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 $\rightsquigarrow_{R/(E \cup B)}$ 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 $\Pi$, and another ground instance $u_\tau$ may satisfy a different set of predicates in $\Pi$. However, if $\mathcal{R}$ is deadlock-free, and the equations $D$ defining the satisfaction relation $u \models p$ between terms of top sort $\text{State}$ and state predicates $\Pi$ 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 $\mathcal{N}_K(\mathcal{R}, \text{State})_{\Pi(u)}$, whose symbolic transitions are performed by a $\Pi$-aware narrowing relation $\rightsquigarrow_{\Pi}$ explained in what follows.
Narrowing-Based Symbolic LTL Model Checking

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

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 \( u\varphi \) may satisfy some predicates in \( \Pi \), and another ground instance \( u\tau \) may satisfy a different set of predicates in \( \Pi \).
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 $\mathcal{R}$?

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 $\Pi$, and another ground instance $u\tau$ may satisfy a different set of predicates in $\Pi$.

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 $\Pi$ 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 $\mathcal{NK}(\mathcal{R}, \text{State})_{\Pi}(u)$, whose symbolic transitions are performed by a $\Pi$-aware narrowing relation $\leadsto_{\Pi}$ explained in what follows.
The Narrowing Relation $\leadsto_\Pi$

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 $\text{State}$, 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 $\Pi$-aware narrowing relation between terms $u, w \in T_{\Sigma, \text{State}}(X)$ is defined as follows:
The Narrowing Relation $\sim_{\Pi}$

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 $\text{State}$, 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 $\Pi$-aware narrowing relation between terms $u, w \in T_{\Sigma, \text{State}}(X)$ is defined as follows:

$$u \sim_{\Pi}^{\alpha \gamma} w$$

holds iff (by definition)
The Narrowing Relation $\leadsto_{\Pi}$

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 $\Pi$-aware narrowing relation between terms $u, w \in T_{\Sigma,State}(X)$ is defined as follows:

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

holds iff (by definition)

- $\exists v$ s.t. $u \overset{\alpha}{\leadsto}_{R/(E \cup B)} v$
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 $\Pi$-aware narrowing relation between terms $u, w \in T_{\Sigma,State}(X)$ is defined as follows:

$$u \xrightarrow{\alpha \gamma} w$$

holds iff (by definition)

- $\exists v \text{ s.t. } u \xrightarrow{\alpha} R/(E \cup B) v$
- $\exists (b_1, \ldots, b_k) \in \{true, false\}^k$
The Narrowing Relation $\sim_{\Pi}$

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 $\text{State}$, 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 $\Pi$-aware narrowing relation between terms $u, w \in T_{\Sigma,\text{State}}(X)$ is defined as follows:

$$u \stackrel{\alpha\gamma}{\sim_{\Pi}} w$$

holds iff (by definition)

- $\exists v \text{ s.t. } u \stackrel{\alpha}{\sim_{R/(E\cup B)}} v$
- $\exists (b_1, \ldots, b_k) \in \{\text{true}, \text{false}\}^k$
- $\exists \gamma \in \text{Unif}_{E \cup D \cup B}(v \models p_1 = b_1 \land \ldots \land v \models p_k = b_k)$
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 $\Pi$-aware narrowing relation between terms $u, w \in T_{\Sigma,\text{State}}(X)$ is defined as follows:

$u \xrightarrow{\alpha \gamma} w$

holds iff (by definition)

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

such that $w = v\gamma$. 
The Kripke Structure $\mathcal{NK}(\mathcal{R}, \text{State})_\Pi(u)$

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

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

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

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

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

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

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

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

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

**Theorem**

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

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

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

**Theorem**

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

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

By the above Theorem, the Kripke structure $\mathcal{N}\mathcal{K}(R, \text{State})_{\Pi}(u)$ 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 $\mathcal{NK}(\mathcal{R}, \text{State})\Pi(u)$

By the above Theorem, the Kripke structure $\mathcal{NK}(\mathcal{R}, \text{State})\Pi(u)$ 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 $\mathcal{NK}(\mathcal{R}, \text{State})\Pi(u)$ to a finite state space:
State Space Reduction in $\mathcal{NK}(\mathcal{R}, \text{State})_\Pi(u)$

