Program Verification: Lecture 26

José Meseguer

University of Illinois at Urbana-Champaign

Extending Narrowing-Based Symbolic Model Checking

So far, the narrowing-based symbolic model checking of infinite-state systems applies to topmost theories of the form $\mathcal{R} = (\Omega, B, R)$, where B is a set of equational axioms.

Extending Narrowing-Based Symbolic Model Checking

So far, the narrowing-based symbolic model checking of infinite-state systems applies to topmost theories of the form $\mathcal{R} = (\Omega, B, R)$, where B is a set of equational axioms.

This leaves out topmost theories of the form, $\mathcal{R} = (\Sigma, E \cup B, R)$.

Extending Narrowing-Based Symbolic Model Checking

So far, the narrowing-based symbolic model checking of infinite-state systems applies to topmost theories of the form $\mathcal{R} = (\Omega, B, R)$, where B is a set of equational axioms.

This leaves out topmost theories of the form, $\mathcal{R} = (\Sigma, E \cup B, R)$. But it is quite common for concurrent systems to update their states by means of auxiliary functions defined by equations E modulo B.

Extending Narrowing-Based Symbolic Model Checking

So far, the narrowing-based symbolic model checking of infinite-state systems applies to topmost theories of the form $\mathcal{R} = (\Omega, B, R)$, where B is a set of equational axioms.

This leaves out topmost theories of the form, $\mathcal{R} = (\Sigma, E \cup B, R)$. But it is quite common for concurrent systems to update their states by means of auxiliary functions defined by equations E modulo B. Can we extend narrowing to richer topmost theories?

Extending Narrowing-Based Symbolic Model Checking

So far, the narrowing-based symbolic model checking of infinite-state systems applies to topmost theories of the form $\mathcal{R} = (\Omega, B, R)$, where B is a set of equational axioms.

This leaves out topmost theories of the form, $\mathcal{R} = (\Sigma, E \cup B, R)$. But it is quite common for concurrent systems to update their states by means of auxiliary functions defined by equations E modulo B. Can we extend narrowing to richer topmost theories?

Besides symbolic verification of invariants by narrowing, since LTL allows verification of richer properties than just invariants, this raises the question:

Extending Narrowing-Based Symbolic Model Checking

So far, the narrowing-based symbolic model checking of infinite-state systems applies to topmost theories of the form $\mathcal{R} = (\Omega, B, R)$, where B is a set of equational axioms.

This leaves out topmost theories of the form, $\mathcal{R} = (\Sigma, E \cup B, R)$. But it is quite common for concurrent systems to update their states by means of auxiliary functions defined by equations E modulo B. Can we extend narrowing to richer topmost theories?

Besides symbolic verification of invariants by narrowing, since LTL allows verification of richer properties than just invariants, this raises the question: Could symbolic model checking of invariants be extended to symbolic LTL model checking of infinite-state systems?

Extending Narrowing-Based Symbolic Model Checking

So far, the narrowing-based symbolic model checking of infinite-state systems applies to topmost theories of the form $\mathcal{R} = (\Omega, B, R)$, where B is a set of equational axioms.

This leaves out topmost theories of the form, $\mathcal{R} = (\Sigma, E \cup B, R)$. But it is quite common for concurrent systems to update their states by means of auxiliary functions defined by equations E modulo B. Can we extend narrowing to richer topmost theories?

Besides symbolic verification of invariants by narrowing, since LTL allows verification of richer properties than just invariants, this raises the question: Could symbolic model checking of invariants be extended to symbolic LTL model checking of infinite-state systems?

Before answering these two questions (in the positive), this lecture first introduces some symbolic techniques needed for this purpose.

The Need for $E \cup B$-Unification

Symbolic model checking of a topmost rewrite theory $\mathcal{R} = (\Omega, B, R)$ is based on the modulo B narrowing relation $\rightsquigarrow_{R,B}$.

The Need for $E \cup B$-Unification

Symbolic model checking of a topmost rewrite theory $\mathcal{R} = (\Omega, B, R)$ is based on the modulo B narrowing relation $\leadsto_{R,B}$.

To extend this kind of symbolic model checking to admissible topmost rewrite theories of the form $\mathcal{R} = (\Sigma, E \cup B, R)$ we need to perform narrowing modulo $E \cup B$ with a relation $\leadsto_{R,E\cup B}$.

The Need for $E \cup B$-Unification

Symbolic model checking of a topmost rewrite theory $\mathcal{R} = (\Omega, B, R)$ is based on the modulo B narrowing relation $\leadsto_{R,B}$.

To extend this kind of symbolic model checking to admissible topmost rewrite theories of the form $\mathcal{R} = (\Sigma, E \cup B, R)$ we need to perform narrowing modulo $E \cup B$ with a relation $\leadsto_{R,E \cup B}$. The definition of narrowing modulo in Lecture 23 remains the same, just by generalizing B to $E \cup B$:

The Need for $E \cup B$-Unification

Symbolic model checking of a topmost rewrite theory $\mathcal{R} = (\Omega, B, R)$ is based on the modulo B narrowing relation $\leadsto_{R,B}$.

To extend this kind of symbolic model checking to admissible topmost rewrite theories of the form $\mathcal{R} = (\Sigma, E \cup B, R)$ we need to perform narrowing modulo $E \cup B$ with a relation $\leadsto_{R,E\cup B}$. The definition of narrowing modulo in Lecture 23 remains the same, just by generalizing B to $E \cup B$:

Given a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$, and a term $t \in T_\Sigma(X)$, an R-narrowing step modulo $E \cup B$, denoted $t \leadsto^{\theta}_{R,E\cup B} v$ holds iff there exists a non-variable position p in t, a rule $l \to r$ in R, and a $E \cup B$-unifier $\theta \in Unif_{E\cup B}(t|_p = l)$ such that $v = t[r]_p\theta$.

