Program Verification: Lecture 26

José Meseguer

University of Illinois at Urbana-Champaign

Equational Abstractions

An equational abstraction of a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ is another rewrite theory $\mathcal{R}/G = (\Sigma, E \cup B \cup G, R)$, where G a set of Σ-equations.

Equational Abstractions

An equational abstraction of a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ is another rewrite theory $\mathcal{R}/G = (\Sigma, E \cup B \cup G, R)$, where G a set of Σ-equations. Equational abstractions can be very useful for both symbolic and explicit state model checking.

Equational Abstractions

An equational abstraction of a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ is another rewrite theory $\mathcal{R}/G = (\Sigma, E \cup B \cup G, R)$, where G a set of Σ-equations. Equational abstractions can be very useful for both symbolic and explicit state model checking.

This is because, some properties that may be hard to model check in $\mathcal{R}$ may be model checked in $\mathcal{R}/G$ with the guarantee that if they hold in $\mathcal{R}/G$ they also hold in $\mathcal{R}$.

Equational Abstractions

An equational abstraction of a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ is another rewrite theory $\mathcal{R}/G = (\Sigma, E \cup B \cup G, R)$, where G a set of Σ-equations. Equational abstractions can be very useful for both symbolic and explicit state model checking.

This is because, some properties that may be hard to model check in $\mathcal{R}$ may be model checked in $\mathcal{R}/G$ with the guarantee that if they hold in $\mathcal{R}/G$ they also hold in $\mathcal{R}$.

Even if $\mathcal{R}$ is admissible, $\mathcal{R}/G$ may not be so.

Equational Abstractions

An equational abstraction of a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ is another rewrite theory $\mathcal{R}/G = (\Sigma, E \cup B \cup G, R)$, where G a set of Σ-equations. Equational abstractions can be very useful for both symbolic and explicit state model checking.

This is because, some properties that may be hard to model check in $\mathcal{R}$ may be model checked in $\mathcal{R}/G$ with the guarantee that if they hold in $\mathcal{R}/G$ they also hold in $\mathcal{R}$.

Even if $\mathcal{R}$ is admissible, $\mathcal{R}/G$ may not be so. But we can always reason on the Σ transition systems $\mathbb{T}_{\mathcal{R}} = (\mathbb{T}_{\Sigma/E \cup B}, \to_{R/E \cup B})$ and $\mathbb{T}_{\mathcal{R}/G} = (T_{\Sigma/E \cup B \cup G}, \to_{R/E \cup B \cup G})$.

Equational Abstractions

An equational abstraction of a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ is another rewrite theory $\mathcal{R}/G = (\Sigma, E \cup B \cup G, R)$, where G a set of Σ-equations. Equational abstractions can be very useful for both symbolic and explicit state model checking.

This is because, some properties that may be hard to model check in $\mathcal{R}$ may be model checked in $\mathcal{R}/G$ with the guarantee that if they hold in $\mathcal{R}/G$ they also hold in $\mathcal{R}$.

Even if $\mathcal{R}$ is admissible, $\mathcal{R}/G$ may not be so. But we can always reason on the Σ transition systems $\mathbb{T}_{\mathcal{R}} = (\mathbb{T}_{\Sigma/E \cup B}, \to_{R/E \cup B})$ and $\mathbb{T}_{\mathcal{R}/G} = (T_{\Sigma/E \cup B \cup G}, \to_{R/E \cup B \cup G})$.

Ex.26.1 Prove that if $\mathcal{R}$ is admissible, the unique Σ-isomorphism $[\_!_{\vec{E}/B}] : \mathbb{T}_{\Sigma/E \cup B} \to \mathbb{C}_{\Sigma/\vec{E} \cup B}$ defines an isomorphism of Σ-transition systems. I.e., prove that for any Σ-terms u, v we have $[u]_{E \cup B} \to_{R/E \cup B} [u]_{E \cup B}$ in $\mathbb{T}_{\mathcal{R}}$ iff $[u!_{\vec{E}/B}]_B \to_{\mathcal{R}} [v!_{\vec{E}/B}]_B$ in $\mathbb{C}_{\mathcal{R}}$.