By the above Theorem, the Kripke structure $\mathcal{NK}(\mathcal{R}, \text{State})_\Pi(u)$ 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 $\mathcal{NK}(\mathcal{R}, \text{State})_\Pi(u)$ 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.
State Space Reduction in $\mathcal{NK}(\mathcal{R}, \text{State})_{\Pi}(u)$

By the above Theorem, the Kripke structure $\mathcal{NK}(\mathcal{R}, \text{State})_{\Pi}(u)$ 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 $\mathcal{NK}(u)$ is finite. When $\mathcal{NK}(u)$ is infinite, we can try one of the following four possibilities to reduce the state space of $\mathcal{NK}(\mathcal{R}, \text{State})_{\Pi}(u)$ 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 $\mathcal{R}/G$. 

Let us explore these possibilities in more detail.
State Space Reduction in $\mathcal{NK}(\mathcal{R}, \text{State})_{\Pi}(u)$

By the above Theorem, the Kripke structure $\mathcal{NK}(\mathcal{R}, \text{State})_{\Pi}(u)$ 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 $\mathcal{NK}(u)$ is finite. When $\mathcal{NK}(u)$ is infinite, we can try one of the following four possibilities to reduce the state space of $\mathcal{NK}(\mathcal{R}, \text{State})_{\Pi}(u)$ 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.
State Space Reduction in $NK(\mathcal{R}, State)_{\Pi}(u)$

By the above Theorem, the Kripke structure $NK(\mathcal{R}, State)_{\Pi}(u)$ 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 $NK(\mathcal{R}, State)_{\Pi}(u)$ to a finite state space:

1. Perform LTL model checking by folding variant narrowing, provided the folding $\sim_{\Pi}$-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 $\sim_{\Pi}$-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 $\sim_{\Pi}$-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 $\rightsquigarrow_\Pi$, $\mathcal{N}\mathcal{K}(R, State)_\Pi(u)$ 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!) $FNG_\mathcal{R}^{\Pi}(u)$ for $\mathcal{NK}(\mathcal{R}, State)_\Pi(u)$. 
The Folding $\rightsquigarrow_{\Pi}$-narrowing graph from $u$

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

The construction of $\mathcal{FNG}_{\Pi}(u)$ is entirely similar to that of $\mathcal{FNG}_{\mathcal{R}}(u)$ in Lecture 24, just replacing the folding relation $v \simeq_{E\cup B} w$ by the folding relation $v \simeq_{E\cup D\cup B} w$ defined by the equivalence:
The Folding $\rightsquigarrow_{\Pi}$-narrowing graph from $u$

Replacing $\rightsquigarrow_{R/(E\cup B)}$ by $\rightsquigarrow_{\Pi}$, $\mathcal{N}\mathcal{K}(\mathcal{R}, State)_{\Pi}(u)$ 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!) $FNG_{\mathcal{R}}(u)$ for $\mathcal{N}\mathcal{K}(\mathcal{R}, State)_{\Pi}(u)$.

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

$$v \preceq_{E\cup D\cup B} w \Leftrightarrow_{def} v \preceq_{E\cup B} w \land \forall p \in \Pi, (v \models p)!_{E\cup D,B} = (w \models p)!_{E\cup D,B}.$$
The Folding $\sim_{\Pi}$-narrowing graph from $u$

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

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 $v \preceq_{E\cup B} w$ by the folding relation $v \preceq_{E\cup D\cup B} w$ defined by the equivalence:

$$v \preceq_{E\cup D\cup B} w \iff v \preceq_{E\cup B} w \land \forall p \in \Pi, (v \models p)!_{E\cup D,B} = (w \models p)!_{E\cup D,B}.$$ 

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. $\varphi$ a safety formula) we have:

$$FNG_{\Pi}^{\mathcal{R}}(u), u \models \varphi \Rightarrow \text{(resp. } \Leftrightarrow\text{)} \mathcal{N}\mathcal{K}(\mathcal{R},\text{State})_{\Pi}(u), u \models \varphi.$$
### Π-(Bi)Simulation maps of Kripke Structures

A (Π-)simulation (resp. (Π-)bisimulation) map $f : A \to B$ of Kripke structures over $\Pi$ 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 $a \models_A p \iff f(a) \models_B p$. 