The Need for $E \cup B$-Unification

Symbolic model checking of a topmost rewrite theory $\mathcal{R} = (\Omega, B, R)$ is based on the modulo B narrowing relation $\leadsto_{R,B}$.

To extend this kind of symbolic model checking to admissible topmost rewrite theories of the form $\mathcal{R} = (\Sigma, E \cup B, R)$ we need to perform narrowing modulo $E \cup B$ with a relation $\leadsto_{R,E \cup B}$. The definition of narrowing modulo in Lecture 23 remains the same, just by generalizing B to $E \cup B$:

Given a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$, and a term $t \in T_{\Sigma}(X)$, an R-narrowing step modulo $E \cup B$, denoted $t \leadsto^{\theta}_{R,E \cup B} v$ holds iff there exists a non-variable position p in t, a rule $l \to r$ in R, and a $E \cup B$-unifier $\theta \in Unif_{E \cup B}(t|_p = l)$ such that $v = t[r]_p \theta$.

But the million-dolar question is: How do we compute a complete set $Unif_{E \cup B}(t|_p = l)$ of $E \cup B$-unifiers?

$E \cup B$-Unification

The notion of a $E \cup B$-unifier of a Σ-equation $u = v$ is as expected: it is a substitution θ such that $u\theta =_{E \cup B} v\theta$.

$E \cup B$-Unification

The notion of a $E \cup B$-unifier of a Σ-equation $u = v$ is as expected: it is a substitution θ such that $u\theta =_{E \cup B} v\theta$.

The notion of a complete set $Unif_{E \cup B}(u = v)$ of $E \cup B$-unifiers is also as expected: $Unif_{E \cup B}(u = v)$ is a set of $E \cup B$-unifiers of $u = v$ such that for any $E \cup B$-unifier α of $u = v$ there exists a unifier $\gamma \in Unif_{E \cup B}(u = v)$ of which α is an "instance modulo $E \cup B$."

$E \cup B$-Unification

The notion of a $E \cup B$-unifier of a Σ-equation $u = v$ is as expected: it is a substitution θ such that $u\theta =_{E \cup B} v\theta$.

The notion of a complete set $Unif_{E \cup B}(u = v)$ of $E \cup B$-unifiers is also as expected: $Unif_{E \cup B}(u = v)$ is a set of $E \cup B$-unifiers of $u = v$ such that for any $E \cup B$-unifier α of $u = v$ there exists a unifier $\gamma \in Unif_{E \cup B}(u = v)$ of which α is an "instance modulo $E \cup B$." That is, there is a substitution δ such that $\alpha =_{E \cup B} \gamma\delta$, where, by definition, given substitutions μ, ν
$\mu =_{E \cup B} \nu \;\Leftrightarrow_{def}\; (\forall x \in dom(\mu) \cup dom(\nu))\; \mu(x) =_{E \cup B} \nu(x)$.

$E \cup B$-Unification

The notion of a $E \cup B$-unifier of a Σ-equation $u = v$ is as expected: it is a substitution θ such that $u\theta =_{E \cup B} v\theta$.

The notion of a complete set $Unif_{E \cup B}(u = v)$ of $E \cup B$-unifiers is also as expected: $Unif_{E \cup B}(u = v)$ is a set of $E \cup B$-unifiers of $u = v$ such that for any $E \cup B$-unifier α of $u = v$ there exists a unifier $\gamma \in Unif_{E \cup B}(u = v)$ of which α is an "instance modulo $E \cup B$." That is, there is a substitution δ such that $\alpha =_{E \cup B} \gamma\delta$, where, by definition, given substitutions μ, ν
$\mu =_{E \cup B} \nu \;\; \Leftrightarrow_{def} \;\; (\forall x \in dom(\mu) \cup dom(\nu))\; \mu(x) =_{E \cup B} \nu(x)$.

For $E \cup B$ an arbitrary set of equations $E \cup B$, computing such a set $Unif_{E \cup B}(u = v)$ is a very complex matter.

$E \cup B$-Unification

The notion of a $E \cup B$-unifier of a Σ-equation $u = v$ is as expected: it is a substitution θ such that $u\theta =_{E\cup B} v\theta$.

The notion of a complete set $Unif_{E\cup B}(u = v)$ of $E \cup B$-unifiers is also as expected: $Unif_{E\cup B}(u = v)$ is a set of $E \cup B$-unifiers of $u = v$ such that for any $E \cup B$-unifier α of $u = v$ there exists a unifier $\gamma \in Unif_{E\cup B}(u = v)$ of which α is an "instance modulo $E \cup B$." That is, there is a substitution δ such that $\alpha =_{E\cup B} \gamma\delta$, where, by definition, given substitutions μ, ν
$\mu =_{E\cup B} \nu \;\Leftrightarrow_{def}\; (\forall x \in dom(\mu) \cup dom(\nu))\; \mu(x) =_{E\cup B} \nu(x)$.

For $E \cup B$ an arbitrary set of equations $E \cup B$, computing such a set $Unif_{E\cup B}(u = v)$ is a very complex matter. But for our purposes we may assume that the oriented equations $\vec{E}$ are convergent modulo B, which makes the task much easier.

