A}} p \Leftrightarrow f(a) \models_{\mathcal{B}} p$. The following theorem holds for a Π-(bi)simulation map between Kripke structures (see Appendix 1):

Theorem

If $f : \mathcal{A} \to \mathcal{B}$ is a Π-simulation (resp. Π-bisimulation) map of Kripke structures over Π, then, for any $a \in A$ and $\varphi \in LTL(\Pi)$,

$$\mathcal{B}, f(a) \models \varphi \;\Rightarrow\; (resp. \;\Leftrightarrow)\; \mathcal{A}, a \models \varphi.$$

Π-(Bi)Simulation maps of Kripke Structures

A (Π-)simulation (resp. (Π-)bisimulation) map $f : \mathcal{A} \to \mathcal{B}$ of Kripke structures over Π is, by definition, a simulation (resp. bisimulation) map of the underlying transition systems (see Lecture 25) s.t. for each $p \in \Pi$, and $a \in A$ we have $a \models_{\mathcal{A}} p \Leftrightarrow f(a) \models_{\mathcal{B}} p$. The following theorem holds for a Π-(bi)simulation map between Kripke structures (see Appendix 1):

Theorem

If $f : \mathcal{A} \to \mathcal{B}$ is a Π-simulation (resp. Π-bisimulation) map of Kripke structures over Π, then, for any $a \in A$ and $\varphi \in LTL(\Pi)$,

$$\mathcal{B}, f(a) \models \varphi \;\Rightarrow\; (resp.\; \Leftrightarrow)\; \mathcal{A}, a \models \varphi.$$

If the satisfaction of state predicates Π in a topmost $\mathcal{R} = (\Sigma, E \cup B, R)$ is defined by equations D s.t. $E \cup D \cup B$ is FVP, then an equational abstraction (resp. bisimilar equational abstraction) $\mathcal{R}/G$ such that $E \cup D \cup G \cup B$ is FVP will define a Π-simulation (resp. Π-bisimulation) map $[\_]_{E \cup G \cup B}$ between the Kripke structures $\mathcal{K}(\mathcal{R}, State)_{\Pi}$ and $\mathcal{K}(\mathcal{R}/\mathcal{G}, State)_{\Pi}$, provided $E \cup D \cup G \cup B$ protects the Booleans.

Π-(Bi)Simulation maps of Kripke Structures

A (Π-)simulation (resp. (Π-)bisimulation) map $f : \mathcal{A} \to \mathcal{B}$ of Kripke structures over Π is, by definition, a simulation (resp. bisimulation) map of the underlying transition systems (see Lecture 25) s.t. for each $p \in \Pi$, and $a \in A$ we have $a \models_{\mathcal{A}} p \Leftrightarrow f(a) \models_{\mathcal{B}} p$. The following theorem holds for a Π-(bi)simulation map between Kripke structures (see Appendix 1):

Theorem

If $f : \mathcal{A} \to \mathcal{B}$ is a Π-simulation (resp. Π-bisimulation) map of Kripke structures over Π, then, for any $a \in A$ and $\varphi \in LTL(\Pi)$,

$$\mathcal{B}, f(a) \models \varphi \;\Rightarrow\; (resp.\; \Leftrightarrow)\; \mathcal{A}, a \models \varphi.$$

If the satisfaction of state predicates Π in a topmost $\mathcal{R} = (\Sigma, E \cup B, R)$ is defined by equations D s.t. $E \cup D \cup B$ is FVP, then an equational abstraction (resp. bisimilar equational abstraction) $\mathcal{R}/G$ such that $E \cup D \cup G \cup B$ is FVP will define a Π-simulation (resp. Π-bisimulation) map $[\_]_{E \cup G \cup B}$ between the Kripke structures $\mathcal{K}(\mathcal{R}, State)_\Pi$ and $\mathcal{K}(\mathcal{R}/\mathcal{G}, State)_\Pi$, provided $E \cup D \cup G \cup B$ protects the Booleans.

Symbolic State Space Reduction Theorem

Under the assumptions in pg. 7, let $\mathcal{R}/G$ be an equational abstraction (resp. bisimilar equational abstraction) defining a Π-simulation (resp. Π-bisimulation) map $[\_]_{E \cup G \cup B}$ between $\mathcal{K}(\mathcal{R}, State)_{\Pi}$ and $\mathcal{K}(\mathcal{R}/\mathcal{G}, State)_{\Pi}$. Then we have (see proof in Appendix 1):

Symbolic State Space Reduction Theorem

Under the assumptions in pg. 7, let $\mathcal{R}/G$ be an equational abstraction (resp. bisimilar equational abstraction) defining a Π-simulation (resp. Π-bisimulation) map $[\_]_{E \cup G \cup B}$ between $\mathcal{K}(\mathcal{R}, State)_{\Pi}$ and $\mathcal{K}(\mathcal{R}/\mathcal{G}, State)_{\Pi}$. Then we have (see proof in Appendix 1):