The following theorem holds for a Π-(bi)simulation map between Kripke structures (see Appendix 1):

**Theorem**

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

$B, f(a) \models \varphi \implies (\text{resp. } \iff) A, a \models \varphi$. 

If the satisfaction of state predicates $\Π$ in a topmost $R = (\Sigma, E \cup B, R)$ is defined by equations $\Delta$ s.t. $E \cup D \cup B$ is FVP, then an equational abstraction (resp. bisimilar equational abstraction) $R / G$ such that $E \cup G \cup B$ is FVP will define a Π-simulation (resp. Π-bisimulation) map $[E \cup G \cup B]$ between the Kripke structures $K(\mathcal{R}, \text{State})_\Pi(u)$ and $K(\mathcal{R} / G, \text{State})_\Pi(u)$, 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 $f : A \rightarrow B$ of Kripke structures over $\Pi$ 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 $a \models_A p \iff f(a) \models_B p$. 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 \( f : A \rightarrow B \) of Kripke structures over \( \Pi \) 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 \( a \models_A p \iff f(a) \models_B p \). The following theorem holds for a Π-(bi)simulation map between Kripke structures (see Appendix 1):

**Theorem**

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

\[
B, f(a) \models \varphi \ \Rightarrow \ (\text{resp. } \Leftrightarrow) \ A, a \models \varphi.
\]
**Π-(Bi)Simulation maps of Kripke Structures**

A (Π-)simulation (resp. (Π-)bisimulation) map \( f : A \to B \) of Kripke structures over \( \Pi \) 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 \( a \models_A p \iff f(a) \models_B p \).

The following theorem holds for a Π-(bi)simulation map between Kripke structures (see Appendix 1):

**Theorem**

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

\[
B, f(a) \models \varphi \Rightarrow (\text{resp. } \iff) A, a \models \varphi.
\]

If the satisfaction of state predicates \( \Pi \) 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 \( \mathcal{K}(\mathcal{R}, \text{State})_\Pi \) and \( \mathcal{K}(\mathcal{R}/G, \text{State})_\Pi \), 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 \( f : A \rightarrow B \) of Kripke structures over \( \Pi \) 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 \( a \models_A p \iff f(a) \models_B p \). The following theorem holds for a Π-(bi)simulation map between Kripke structures (see Appendix 1):

**Theorem**

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

\[
B, f(a) \models \varphi \Rightarrow (\text{resp. } \iff) A, a \models \varphi.
\]

If the satisfaction of state predicates \( \Pi \) 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 \([\cdot]_{E \cup G \cup B} \) between the Kripke structures \( \mathcal{K}(\mathcal{R}, \text{State})_\Pi \) and \( \mathcal{K}(\mathcal{R}/G, \text{State})_\Pi \), 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 $\Pi$-simulation (resp. $\Pi$-bisimulation) map $[-]_{EUGUB}$ between $\mathcal{K}(\mathcal{R}, \text{State})_\Pi$ and $\mathcal{K}(\mathcal{R}/G, \text{State})_\Pi$. Then we have (see proof in Appendix 1):

1. If $[-]_{EUGUB}$ is a $\Pi$-simulation map, for each $u$ and $\phi \in \text{LTL}(\Pi)$, $\text{FNG}_{\Pi \mathcal{R}/G}(u)$, $u\mid = \phi \Rightarrow \text{NK}(\mathcal{R}/G, \text{State})_\Pi(u)$, $u\mid = \phi \Rightarrow \text{NK}(\mathcal{R}, \text{State})_\Pi(u)$.

2. If $[-]_{EUGUB}$ is a $\Pi$-bisimulation map, for each $u$ and $\phi \in \text{LTL}(\Pi)$, $\text{FNG}_{\Pi \mathcal{R}/G}(u)$, $u\mid = \phi \Rightarrow \text{NK}(\mathcal{R}/G, \text{State})_\Pi(u)$, $u\mid = \phi \Leftrightarrow \text{NK}(\mathcal{R}, \text{State})_\Pi(u)$.