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B

For $\vec{E}$ convergent modulo B, by the Church-Rosser Theorem, for any Σ-equation $u = v$ and substitution θ we have the equivalence:

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B

For $\vec{E}$ convergent modulo B, by the Church-Rosser Theorem, for any Σ-equation $u = v$ and substitution θ we have the equivalence:

$$(\dagger) \quad u\theta =_{E \cup B} v\theta \quad \Leftrightarrow \quad (u\theta)!_{\vec{E}/B} =_B (v\theta)!_{\vec{E}/B}$$

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B

For $\vec{E}$ convergent modulo B, by the Church-Rosser Theorem, for any Σ-equation $u = v$ and substitution θ we have the equivalence:

$$(\dagger) \quad u\theta =_{E \cup B} v\theta \quad \Leftrightarrow \quad (u\theta)!_{\vec{E}/B} =_B (v\theta)!_{\vec{E}/B}$$

This suggest the idea of computing $E \cup B$-unifiers by narrowing! using a theory transformation $(\Sigma, E \cup B) \mapsto (\Sigma^{\equiv}, E^{\equiv} \cup B)$, where:

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B

For $\vec{E}$ convergent modulo B, by the Church-Rosser Theorem, for any Σ-equation $u = v$ and substitution θ we have the equivalence:

$$(\dagger) \quad u\theta =_{E \cup B} v\theta \quad \Leftrightarrow \quad (u\theta)!_{\vec{E}/B} =_B (v\theta)!_{\vec{E}/B}$$

This suggest the idea of computing $E \cup B$-unifiers by narrowing! using a theory transformation $(\Sigma, E \cup B) \mapsto (\Sigma^{\equiv}, E^{\equiv} \cup B)$, where:

1. $\Sigma^{\equiv}$ extends Σ by adding: (a) for each connected component $[s]$ in Σ not having a top sort $\top_{[s]}$, such a new top sort $\top_{[s]}$; (b) a new sort *Pred* with a constant *tt*; and (c) for each connected component $[s]$ in Σ a binary equality predicate $\_ \equiv \_ : \top_{[s]}\ \top_{[s]} \to Pred$.

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B

For $\vec{E}$ convergent modulo B, by the Church-Rosser Theorem, for any Σ-equation $u = v$ and substitution θ we have the equivalence:

$$(\dagger) \quad u\theta =_{E \cup B} v\theta \quad \Leftrightarrow \quad (u\theta)!_{\vec{E}/B} =_B (v\theta)!_{\vec{E}/B}$$

This suggest the idea of computing $E \cup B$-unifiers by narrowing! using a theory transformation $(\Sigma, E \cup B) \mapsto (\Sigma^{\equiv}, E^{\equiv} \cup B)$, where:

1. $\Sigma^{\equiv}$ extends Σ by adding: (a) for each connected component $[s]$ in Σ not having a top sort $\top_{[s]}$, such a new top sort $\top_{[s]}$; (b) a new sort $Pred$ with a constant tt; and (c) for each connected component $[s]$ in Σ a binary equality predicate $\_ \equiv \_ : \top_{[s]}\ \top_{[s]} \to Pred$.

2. $E^{\equiv}$ extends E by adding for each connected component $[s]$ in Σ an equation $x{:}\top_{[s]} \equiv x{:}\top_{[s]} = tt$.

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B (II)

It is easy to check (exercise!) that if $\vec{E}$ is convergent modulo B, then $\vec{E}^{\equiv}$ is convergent modulo B. But then (†) becomes:

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B (II)

It is easy to check (exercise!) that if $\vec{E}$ is convergent modulo B, then $\vec{E}^{\equiv}$ is convergent modulo B. But then (†) becomes:

$$u\theta =_{E \cup B} v\theta \;\;\Leftrightarrow\;\; (u\theta \equiv v\theta)!_{\vec{E}^{\equiv}/B} = tt.$$

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B (II)

It is easy to check (exercise!) that if $\vec{E}$ is convergent modulo B, then $\vec{E}^{\equiv}$ is convergent modulo B. But then (†) becomes:

$$u\theta =_{E \cup B} v\theta \;\;\Leftrightarrow\;\; (u\theta \equiv v\theta)!_{\vec{E}^{\equiv}/B} = tt.$$

Indeed, by convergence, $(u\theta \equiv v\theta)!_{\vec{E}^{\equiv}/B} = tt$ iff we have:

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B (II)

It is easy to check (exercise!) that if $\vec{E}$ is convergent modulo B, then $\vec{E}^{\equiv}$ is convergent modulo B. But then (†) becomes:

$$u\theta =_{E \cup B} v\theta \;\; \Leftrightarrow \;\; (u\theta \equiv v\theta)!_{\vec{E}^{\equiv}/B} = tt.$$

Indeed, by convergence, $(u\theta \equiv v\theta)!_{\vec{E}^{\equiv}/B} = tt$ iff we have:

$$(\ddagger) \quad u\theta \equiv v\theta \to^{*}_{\vec{E}/B} (u\theta)!_{\vec{E}/B} \equiv (v\theta)!_{\vec{E}/B} \to_{\vec{E}^{\equiv}/B} tt$$

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B (II)

It is easy to check (exercise!) that if $\vec{E}$ is convergent modulo B, then $\vec{E}^{\equiv}$ is convergent modulo B. But then (†) becomes:

$$u\theta =_{E \cup B} v\theta \;\;\Leftrightarrow\;\; (u\theta \equiv v\theta)!_{\vec{E}^{\equiv}/B} = tt.$$

Indeed, by convergence, $(u\theta \equiv v\theta)!_{\vec{E}^{\equiv}/B} = tt$ iff we have:

$$(\ddagger) \quad u\theta \equiv v\theta \to^{*}_{\vec{E}/B} (u\theta)!_{\vec{E}/B} \equiv (v\theta)!_{\vec{E}/B} \to_{\vec{E}^{\equiv}/B} tt$$

with a rule $x{:}\top_{[s]} \equiv x{:}\top_{[s]} \to tt$ in $\vec{E}^{\equiv} \setminus \vec{E}$ used only in the last step to check $(u\theta)!_{\vec{E}/B} =_B (v\theta)!_{\vec{E}/B}$.