The Kripke Structures $\mathbb{T}_{\mathcal{R}}$ and $\mathbb{T}_{\mathcal{R}/G}$

Choosing a top sort *State* of states in Σ, we can define Kripke structures $\mathbb{T}_{\mathcal{R}} = (T_{\Sigma/E\cup B,State}, \rightarrow_{R/E\cup B}, \models_{\mathbb{T}_{\mathcal{R}}})$ and $\mathbb{T}_{\mathcal{R}/G} = (T_{\Sigma/E\cup B\cup G,State}, \rightarrow_{R/E\cup B\cup G}, \models_{\mathbb{T}_{\mathcal{R}/G}})$,

The Kripke Structures $\mathbb{T}_{\mathcal{R}}$ and $\mathbb{T}_{\mathcal{R}/G}$

Choosing a top sort *State* of states in Σ, we can define Kripke structures $\mathbb{T}_{\mathcal{R}} = (T_{\Sigma/E \cup B, State}, \to_{R/E \cup B}, \models_{\mathbb{T}_{\mathcal{R}}})$ and $\mathbb{T}_{\mathcal{R}/G} = (T_{\Sigma/E \cup B \cup G, State}, \to_{R/E \cup B \cup G}, \models_{\mathbb{T}_{\mathcal{R}/G}})$, where the set Π of state predicates is the set $T_{\Sigma}(X)_{State}$ with X an infinite set of variables, and

The Kripke Structures $\mathbb{T}_{\mathcal{R}}$ and $\mathbb{T}_{\mathcal{R}/G}$

Choosing a top sort *State* of states in Σ, we can define Kripke structures $\mathbb{T}_{\mathcal{R}} = (T_{\Sigma/E \cup B, State}, \rightarrow_{R/E \cup B}, \models_{\mathbb{T}_{\mathcal{R}}})$ and $\mathbb{T}_{\mathcal{R}/G} = (T_{\Sigma/E \cup B \cup G, State}, \rightarrow_{R/E \cup B \cup G}, \models_{\mathbb{T}_{\mathcal{R}/G}})$, where the set Π of state predicates is the set $T_{\Sigma}(X)_{State}$ with X an infinite set of variables, and its interpretation in $\mathbb{T}_{\mathcal{R}}$ (resp. $\mathbb{T}_{\mathcal{R}/G}$) is given by:

The Kripke Structures $\mathbb{T}_{\mathcal{R}}$ and $\mathbb{T}_{\mathcal{R}/G}$

Choosing a top sort *State* of states in Σ, we can define Kripke structures $\mathbb{T}_{\mathcal{R}} = (T_{\Sigma/E \cup B, State}, \rightarrow_{R/E \cup B}, \models_{\mathbb{T}_{\mathcal{R}}})$ and $\mathbb{T}_{\mathcal{R}/G} = (T_{\Sigma/E \cup B \cup G, State}, \rightarrow_{R/E \cup B \cup G}, \models_{\mathbb{T}_{\mathcal{R}/G}})$, where the set Π of state predicates is the set $T_\Sigma(X)_{State}$ with X an infinite set of variables, and its interpretation in $\mathbb{T}_{\mathcal{R}}$ (resp. $\mathbb{T}_{\mathcal{R}/G}$) is given by:

$$u_{\mathbb{T}_{\mathcal{R}}} = [\![u]\!]_{E \cup B} =_{def} \{[u\theta]_{E \cup B} \mid \theta \in [X \rightarrow T_\Sigma]\}$$

resp.

The Kripke Structures $\mathbb{T}_{\mathcal{R}}$ and $\mathbb{T}_{\mathcal{R}/G}$

Choosing a top sort *State* of states in Σ, we can define Kripke structures $\mathbb{T}_{\mathcal{R}} = (T_{\Sigma/E\cup B,State}, \rightarrow_{R/E\cup B}, \models_{\mathbb{T}_{\mathcal{R}}})$ and $\mathbb{T}_{\mathcal{R}/G} = (T_{\Sigma/E\cup B\cup G,State}, \rightarrow_{R/E\cup B\cup G}, \models_{\mathbb{T}_{\mathcal{R}/G}})$, where the set Π of state predicates is the set $T_{\Sigma}(X)_{State}$ with X an infinite set of variables, and its interpretation in $\mathbb{T}_{\mathcal{R}}$ (resp. $\mathbb{T}_{\mathcal{R}/G}$) is given by:

$$u_{\mathbb{T}_{\mathcal{R}}} = [\![u]\!]_{E\cup B} =_{def} \{[u\theta]_{E\cup B} \mid \theta \in [X \to T_{\Sigma}]\}$$

resp.

$$u_{\mathbb{T}_{\mathcal{R}/G}} = [\![u]\!]_{E\cup B\cup G} =_{def} \{[u\theta]_{E\cup B\cup G} \mid \theta \in [X \to T_{\Sigma}]\}$$

The Kripke Structures $\mathbb{T}_{\mathcal{R}}$ and $\mathbb{T}_{\mathcal{R}/G}$