Furthermore, if $\phi$ a safety formula, the leftmost implication in (1) and (2) becomes an equivalence.
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 $\Pi$-simulation (resp. $\Pi$-bisimulation) map $\llbracket - \rrbracket_{E \cup G \cup B}$ between $\mathcal{K}(\mathcal{R}, \text{State})_{\Pi}$ and $\mathcal{K}(\mathcal{R}/G, \text{State})_{\Pi}$. Then we have (see proof in Appendix 1):

**Theorem**

1. If $\llbracket - \rrbracket_{E \cup G \cup B}$ is a $\Pi$-simulation map, for each $u$ and $\varphi \in LTL(\Pi)$,
   \[ FNG^\Pi_{\mathcal{R}/G}(u), u \models \varphi \Rightarrow \mathcal{N}\mathcal{K}(\mathcal{R}/G, \text{State})_{\Pi}(u), u \models \varphi \Rightarrow \mathcal{N}\mathcal{K}(\mathcal{R}, \text{State})_{\Pi}(u), u \models \varphi. \]
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 $\Pi$-simulation (resp. $\Pi$-bisimulation) map $[-]_{EUGUB}$ between $\mathcal{K}(\mathcal{R}, \text{State})_{\Pi}$ and $\mathcal{K}(\mathcal{R}/G, \text{State})_{\Pi}$. Then we have (see proof in Appendix 1):

**Theorem**

1. If $[-]_{EUGUB}$ is a $\Pi$-simulation map, for each $u$ and $\varphi \in LTL(\Pi)$,
   \[ FNG_{\mathcal{R}/G}(u), u \models \varphi \Rightarrow NK(\mathcal{R}/G, \text{State})_{\Pi}(u), u \models \varphi \Rightarrow NK(\mathcal{R}, \text{State})_{\Pi}(u), u \models \varphi. \]

2. If $[-]_{EUGUB}$ is a $\Pi$-bisimulation map, for each $u$ and $\varphi \in LTL(\Pi)$,
   \[ FNG_{\mathcal{R}/G}(u), u \models \varphi \Rightarrow NK(\mathcal{R}/G, \text{State})_{\Pi}(u), u \models \varphi \Leftrightarrow NK(\mathcal{R}, \text{State})_{\Pi}(u), u \models \varphi. \]
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 $\Pi$-simulation (resp. $\Pi$-bisimulation) map $[\cdot]_{EUGUB}$ between $\mathcal{K}(\mathcal{R}, \text{State})_{\Pi}$ and $\mathcal{K}(\mathcal{R}/G, \text{State})_{\Pi}$. Then we have (see proof in Appendix 1):

**Theorem**

1. If $[\cdot]_{EUGUB}$ is a $\Pi$-simulation map, for each $u$ and $\varphi \in LTL(\Pi)$,
   \[
   FNG_{\mathcal{R}/G}(u), u \models \varphi \Rightarrow NK(\mathcal{R}/G, \text{State})_{\Pi}(u), u \models \varphi \Rightarrow NK(\mathcal{R}, \text{State})_{\Pi}(u), u \models \varphi.
   \]

2. If $[\cdot]_{EUGUB}$ is a $\Pi$-bisimulation map, for each $u$ and $\varphi \in LTL(\Pi)$,
   \[
   FNG_{\mathcal{R}/G}(u), u \models \varphi \Rightarrow NK(\mathcal{R}/G, \text{State})_{\Pi}(u), u \models \varphi \Leftrightarrow NK(\mathcal{R}, \text{State})_{\Pi}(u), u \models \varphi.
   \]

Furthermore, if $\varphi$ 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) $FNG_{\overline{R}}^{\Pi}(u)$
Bounded Narrowing-Based LTL Model Checking

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

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

Algorithm: Given a bound $n$, incrementally build a depth $\leq k$ under-approximation of $FNG_{\Pi}^{\Pi}(u)$, 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) $FNG_{\mathcal{R}}^{\Pi}(u)$ (a more expensive, but more accurate, version under-approximates $\mathcal{N}\mathcal{K}(\mathcal{R}, \text{State})_{\Pi}(u)$).