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B (II)

It is easy to check (exercise!) that if $\vec{E}$ is convergent modulo B, then $\vec{E}^{\equiv}$ is convergent modulo B. But then (†) becomes:

$$u\theta =_{E \cup B} v\theta \quad \Leftrightarrow \quad (u\theta \equiv v\theta)!_{\vec{E}^{\equiv}/B} = tt.$$

Indeed, by convergence, $(u\theta \equiv v\theta)!_{\vec{E}^{\equiv}/B} = tt$ iff we have:

$$(\ddagger) \quad u\theta \equiv v\theta \to^{*}_{\vec{E}/B} (u\theta)!_{\vec{E}/B} \equiv (v\theta)!_{\vec{E}/B} \to_{\vec{E}^{\equiv}/B} tt$$

with a rule $x{:}\top_{[s]} \equiv x{:}\top_{[s]} \to tt$ in $\vec{E}^{\equiv} \setminus \vec{E}$ used only in the last step to check $(u\theta)!_{\vec{E}/B} =_B (v\theta)!_{\vec{E}/B}$. Thus, by (†) we get:

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B (II)

It is easy to check (exercise!) that if $\vec{E}$ is convergent modulo B, then $\vec{E}^{\equiv}$ is convergent modulo B. But then (†) becomes:

$$u\theta =_{E \cup B} v\theta \quad \Leftrightarrow \quad (u\theta \equiv v\theta)!_{\vec{E}^{\equiv}/B} = tt.$$

Indeed, by convergence, $(u\theta \equiv v\theta)!_{\vec{E}^{\equiv}/B} = tt$ iff we have:

$$(\ddagger) \quad u\theta \equiv v\theta \rightarrow^{*}_{\vec{E}/B} (u\theta)!_{\vec{E}/B} \equiv (v\theta)!_{\vec{E}/B} \rightarrow_{\vec{E}^{\equiv}/B} tt$$

with a rule $x{:}\top_{[s]} \equiv x{:}\top_{[s]} \rightarrow tt$ in $\vec{E}^{\equiv} \setminus \vec{E}$ used only in the last step to check $(u\theta)!_{\vec{E}/B} =_B (v\theta)!_{\vec{E}/B}$. Thus, by (†) we get:

Theorem. θ is a $E \cup B$-unifier of $u = v$ iff $(u\theta \equiv v\theta)!_{\vec{E}^{\equiv}/B} = tt$.

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B (III)

This gives us our desired $E \cup B$-unification semi-algorithm, whose proof of correctness follows easily (exercise!) by repeated application of the Lifting Lemma for the rewrite theory $(\Sigma^{\equiv}, B, \vec{E}^{\equiv})$, just by observing that θ is a $E \cup B$-unifier of $u = v$ iff its $\vec{E}/B$-normalized form $\theta!_{\vec{E}/B}$ is so.

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B (III)

This gives us our desired $E \cup B$-unification semi-algorithm, whose proof of correctness follows easily (exercise!) by repeated application of the Lifting Lemma for the rewrite theory $(\Sigma^{\equiv}, B, \vec{E}^{\equiv})$, just by observing that θ is a $E \cup B$-unifier of $u = v$ iff its $\vec{E}/B$-normalized form $\theta!_{\vec{E}/B}$ is so.

Theorem. For $\vec{E}$ convergent modulo B and applied with B-extensions (see pg. 7 of Lecture 23), the set

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B (III)

This gives us our desired $E \cup B$-unification semi-algorithm, whose proof of correctness follows easily (exercise!) by repeated application of the Lifting Lemma for the rewrite theory $(\Sigma^{\equiv}, B, \vec{E}^{\equiv})$, just by observing that θ is a $E \cup B$-unifier of $u = v$ iff its $\vec{E}/B$-normalized form $\theta!_{\vec{E}/B}$ is so.

Theorem. For $\vec{E}$ convergent modulo B and applied with B-extensions (see pg. 7 of Lecture 23), the set

$$Unif_{E \cup B}(u = v) =_{def} \{\gamma \mid (u \equiv v) \overset{\gamma}{\rightsquigarrow}^{*}_{\vec{E}^{\equiv},B} tt\}$$

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B (III)

This gives us our desired $E \cup B$-unification semi-algorithm, whose proof of correctness follows easily (exercise!) by repeated application of the Lifting Lemma for the rewrite theory $(\Sigma^{\equiv}, B, \vec{E}^{\equiv})$, just by observing that θ is a $E \cup B$-unifier of $u = v$ iff its $\vec{E}/B$-normalized form $\theta!_{\vec{E}/B}$ is so.

Theorem. For $\vec{E}$ convergent modulo B and applied with B-extensions (see pg. 7 of Lecture 23), the set

$$Unif_{E \cup B}(u = v) =_{def} \{\gamma \mid (u \equiv v) \overset{\gamma}{\leadsto}{}^{*}_{\vec{E}^{\equiv},B}\ tt\}$$

is a complete set of $E \cup B$-unifiers of the equation $u = v$.

$E \cup B$-Unification for $\vec{E}$ Convergent Modulo B (III)

This gives us our desired $E \cup B$-unification semi-algorithm, whose proof of correctness follows easily (exercise!) by repeated application of the Lifting Lemma for the rewrite theory $(\Sigma^{\equiv}, B, \vec{E}^{\equiv})$, just by observing that θ is a $E \cup B$-unifier of $u = v$ iff its $\vec{E}/B$-normalized form $\theta!_{\vec{E}/B}$ is so.

Theorem. For $\vec{E}$ convergent modulo B and applied with B-extensions (see pg. 7 of Lecture 23), the set

$$Unif_{E \cup B}(u = v) =_{def} \{\gamma \mid (u \equiv v) \overset{\gamma}{\leadsto}{}^{*}_{\vec{E}^{\equiv},B}\ tt\}$$

is a complete set of $E \cup B$-unifiers of the equation $u = v$.