Theorem

1. *If* $[\_]_{E \cup G \cup B}$ *is a* Π*-simulation map, for each* u *and* $\varphi \in LTL(\Pi)$*,*

$$FNG^{\Pi}_{\mathcal{R}/\mathcal{G}}(u), u \models \varphi \Rightarrow \mathcal{NK}(\mathcal{R}/\mathcal{G}, State)_{\Pi}(u), u \models \varphi \Rightarrow \mathcal{NK}(\mathcal{R}, State)_{\Pi}(u), u \models \varphi.$$

Symbolic State Space Reduction Theorem

Under the assumptions in pg. 7, let $\mathcal{R}/G$ be an equational abstraction (resp. bisimilar equational abstraction) defining a Π-simulation (resp. Π-bisimulation) map $[\_]_{E \cup G \cup B}$ between $\mathcal{K}(\mathcal{R}, State)_{\Pi}$ and $\mathcal{K}(\mathcal{R}/\mathcal{G}, State)_{\Pi}$. Then we have (see proof in Appendix 1):

Theorem

1. *If* $[\_]_{E \cup G \cup B}$ *is a* Π*-simulation map, for each* u *and* $\varphi \in LTL(\Pi)$,

   $FNG^{\Pi}_{\mathcal{R}/\mathcal{G}}(u), u \models \varphi \Rightarrow \mathcal{NK}(\mathcal{R}/\mathcal{G}, State)_{\Pi}(u), u \models \varphi \Rightarrow \mathcal{NK}(\mathcal{R}, State)_{\Pi}(u), u \models \varphi.$

2. *If* $[\_]_{E \cup G \cup B}$ *is a* Π*-bisimulation map, for each* u *and* $\varphi \in LTL(\Pi)$,

   $FNG^{\Pi}_{\mathcal{R}/\mathcal{G}}(u), u \models \varphi \Rightarrow \mathcal{NK}(\mathcal{R}/\mathcal{G}, State)_{\Pi}(u), u \models \varphi \Leftrightarrow \mathcal{NK}(\mathcal{R}, State)_{\Pi}(u), u \models \varphi.$

Symbolic State Space Reduction Theorem

Under the assumptions in pg. 7, let $\mathcal{R}/G$ be an equational abstraction (resp. bisimilar equational abstraction) defining a Π-simulation (resp. Π-bisimulation) map $[\_]_{E \cup G \cup B}$ between $\mathcal{K}(\mathcal{R}, State)_{\Pi}$ and $\mathcal{K}(\mathcal{R}/\mathcal{G}, State)_{\Pi}$. Then we have (see proof in Appendix 1):

Theorem

1. *If $[\_]_{E \cup G \cup B}$ is a Π-simulation map, for each u and $\varphi \in LTL(\Pi)$,*
   $FNG^{\Pi}_{\mathcal{R}/\mathcal{G}}(u), u \models \varphi \Rightarrow \mathcal{NK}(\mathcal{R}/\mathcal{G}, State)_{\Pi}(u), u \models \varphi \Rightarrow \mathcal{NK}(\mathcal{R}, State)_{\Pi}(u), u \models \varphi.$
2. *If $[\_]_{E \cup G \cup B}$ is a Π-bisimulation map, for each u and $\varphi \in LTL(\Pi)$,*
   $FNG^{\Pi}_{\mathcal{R}/\mathcal{G}}(u), u \models \varphi \Rightarrow \mathcal{NK}(\mathcal{R}/\mathcal{G}, State)_{\Pi}(u), u \models \varphi \Leftrightarrow \mathcal{NK}(\mathcal{R}, State)_{\Pi}(u), u \models \varphi.$

Furthermore, if φ a safety formula, the leftmost implication in (1) and (2) becomes an equivalence.

Bounded Narrowing-Based LTL Model Checking

- Construct a depth $\leq k$ under-approximation of the folding narrowing graph (and Kripke structure) $FNG^{\Pi}_{\mathcal{R}}(u)$

Bounded Narrowing-Based LTL Model Checking

- Construct a depth $\leq k$ under-approximation of the folding narrowing graph (and Kripke structure) $FNG^{\Pi}_{\mathcal{R}}(u)$ (a more expensive, but more accurate, version under-approximates $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$).

Bounded Narrowing-Based LTL Model Checking

- Construct a depth $\leq k$ under-approximation of the folding narrowing graph (and Kripke structure) $FNG^{\Pi}_{\mathcal{R}}(u)$ (a more expensive, but more accurate, version under-approximates $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$).

Algorithm: Given a bound n, incrementally build a depth $\leq k$ under-approximation of $FNG^{\Pi}_{\mathcal{R}}(u)$, increasing $k \leq n$ iteratively.