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 $\varphi$ in the depth $\leq k$ under-approximation of $FNG_{\mathcal{R}}^{\Pi}(u)$. 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) $FNG^\Pi_R(u)$ (a more expensive, but more accurate, version under-approximates $NK(R, State)_{\Pi}(u)$).

Algorithm: Given a bound $n$, incrementally build a depth $\leq k$ under-approximation of $FNG^\Pi_R(u)$, increasing $k \leq n$ iteratively.

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

2. Suppose that there is no counterexample at depth $\leq k$. If no new nodes are added to the $\leq k$ under-approximation, $FNG^\Pi_R(u)$ has been actually generated! Then return $true$; Otherwise, go to Step 1 with the depth $\leq k + 1$ under-approximation of $FNG^\Pi_R(u)$. 

Meseguer Lecture 24
Bounded Narrowing-Based LTL Model Checking

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

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

1. Apply a standard explicit-state LTL model checking algorithm to verify $\varphi$ in the depth $\leq k$ under-approximation of $\text{FNG}_R^\Pi(u)$. 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 $\varphi$ 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) $FNG^\Pi_R(u)$ (a more expensive, but more accurate, version under-approximates $\mathcal{NK}(\mathcal{R}, \mathcal{State})^\Pi(u)$).

Algorithm: Given a bound $n$, incrementally build a depth $\leq k$ under-approximation of $FNG^\Pi_R(u)$, increasing $k \leq n$ iteratively.

1. Apply a standard explicit-state LTL model checking algorithm to verify $\varphi$ in the depth $\leq k$ under-approximation of $FNG^\Pi_R(u)$. 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 $\varphi$ up to the current bound $n$.
   2. Otherwise, generate the depth $\leq k + 1$ under-approximation of $FNG^\Pi_R(u)$. 
Bounded Narrowing-Based LTL Model Checking

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

Algorithm: Given a bound $n$, incrementally build a depth $\leq k$ under-approximation of $FNG^\Pi_R(u)$, increasing $k \leq n$ iteratively.

1. Apply a standard explicit-state LTL model checking algorithm to verify $\varphi$ in the depth $\leq k$ under-approximation of $FNG^\Pi_R(u)$. 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 $\varphi$ up to the current bound $n$.
   2. Otherwise, generate the depth $\leq k + 1$ under-approximation of $FNG^\Pi_R(u)$.
      1. If no new nodes are added to the $\leq k$ under-approximation, $FNG^\Pi_R(u)$ 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) $FNG^\Pi_R(u)$ (a more expensive, but more accurate, version under-approximates $\mathcal{N}\mathcal{K}(R, \text{State})^\Pi(u)$).

Algorithm: Given a bound $n$, incrementally build a depth $\leq k$ under-approximation of $FNG^\Pi_R(u)$, increasing $k \leq n$ iteratively.

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

2. Suppose that there is no counterexample at depth $\leq k$.
   - If $k = n$, stop and report that the model does not violate $\varphi$ up to the current bound $n$.
   - Otherwise, generate the depth $\leq k + 1$ under-approximation of $FNG^\Pi_R(u)$
     - If no new nodes are added to the $\leq k$ under-approximation, $FNG^\Pi_R(u)$ has been actually generated! Then return true;
     - Otherwise, go to Step 1 with the depth $\leq k + 1$ under-approximation of $FNG^\Pi_R(u)$. 
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/](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. 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:
  - Lamport’s Bakery protocol
  - Readers-Writers problem
  - Readers-Writers problem (simplified)
  - Dijkstra’s mutual exclusion algorithm
  - Burns’s mutual exclusion algorithm
  - Token ring mutual exclusion
  - Vending Machine example
  - Plotter example
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/](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

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

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? .)
 logistical folding model check in BAKERY-SATISFACTION-ABS :
    N: Nat ; N: Nat ; IS: ProcIdleSet |= [] ex?
result:
  true