For narrowing-based model checking, we obtain as an immediate corollary the following vast generalization of the Completeness of Narrowing Search Theorem in Lecture 23 for topmost theories:

Symbolic Model Checking of Topmost Rewrite Theories

For $\mathcal{R} = (\Sigma, E \cup B, R)$ topmost, narrowing with R modulo axioms $E \cup B$ supports the following symbolic model checking method:

Symbolic Model Checking of Topmost Rewrite Theories

For $\mathcal{R} = (\Sigma, E \cup B, R)$ topmost, narrowing with R modulo axioms $E \cup B$ supports the following symbolic model checking method:

Theorem (Completeness of Narrowing Search). For a topmost and admissible $\mathcal{R} = (\Sigma, E \cup B, R)$ with $\vec{E}$ convergent modulo B and $u_1 \vee \ldots \vee u_n$ and $v_1 \vee \ldots \vee v_m$ Σ-pattern disjunctions,

Symbolic Model Checking of Topmost Rewrite Theories

For $\mathcal{R} = (\Sigma, E \cup B, R)$ topmost, narrowing with R modulo axioms $E \cup B$ supports the following symbolic model checking method:

Theorem (Completeness of Narrowing Search). For a topmost and admissible $\mathcal{R} = (\Sigma, E \cup B, R)$ with $\vec{E}$ convergent modulo B and $u_1 \vee \ldots \vee u_n$ and $v_1 \vee \ldots \vee v_m$ Σ-pattern disjunctions,

$$\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$$

Symbolic Model Checking of Topmost Rewrite Theories

For $\mathcal{R} = (\Sigma, E \cup B, R)$ topmost, narrowing with R modulo axioms $E \cup B$ supports the following symbolic model checking method:

Theorem (Completeness of Narrowing Search). For a topmost and admissible $\mathcal{R} = (\Sigma, E \cup B, R)$ with $\vec{E}$ convergent modulo B and $u_1 \vee \ldots \vee u_n$ and $v_1 \vee \ldots \vee v_m$ Σ-pattern disjunctions,

$$\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$$

holds iff

Symbolic Model Checking of Topmost Rewrite Theories

For $\mathcal{R} = (\Sigma, E \cup B, R)$ topmost, narrowing with R modulo axioms $E \cup B$ supports the following symbolic model checking method:

Theorem (Completeness of Narrowing Search). For a topmost and admissible $\mathcal{R} = (\Sigma, E \cup B, R)$ with $\vec{E}$ convergent modulo B and $u_1 \vee \ldots \vee u_n$ and $v_1 \vee \ldots \vee v_m$ Σ-pattern disjunctions,

$$\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$$

holds iff exist i, j, $1 \leq i \leq n$, $1 \leq j \leq m$, and a narrowing sequence $u_i \overset{\theta}{\leadsto}^{*}_{R,(E \cup B)} w$ such that $\mathit{Unif}_{E \cup B}(w = v_j) \neq \emptyset$.

Symbolic Model Checking of Topmost Rewrite Theories

For $\mathcal{R} = (\Sigma, E \cup B, R)$ topmost, narrowing with R modulo axioms $E \cup B$ supports the following symbolic model checking method:

Theorem (Completeness of Narrowing Search). For a topmost and admissible $\mathcal{R} = (\Sigma, E \cup B, R)$ with $\vec{E}$ convergent modulo B and $u_1 \vee \ldots \vee u_n$ and $v_1 \vee \ldots \vee v_m$ Σ-pattern disjunctions,

$$\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$$

holds iff exist i, j, $1 \leq i \leq n$, $1 \leq j \leq m$, and a narrowing sequence $u_i \overset{\theta}{\rightsquigarrow}^{*}_{R,(E \cup B)} w$ such that $Unif_{E \cup B}(w = v_j) \neq \emptyset$.

The proof, by applying the Lifting Lemma, generalizes the similar proof in Lecture 23 and is left as an exercise.

Performance Barriers for Symbolic Reachability

In the above, generalized **Completeness of Narrowing Search Theorem**, narrowing happens at two levels: (i) with R modulo $E \cup B$, i.e., $\rightsquigarrow^*_{R,(E\cup B)}$, for reachability analysis, and (ii) with $\vec{E}^{\equiv}$ modulo B, i.e., $\rightsquigarrow^*_{\vec{E}^{\equiv},B}$, for computing $E \cup B$-unifiers.

Performance Barriers for Symbolic Reachability

In the above, generalized **Completeness of Narrowing Search Theorem**, narrowing happens at two levels: (i) with R modulo $E \cup B$, i.e., $\rightsquigarrow^*_{R,(E \cup B)}$, for reachability analysis, and (ii) with $\vec{E}^{\equiv}$ modulo B, i.e., $\rightsquigarrow^*_{\vec{E}^{\equiv},B}$, for computing $E \cup B$-unifiers.

From a performance point of view this is very challenging, since this gives us what we might describe as a "nested narrowing tree," wich can by infinite at both of the narrowing levels.

Performance Barriers for Symbolic Reachability

In the above, generalized **Completeness of Narrowing Search Theorem**, narrowing happens at two levels: (i) with R modulo $E \cup B$, i.e., $\rightsquigarrow^*_{R,(E \cup B)}$, for reachability analysis, and (ii) with $\vec{E}^{\equiv}$ modulo B, i.e., $\rightsquigarrow^*_{\vec{E}^{\equiv},B}$, for computing $E \cup B$-unifiers.

From a performance point of view this is very challenging, since this gives us what we might describe as a "nested narrowing tree," wich can by infinite at both of the narrowing levels.