Bounded Narrowing-Based LTL Model Checking

- Construct a depth $\leq k$ under-approximation of the folding narrowing graph (and Kripke structure) $FNG^{\Pi}_{\mathcal{R}}(u)$ (a more expensive, but more accurate, version under-approximates $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$).

Algorithm: Given a bound n, incrementally build a depth $\leq k$ under-approximation of $FNG^{\Pi}_{\mathcal{R}}(u)$, increasing $k \leq n$ iteratively.

1. Apply a standard explicit-state LTL model checking algorithm to verify φ in the depth $\leq k$ under-approximation of $FNG^{\Pi}_{\mathcal{R}}(u)$.
   If a counterexample is found, stop and return the counterexample.

Bounded Narrowing-Based LTL Model Checking

- Construct a depth $\leq k$ under-approximation of the folding narrowing graph (and Kripke structure) $FNG_{\mathcal{R}}^{\Pi}(u)$ (a more expensive, but more accurate, version under-approximates $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$).

Algorithm: Given a bound n, incrementally build a depth $\leq k$ under-approximation of $FNG_{\mathcal{R}}^{\Pi}(u)$, increasing $k \leq n$ iteratively.

1. Apply a standard explicit-state LTL model checking algorithm to verify φ in the depth $\leq k$ under-approximation of $FNG_{\mathcal{R}}^{\Pi}(u)$.
   If a counterexample is found, stop and return the counterexample.
2. Suppose that there is *no* counterexample at depth $\leq k$.

Bounded Narrowing-Based LTL Model Checking

- Construct a depth $\leq k$ under-approximation of the folding narrowing graph (and Kripke structure) $FNG^{\Pi}_{\mathcal{R}}(u)$ (a more expensive, but more accurate, version under-approximates $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$).

Algorithm: Given a bound n, incrementally build a depth $\leq k$ under-approximation of $FNG^{\Pi}_{\mathcal{R}}(u)$, increasing $k \leq n$ iteratively.

1. Apply a standard explicit-state LTL model checking algorithm to verify φ in the depth $\leq k$ under-approximation of $FNG^{\Pi}_{\mathcal{R}}(u)$. If a counterexample is found, stop and return the counterexample.
2. Suppose that there is *no* counterexample at depth $\leq k$.
    1. If $k = n$, stop and report that the model does not violate φ up to the current bound n.

Bounded Narrowing-Based LTL Model Checking

- Construct a depth $\leq k$ under-approximation of the folding narrowing graph (and Kripke structure) $FNG^{\Pi}_{\mathcal{R}}(u)$ (a more expensive, but more accurate, version under-approximates $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$).

Algorithm: Given a bound n, incrementally build a depth $\leq k$ under-approximation of $FNG^{\Pi}_{\mathcal{R}}(u)$, increasing $k \leq n$ iteratively.

1. Apply a standard explicit-state LTL model checking algorithm to verify φ in the depth $\leq k$ under-approximation of $FNG^{\Pi}_{\mathcal{R}}(u)$.
   If a counterexample is found, stop and return the counterexample.
2. Suppose that there is *no* counterexample at depth $\leq k$.
   1. If $k = n$, stop and report that the model does not violate φ up to the current bound n.
   2. Otherwise, generate the depth $\leq k + 1$ under-approximation of $FNG^{\Pi}_{\mathcal{R}}(u)$

Bounded Narrowing-Based LTL Model Checking

- Construct a depth $\leq k$ under-approximation of the folding narrowing graph (and Kripke structure) $FNG^{\Pi}_{\mathcal{R}}(u)$ (a more expensive, but more accurate, version under-approximates $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$).

Algorithm: Given a bound n, incrementally build a depth $\leq k$ under-approximation of $FNG^{\Pi}_{\mathcal{R}}(u)$, increasing $k \leq n$ iteratively.

1. Apply a standard explicit-state LTL model checking algorithm to verify φ in the depth $\leq k$ under-approximation of $FNG^{\Pi}_{\mathcal{R}}(u)$. If a counterexample is found, stop and return the counterexample.
2. Suppose that there is *no* counterexample at depth $\leq k$.
   1. If $k = n$, stop and report that the model does not violate φ up to the current bound n.
   2. Otherwise, generate the depth $\leq k+1$ under-approximation of $FNG^{\Pi}_{\mathcal{R}}(u)$
      1. If no new nodes are added to the $\leq k$ under-approximation, $FNG^{\Pi}_{\mathcal{R}}(u)$ has been actually generated! Then return *true*;

Bounded Narrowing-Based LTL Model Checking

- Construct a depth $\leq k$ under-approximation of the folding narrowing graph (and Kripke structure) $FNG_{\mathcal{R}}^{\Pi}(u)$ (a more expensive, but more accurate, version under-approximates $\mathcal{NK}(\mathcal{R}, State)_{\Pi}(u)$).