Choosing a top sort *State* of states in Σ, we can define Kripke structures $\mathbb{T}_{\mathcal{R}} = (T_{\Sigma/E\cup B,State}, \rightarrow_{R/E\cup B}, \models_{\mathbb{T}_{\mathcal{R}}})$ and $\mathbb{T}_{\mathcal{R}/G} = (T_{\Sigma/E\cup B\cup G,State}, \rightarrow_{R/E\cup B\cup G}, \models_{\mathbb{T}_{\mathcal{R}/G}})$, where the set Π of state predicates is the set $T_{\Sigma}(X)_{State}$ with X an infinite set of variables, and its interpretation in $\mathbb{T}_{\mathcal{R}}$ (resp. $\mathbb{T}_{\mathcal{R}/G}$) is given by:

$$u_{\mathbb{T}_{\mathcal{R}}} = [\![u]\!]_{E\cup B} =_{def} \{[u\theta]_{E\cup B} \mid \theta \in [X \rightarrow T_{\Sigma}]\}$$

resp.

$$u_{\mathbb{T}_{\mathcal{R}/G}} = [\![u]\!]_{E\cup B\cup G} =_{def} \{[u\theta]_{E\cup B\cup G} \mid \theta \in [X \rightarrow T_{\Sigma}]\}$$

One reason why equational abstractions are so useful is summarized by the following theorem, whose easy proof is given in the Appendix.

Main Theorem About Equational Abstractions

Theorem. For $\mathcal{R}/G$ an equational abstraction of $\mathcal{R}$ and any state predicates $u_1, \ldots, u_n, v_1, \ldots, v_m \in T_\Sigma(X)_{State}$ the following holds:

Main Theorem About Equational Abstractions

Theorem. For $\mathcal{R}/G$ an equational abstraction of $\mathcal{R}$ and any state predicates $u_1, \ldots, u_n, v_1, \ldots, v_m \in T_\Sigma(X)_{State}$ the following holds:

$$\mathbb{T}_\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \diamond(v_1 \vee \ldots \vee v_m) \;\; \Rightarrow \;\; \mathbb{T}_{\mathcal{R}/G}, (u_1 \vee \ldots \vee u_n) \models_{S4} \diamond(v_1 \vee \ldots \vee v_m)$$

Main Theorem About Equational Abstractions

Theorem. For $\mathcal{R}/G$ an equational abstraction of $\mathcal{R}$ and any state predicates $u_1, \ldots, u_n, v_1, \ldots, v_m \in T_\Sigma(X)_{State}$ the following holds:

$$\mathbb{T}_\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m) \;\Rightarrow\; \mathbb{T}_{\mathcal{R}/G}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$$

and therefore the dual, contrapositive implication also holds:

Main Theorem About Equational Abstractions

Theorem. For $\mathcal{R}/G$ an equational abstraction of $\mathcal{R}$ and any state predicates $u_1, \ldots, u_n, v_1, \ldots, v_m \in T_\Sigma(X)_{State}$ the following holds:

$$\mathbb{T}_\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \diamond(v_1 \vee \ldots \vee v_m) \;\;\Rightarrow\;\; \mathbb{T}_{\mathcal{R}/G}, (u_1 \vee \ldots \vee u_n) \models_{S4} \diamond(v_1 \vee \ldots \vee v_m)$$

and therefore the dual, contrapositive implication also holds:

$$\mathbb{T}_{\mathcal{R}/G}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Box(v_1 \vee \ldots \vee v_m)^c \;\;\Rightarrow\;\; \mathbb{T}_\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Box(v_1 \vee \ldots \vee v_m)^c$$

Main Theorem About Equational Abstractions

Theorem. For $\mathcal{R}/G$ an equational abstraction of $\mathcal{R}$ and any state predicates $u_1, \ldots, u_n, v_1, \ldots, v_m \in T_\Sigma(X)_{State}$ the following holds:

$$\mathbb{T}_\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m) \;\; \Rightarrow \;\; \mathbb{T}_{\mathcal{R}/G}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$$

and therefore the dual, contrapositive implication also holds:

$$\mathbb{T}_{\mathcal{R}/G}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Box(v_1 \vee \ldots \vee v_m)^c \;\; \Rightarrow \;\; \mathbb{T}_\mathcal{R}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Box(v_1 \vee \ldots \vee v_m)^c$$

where, by definition,

$$[\![(v_1 \vee \ldots \vee v_m)^c]\!]_{E \cup B} =_{def} T_{\Sigma/E \cup B, State} \setminus [\![(v_1 \vee \ldots \vee v_m)]\!]_{E \cup B}$$