To overcome these performance barriers, the technique of folding an infinite narrowing tree into a (hopefully finite) narrowing graph can be applied at both levels.

Performance Barriers for Symbolic Reachability

In the above, generalized **Completeness of Narrowing Search Theorem**, narrowing happens at two levels: (i) with R modulo $E \cup B$, i.e., $\rightsquigarrow^*_{R,(E \cup B)}$, for reachability analysis, and (ii) with $\vec{E}^{\equiv}$ modulo B, i.e., $\rightsquigarrow^*_{\vec{E}^{\equiv},B}$, for computing $E \cup B$-unifiers.

From a performance point of view this is very challenging, since this gives us what we might describe as a "nested narrowing tree," wich can by infinite at both of the narrowing levels.

To overcome these performance barriers, the technique of folding an infinite narrowing tree into a (hopefully finite) narrowing graph can be applied at both levels. For the symbolic reachability level with $\rightsquigarrow^*_{R,B}$ we have already seen this in Lecture 24.

Performance Barriers for Symbolic Reachability

In the above, generalized **Completeness of Narrowing Search Theorem**, narrowing happens at two levels: (i) with R modulo $E \cup B$, i.e., $\rightsquigarrow^{*}_{R,(E \cup B)}$, for reachability analysis, and (ii) with $\vec{E}^{\equiv}$ modulo B, i.e., $\rightsquigarrow^{*}_{\vec{E}^{\equiv},B}$, for computing $E \cup B$-unifiers.

From a performance point of view this is very challenging, since this gives us what we might describe as a "nested narrowing tree," wich can by infinite at both of the narrowing levels.

To overcome these performance barriers, the technique of folding an infinite narrowing tree into a (hopefully finite) narrowing graph can be applied at both levels. For the symbolic reachability level with $\rightsquigarrow^{*}_{R,B}$ we have already seen this in Lecture 24. Likewise, for $\vec{E}, B$-narrowing with $\vec{E}$ convergent modulo B ($\vec{E}^{\equiv}, B$-narrowing is just a special case), folding variant narrowing delivers the goods:

Folding Variant Narrowing

Folding Variant Narrowing, proposed by S. Escobar, R. Sasse and J. Meseguer[1] for theories $(\Sigma, E \cup B)$ with $\vec{E}$ convergent modulo B, folds the $\vec{E}, B$-narrowing tree of t into a graph in a breadth first manner as follows:

[1] "Folding variant narrowing and optimal variant termination", J. Alg. & Log. Prog., 81, 898–928, 2012.

Folding Variant Narrowing

Folding Variant Narrowing, proposed by S. Escobar, R. Sasse and J. Meseguer[1] for theories $(\Sigma, E \cup B)$ with $\vec{E}$ convergent modulo B, folds the $\vec{E}, B$-narrowing tree of t into a graph in a breadth first manner as follows:

1. It considers only paths $t \overset{\theta}{\leadsto}{}^{n}_{\vec{E},B} u$ in the narrowing tree such that u and θ are $\vec{E}, B$-normalized.

[1] "Folding variant narrowing and optimal variant termination", J. Alg. & Log. Prog., 81, 898–928, 2012.

Folding Variant Narrowing

Folding Variant Narrowing, proposed by S. Escobar, R. Sasse and J. Meseguer[1] for theories $(\Sigma, E \cup B)$ with $\vec{E}$ convergent modulo B, folds the $\vec{E}, B$-narrowing tree of t into a graph in a breadth first manner as follows:

1. It considers only paths $t \overset{\theta}{\leadsto}{}^{n}_{\vec{E},B}\ u$ in the narrowing tree such that u and θ are $\vec{E}, B$-normalized.
2. For any such path $t \overset{\theta}{\leadsto}{}^{n}_{\vec{E},B}\ u$, if there is another such different path $t \overset{\theta'}{\leadsto}{}^{m}_{\vec{E},B}\ u'$ with $m \leq n$ and a B-matching substitution γ such that: (i) $u =_B u'\gamma$, and (ii) $\theta =_B \theta'\gamma$, then the node u is folded into the more general node u'.

[1] "Folding variant narrowing and optimal variant termination", J. Alg. & Log. Prog., 81, 898–928, 2012.

Folding Variant Narrowing (II)

The pairs (u, θ) associated to paths $t \overset{\theta}{\rightsquigarrow}^{n}_{\vec{E},B} u$ in such a graph are called the $\vec{E}, B$-variants of t; and the graph thus obtained is called the folding variant narrowing graph of t.

Folding Variant Narrowing (II)

The pairs (u, θ) associated to paths $t \overset{\theta}{\rightsquigarrow}^{n}_{\vec{E},B} u$ in such a graph are called the $\vec{E}, B$-variants of t; and the graph thus obtained is called the folding variant narrowing graph of t.

Maude supports the enumeration of all variants in the folding variant narrowing graph of t by the `get variants` t `.` command (§14.4, Maude Manual).

Folding Variant Narrowing (II)

The pairs (u, θ) associated to paths $t \overset{\theta}{\leadsto}^{n}_{\vec{E},B} u$ in such a graph are called the $\vec{E}, B$-variants of t; and the graph thus obtained is called the folding variant narrowing graph of t.

Maude supports the enumeration of all variants in the folding variant narrowing graph of t by the `get variants` t `.` command (§14.4, Maude Manual). It also supports variant-based $E \cup B$-unification when $\vec{E}$ is convergent modulo B with the `variant unify` command (§14.9, Maude Manual).

Folding Variant Narrowing (II)

The pairs (u, θ) associated to paths $t \overset{\theta}{\rightsquigarrow}^{n}_{\vec{E},B} u$ in such a graph are called the $\vec{E}, B$-variants of t; and the graph thus obtained is called the folding variant narrowing graph of t.