Algorithm: Given a bound n, incrementally build a depth $\leq k$ under-approximation of $FNG_{\mathcal{R}}^{\Pi}(u)$, increasing $k \leq n$ iteratively.

1. Apply a standard explicit-state LTL model checking algorithm to verify φ in the depth $\leq k$ under-approximation of $FNG_{\mathcal{R}}^{\Pi}(u)$.
   If a counterexample is found, stop and return the counterexample.
2. Suppose that there is *no* counterexample at depth $\leq k$.
   1. If $k = n$, stop and report that the model does not violate φ up to the current bound n.
   2. Otherwise, generate the depth $\leq k+1$ under-approximation of $FNG_{\mathcal{R}}^{\Pi}(u)$
      1. If no new nodes are added to the $\leq k$ under-approximation, $FNG_{\mathcal{R}}^{\Pi}(u)$ has been actually generated! Then return *true*;
      2. Otherwise, go to Step 1 with the depth $\leq k+1$ under-approximation of $FNG_{\mathcal{R}}^{\Pi}(u)$.

Maude's Logical LTL Model Checker Tool

- Maude's Logical LTL Model Checker supports narrowing-based LTL model checking with the techniques discussed in this lecture

Maude's Logical LTL Model Checker Tool

- Maude's Logical LTL Model Checker supports narrowing-based LTL model checking with the techniques discussed in this lecture `https://maude.cs.uiuc.edu/tools/lmc/`

Maude's Logical LTL Model Checker Tool

- Maude's Logical LTL Model Checker supports narrowing-based LTL model checking with the techniques discussed in this lecture
  `https://maude.cs.uiuc.edu/tools/lmc/`
  See also the CS 476 web page for details on how to use the tool and the tool's manual with examples.

Maude's Logical LTL Model Checker Tool

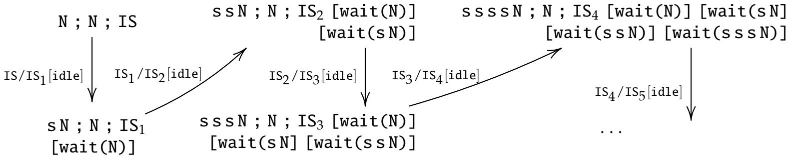
- Maude's Logical LTL Model Checker supports narrowing-based LTL model checking with the techniques discussed in this lecture
  `https://maude.cs.uiuc.edu/tools/lmc/`
  See also the CS 476 web page for details on how to use the tool and the tool's manual with examples.
- Various LTL properties verified for examples such as:

Maude's Logical LTL Model Checker Tool

- Maude's Logical LTL Model Checker supports narrowing-based LTL model checking with the techniques discussed in this lecture `https://maude.cs.uiuc.edu/tools/lmc/`
  See also the CS 476 web page for details on how to use the tool and the tool's manual with examples.
- Various LTL properties verified for examples such as:
  1. Lamport's Bakery protocol
  2. Readers-Writers problem
  3. Readers-Writers problem (simplified)
  4. Dijkstra's mutual exclusion algorithm
  5. Burns's mutual exclusion algorithm
  6. Token ring mutual exclusion
  7. Vending Machine example
  8. Plotter example

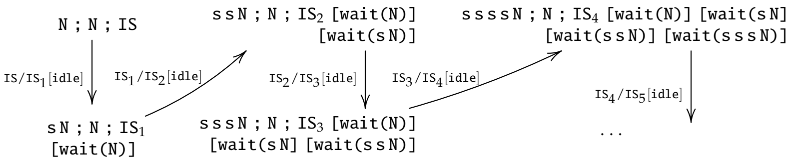
Output 1/3: Bounded Model Checking without Folding

```
Maude> (lmc [10] N:Nat ; N:Nat ; IS:ProcIdleSet |= [] ex? .)
logical model check in BAKERY-SATISFACTION :
  N:Nat ; N:Nat ; IS:ProcIdleSet |= [] ex?
result:
  no counterexample found within bound 10
```



Output 2/3: Bounded Model Checking with Folding

```
Maude> (lfmc [50] N:Nat ; N:Nat ; IS:ProcIdleSet |= [] ex? .)
logical folding model check in BAKERY-SATISFACTION :
  N:Nat ; N:Nat ; IS:ProcIdleSet |= [] ex?
result:
  no counterexample found within bound 50
```



Output 3/3: Unbounded Model Checking with a Bisimilar Equational Abstraction

```
Maude> (lfmc N:Nat ; N:Nat ; IS:ProcIdleSet |= [] ex? .)
logical folding model check in BAKERY-SATISFACTION-ABS :
  N:Nat ; N:Nat ; IS:ProcIdleSet |= [] ex?
result:
  true
```