Main Theorem About Equational Abstractions

Theorem. For $\mathcal{R}/G$ an equational abstraction of $\mathcal{R}$ and any state predicates $u_1, \ldots, u_n, v_1, \ldots, v_m \in T_\Sigma(X)_{State}$ the following holds:

$$\mathbb{T}_{\mathcal{R}}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m) \;\Rightarrow\; \mathbb{T}_{\mathcal{R}/G}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$$

and therefore the dual, contrapositive implication also holds:

$$\mathbb{T}_{\mathcal{R}/G}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Box(v_1 \vee \ldots \vee v_m)^c \;\Rightarrow\; \mathbb{T}_{\mathcal{R}}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Box(v_1 \vee \ldots \vee v_m)^c$$

where, by definition,

$[\![(v_1 \vee \ldots \vee v_m)^c]\!]_{E \cup B} =_{def} T_{\Sigma/E \cup B, State} \setminus [\![(v_1 \vee \ldots \vee v_m)]\!]_{E \cup B}$

resp.

$[\![(v_1 \vee \ldots \vee v_m)^c]\!]_{E \cup B \cup G} =_{def} T_{\Sigma/E \cup B \cup G, State} \setminus [\![(v_1 \vee \ldots \vee v_m)]\!]_{E \cup B \cup G}$.

Main Theorem About Equational Abstractions

Theorem. For $\mathcal{R}/G$ an equational abstraction of $\mathcal{R}$ and any state predicates $u_1, \ldots, u_n, v_1, \ldots, v_m \in T_\Sigma(X)_{State}$ the following holds:

$$\mathbb{T}_{\mathcal{R}}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m) \quad \Rightarrow \quad \mathbb{T}_{\mathcal{R}/G}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$$

and therefore the dual, contrapositive implication also holds:

$$\mathbb{T}_{\mathcal{R}/G}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Box(v_1 \vee \ldots \vee v_m)^c \quad \Rightarrow \quad \mathbb{T}_{\mathcal{R}}, (u_1 \vee \ldots \vee u_n) \models_{S4} \Box(v_1 \vee \ldots \vee v_m)^c$$

where, by definition,

$[\![(v_1 \vee \ldots \vee v_m)^c]\!]_{E \cup B} =_{def} T_{\Sigma/E \cup B, State} \setminus [\![(v_1 \vee \ldots \vee v_m)]\!]_{E \cup B}$

resp.

$[\![(v_1 \vee \ldots \vee v_m)^c]\!]_{E \cup B \cup G} =_{def} T_{\Sigma/E \cup B \cup G, State} \setminus [\![(v_1 \vee \ldots \vee v_m)]\!]_{E \cup B \cup G}$.

Therefore, $\mathbb{T}_{\mathcal{R}/G}, (u_1 \vee \ldots \vee u_n) \not\models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$ proves that $(v_1 \vee \ldots \vee v_m)^c$ is an invariant from $(u_1 \vee \ldots \vee u_n)$ in $\mathbb{T}_{\mathcal{R}}$.

Using Equational Abstractions in Symbolic Model Checking

As a Corollary of the above theorem and the Completeness of Folding Narrowing Search in Lecture 25 we get:

Using Equational Abstractions in Symbolic Model Checking

As a Corollary of the above theorem and the Completeness of Folding Narrowing Search in Lecture 25 we get:

Theorem. For $\mathcal{R} = (\Sigma, E \cup B, R)$ topmost with $E \cup B$ FVP and $G = E' \cup B'$ such that $E \cup E' \cup B \cup B'$ is also FVP, $(v_1 \vee \ldots \vee v_m)^c$ is an invariant from $(u_1 \vee \ldots \vee u_n)$ in $\mathbb{T}_{\mathcal{R}}$ if $\mathbb{T}_{\mathcal{R}/G}, (u_1 \vee \ldots \vee u_n) \not\models_{S4} \Diamond(v_1 \vee \ldots \vee v_m)$,

Using Equational Abstractions in Symbolic Model Checking

As a Corollary of the above theorem and the Completeness of Folding Narrowing Search in Lecture 25 we get:

Theorem. For $\mathcal{R} = (\Sigma, E \cup B, R)$ topmost with $E \cup B$ FVP and $G = E' \cup B'$ such that $E \cup E' \cup B \cup B'$ is also FVP, $(v_1 \vee \ldots \vee v_m)^c$ is an invariant from $(u_1 \vee \ldots \vee u_n)$ in $\mathbb{T}_{\mathcal{R}}$ if $\mathbb{T}_{\mathcal{R}/G}, (u_1 \vee \ldots \vee u_n) \not\models_{S4} \diamond(v_1 \vee \ldots \vee v_m)$, i.e., if there doesn't exists $w \in FNF_{\mathcal{R}/G}(u_1 \vee \ldots \vee u_n)$ having a $E \cup E' \cup B \cup B'$-unifier $\gamma \in Unif_{E \cup E' \cup B \cup B'}(w = v_j)$ for some j, $1 \leq j \leq m$.