Maude supports the enumeration of all variants in the folding variant narrowing graph of t by the `get variants` t `.` command (§14.4, Maude Manual). It also supports variant-based $E \cup B$-unification when $\vec{E}$ is convergent modulo B with the `variant unify` command (§14.9, Maude Manual).

$(\Sigma, E \cup B)$ enjoys the finite variant property (FVP) iff for any Σ-term t its folding variant graph is finite.

Folding Variant Narrowing (II)

The pairs (u, θ) associated to paths $t \leadsto^{\theta\, n}_{\vec{E},B} u$ in such a graph are called the $\vec{E}, B$-variants of t; and the graph thus obtained is called the folding variant narrowing graph of t.

Maude supports the enumeration of all variants in the folding variant narrowing graph of t by the `get variants` t `.` command (§14.4, Maude Manual). It also supports variant-based $E \cup B$-unification when $\vec{E}$ is convergent modulo B with the `variant unify` command (§14.9, Maude Manual).

$(\Sigma, E \cup B)$ enjoys the finite variant property (FVP) iff for any Σ-term t its folding variant graph is finite. This property holds iff for each $f : s_1 \ldots s_n \to s$ in Σ the folding variant graph of $f(x_1 : s_1, \ldots, x_n : s_n)$ is finite, which can be checked in Maude.

An FVP Example: SET

In the theory $(\Sigma, E \cup AC)$ SET below we can preform AC-unification in Maude as follows:

An FVP Example: SET

In the theory $(\Sigma, E \cup AC)$ SET below we can preform AC-unification in Maude as follows:

```
fmod SET is
sort Set .
ops mt a b c d e f g : -> Set [ctor] .
op _U_ : Set Set -> Set [ctor assoc comm] . *** union
vars S S' : Set .
eq S U mt = S [variant] .          *** identity
eq S U S = S [variant] .           *** idempotencu
eq S U S U S' = S U S' [variant] . *** idempotency extension
endfm

unify a U a U b U S =? a U c U S' .

Unifier 1
S --> c U #1:Set
S' --> a U b U #1:Set

Unifier 2
S --> c
S' --> a U b
```

An FVP Example: SET (II)

SET is FVP because S U S' has a finite number of variants:

An FVP Example: SET (II)

SET is FVP because `S U S'` has a finite number of variants:

```
get variants S U S' .

Variant 1
Set: #1:Set U #2:Set
S --> #1:Set
S' --> #2:Set

Variant 2
Set: %1:Set
S --> mt
S' --> %1:Set

Variant 3
Set: %1:Set
S --> %1:Set
S' --> mt

Variant 4
Set: %1:Set
S --> %1:Set
S' --> %1:Set
```

An FVP Example: SET (III)

```
Variant 5
Set: %1:Set U %2:Set U %3:Set
S --> %1:Set U %2:Set
S' --> %1:Set U %3:Set

Variant 6
Set: %1:Set U %2:Set
S --> %1:Set U %2:Set
S' --> %2:Set

Variant 7
Set: %1:Set U %2:Set
S --> %2:Set
S' --> %1:Set U %2:Set

No more variants.
```

Variant Unification for FVP Theories

It is easy to check (exercise!) that if $(\Sigma, E \cup B)$ is FVP, then $(\Sigma^{\equiv}, E^{\equiv} \cup B)$ is also FVP. This means that, when $(\Sigma, E \cup B)$ is FVP, variant unification always provides a finite and complete set of $E \cup B$-unifiers. For example, since SET is FVP any $E \cup AC$-unification problem has a finite number of variant unifiers.

Variant Unification for FVP Theories

It is easy to check (exercise!) that if $(\Sigma, E \cup B)$ is FVP, then $(\Sigma^{\equiv}, E^{\equiv} \cup B)$ is also FVP. This means that, when $(\Sigma, E \cup B)$ is FVP, variant unification always provides a finite and complete set of $E \cup B$-unifiers. For example, since SET is FVP any $E \cup AC$-unification problem has a finite number of variant unifiers.

```
filtered variant unify a U a U b U S =? a U c U S' .

Unifier 1
S --> c U %1:Set
S' --> b U %1:Set

Unifier 2
S --> a U c U #1:Set
S' --> b U #1:Set

Unifier 3
S --> c U #1:Set
S' --> a U b U #1:Set

No more unifiers.
```

Symbolic Model Checking for $\mathcal{R} = (\Sigma, E \cup B, R)$ when $E \cup B$ is FVP

Thus, for $(\Sigma, E \cup B)$ FVP, the Completeness of Narrowing Search Theorem for a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ of pg. 8 makes symbolic model checking tractable. It is supported by the same `{fold} vu-narrow` command already discussed in Lectures 23-24.

Symbolic Model Checking for $\mathcal{R} = (\Sigma, E \cup B, R)$ when $E \cup B$ is FVP

Thus, for $(\Sigma, E \cup B)$ FVP, the Completeness of Narrowing Search Theorem for a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ of pg. 8 makes symbolic model checking tractable. It is supported by the same {`fold`} `vu-narrow` command already discussed in Lectures 23-24.

In summary, we have generalized the symbolic model checking results from Lecture 24 to:

Symbolic Model Checking for $\mathcal{R} = (\Sigma, E \cup B, R)$ when $E \cup B$ is FVP

Thus, for $(\Sigma, E \cup B)$ FVP, the Completeness of Narrowing Search Theorem for a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ of pg. 8 makes symbolic model checking tractable. It is supported by the same `{fold} vu-narrow` command already discussed in Lectures 23-24.

In summary, we have generalized the symbolic model checking results from Lecture 24 to: (i) any topmost rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ with $\vec{E}$ convergent modulo B, and

Symbolic Model Checking for $\mathcal{R} = (\Sigma, E \cup B, R)$ when $E \cup B$ is FVP

Thus, for $(\Sigma, E \cup B)$ FVP, the Completeness of Narrowing Search Theorem for a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ of pg. 8 makes symbolic model checking tractable. It is supported by the same {`fold`} `vu-narrow` command already discussed in Lectures 23-24.

In summary, we have generalized the symbolic model checking results from Lecture 24 to: (i) any topmost rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ with $\vec{E}$ convergent modulo B, and (ii) made it tractable when $E \cup B$ is FVP.

Symbolic Model Checking for $\mathcal{R} = (\Sigma, E \cup B, R)$ when $E \cup B$ is FVP

Thus, for $(\Sigma, E \cup B)$ FVP, the Completeness of Narrowing Search Theorem for a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ of pg. 8 makes symbolic model checking tractable. It is supported by the same `{fold} vu-narrow` command already discussed in Lectures 23-24.

In summary, we have generalized the symbolic model checking results from Lecture 24 to: (i) any topmost rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ with $\vec{E}$ convergent modulo B, and (ii) made it tractable when $E \cup B$ is FVP. For symbolic model checking examples when $E \cup B$ is FVP, see §15 of the The Maude Manual. Further examples will be given in future Lectures.

Completeness of Folding Narrowing Search

Theorem (Completeness of Folding Narrowing Search). For a topmost and admissible $\mathcal{R} = (\Sigma, E \cup B, R)$ with $E \cup B$ FVP, and $u_1 \vee \ldots \vee u_n$ and $v_1 \vee \ldots \vee v_m$ Σ-pattern disjunctions,

$$\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$$

holds iff

Completeness of Folding Narrowing Search

Theorem (Completeness of Folding Narrowing Search). For a topmost and admissible $\mathcal{R} = (\Sigma, E \cup B, R)$ with $E \cup B$ FVP, and $u_1 \vee \ldots \vee u_n$ and $v_1 \vee \ldots \vee v_m$ Σ-pattern disjunctions,

$$\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$$

holds iff there exists a $d \in \mathbb{N}$ and some j, $1 \leq j \leq m$, such that $P_d \wedge v_j \neq \bot$,

Completeness of Folding Narrowing Search

Theorem (Completeness of Folding Narrowing Search). For a topmost and admissible $\mathcal{R} = (\Sigma, E \cup B, R)$ with $E \cup B$ FVP, and $u_1 \vee \ldots \vee u_n$ and $v_1 \vee \ldots \vee v_m$ Σ-pattern disjunctions,

$$\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$$

holds iff there exists a $d \in \mathbb{N}$ and some j, $1 \leq j \leq m$, such that $P_d \wedge v_j \neq \bot$, where $P_d \wedge v_j$ is computed by $E \cup B$-unification.

Completeness of Folding Narrowing Search

Theorem (Completeness of Folding Narrowing Search). For a topmost and admissible $\mathcal{R} = (\Sigma, E \cup B, R)$ with $E \cup B$ FVP, and $u_1 \vee \ldots \vee u_n$ and $v_1 \vee \ldots \vee v_m$ Σ-pattern disjunctions,

$$\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$$

holds iff there exists a $d \in \mathbb{N}$ and some j, $1 \leq j \leq m$, such that $P_d \wedge v_j \neq \bot$, where $P_d \wedge v_j$ is computed by $E \cup B$-unification.

Proof (Sketch): This follows from the **Completeness of Narrowing Search Theorem** in pg. 8, and the **Completeness Theorem of Folding Narrowing** in pg. 12 of Lecture 24,

Completeness of Folding Narrowing Search

Theorem (Completeness of Folding Narrowing Search). For a topmost and admissible $\mathcal{R} = (\Sigma, E \cup B, R)$ with $E \cup B$ FVP, and $u_1 \vee \ldots \vee u_n$ and $v_1 \vee \ldots \vee v_m$ Σ-pattern disjunctions,

$$\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$$

holds iff there exists a $d \in \mathbb{N}$ and some j, $1 \leq j \leq m$, such that $P_d \wedge v_j \neq \bot$, where $P_d \wedge v_j$ is computed by $E \cup B$-unification.

Proof (Sketch): This follows from the **Completeness of Narrowing Search Theorem** in pg. 8, and the **Completeness Theorem of Folding Narrowing** in pg. 12 of Lecture 24, because, since $E \cup B$ is FVP, $Unif_{E \cup B}(u = v)$ is always a finite set for any Σ-equation $u = v$.

Completeness of Folding Narrowing Search

Theorem (Completeness of Folding Narrowing Search). For a topmost and admissible $\mathcal{R} = (\Sigma, E \cup B, R)$ with $E \cup B$ FVP, and $u_1 \vee \ldots \vee u_n$ and $v_1 \vee \ldots \vee v_m$ Σ-pattern disjunctions,

$$\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$$

holds iff there exists a $d \in \mathbb{N}$ and some j, $1 \leq j \leq m$, such that $P_d \wedge v_j \neq \bot$, where $P_d \wedge v_j$ is computed by $E \cup B$-unification.

Proof (Sketch): This follows from the **Completeness of Narrowing Search Theorem** in pg. 8, and the **Completeness Theorem of Folding Narrowing** in pg. 12 of Lecture 24, because, since $E \cup B$ is FVP, $Unif_{E \cup B}(u = v)$ is always a finite set for any Σ-equation $u = v$. Therefore, the Σ-pattern disjunctions P_d and F_d, $d \in \mathbb{N}$, exist and can be effectively computed according to the **Folding Narrowing Search Algorithm** in Lecture 24, by just generalizing Ω to Σ and $Unif_B(u = v)$ to $Unif_{E \cup B}(u = v)$. $\Box$