Using Equational Abstractions in Symbolic Model Checking

As a Corollary of the above theorem and the Completeness of Folding Narrowing Search in Lecture 25 we get:

Theorem. For $\mathcal{R} = (\Sigma, E \cup B, R)$ topmost with $E \cup B$ FVP and $G = E' \cup B'$ such that $E \cup E' \cup B \cup B'$ is also FVP, $(v_1 \vee \ldots \vee v_m)^c$ is an invariant from $(u_1 \vee \ldots \vee u_n)$ in $\mathbb{T}_{\mathcal{R}}$ if $\mathbb{T}_{\mathcal{R}/G}, (u_1 \vee \ldots \vee u_n) \not\models_{S4} \diamond(v_1 \vee \ldots \vee v_m)$, i.e., if there doesn't exists $w \in FNF_{\mathcal{R}/G}(u_1 \vee \ldots \vee u_n)$ having a $E \cup E' \cup B \cup B'$-unifier $\gamma \in Unif_{E \cup E' \cup B \cup B'}(w = v_j)$ for some j, $1 \leq j \leq m$.

Let us see a simple example illustrating the power of this Theorem.

An Equational Abstraction for BAKERY

Recall that it was impossible to verify the mutual exclusion and one-writer invariants for BAKERY from < 0, 0 > by narrowing in a forwards direction: one had to narrow backwards.

An Equational Abstraction for BAKERY

Recall that it was impossible to verify the mutual exclusion and one-writer invariants for BAKERY from < 0, 0 > by narrowing in a forwards direction: one had to narrow backwards. But we can verify both invariants by forwards narrowing in an equational abstraction of BAKERY.

An Equational Abstraction for BAKERY

Recall that it was impossible to verify the mutual exclusion and one-writer invariants for BAKERY from < 0, 0 > by narrowing in a forwards direction: one had to narrow backwards. But we can verify both invariants by forwards narrowing in an equational abstraction of BAKERY. Can you guess the G?

An Equational Abstraction for BAKERY

Recall that it was impossible to verify the mutual exclusion and one-writer invariants for BAKERY from < 0, 0 > by narrowing in a forwards direction: one had to narrow backwards. But we can verify both invariants by forwards narrowing in an equational abstraction of BAKERY. Can you guess the G?

```
mod R&W is
  sorts Nat Config .
  op <_,_> : Nat Nat -> Config [ctor] .
  op 0 : -> Nat [ctor] .
  op s : Nat -> Nat [ctor] .
  vars R W : Nat .

  rl < 0, 0 > => < 0, s(0) > [narrowing] .
  rl < R, s(W) > => < R, W > [narrowing] .
  rl < R, 0 > => < s(R), 0 > [narrowing] .
  rl < s(R), W > => < R, W > [narrowing] .
endm
```

An Equational Abstraction for BAKERY

Recall that it was impossible to verify the mutual exclusion and one-writer invariants for BAKERY from < 0, 0 > by narrowing in a forwards direction: one had to narrow backwards. But we can verify both invariants by forwards narrowing in an equational abstraction of BAKERY. Can you guess the G?

```
mod R&W is
  sorts Nat Config .
  op <_,_> : Nat Nat -> Config [ctor] .
  op 0 : -> Nat [ctor] .
  op s : Nat -> Nat [ctor] .
  vars R W : Nat .

  rl < 0, 0 > => < 0, s(0) > [narrowing] .
  rl < R, s(W) > => < R, W > [narrowing] .
  rl < R, 0 > => < s(R), 0 > [narrowing] .
  rl < s(R), W > => < R, W > [narrowing] .
endm
```

The equation < s(s(N)),0 > = < s(0),0 > is confluent, terminating and FVP and provides the desired abstraction:

An Equational Abstraction for BAKERY (II)

```
mod R&W-ABS is including R&W . eq < s(s(N:Nat)),0 > = < s(0),0 > [variant] .
endm
get variants < R:Nat, W:Nat > .

Variant 1
Config: < #1:Nat,#2:Nat >
R --> #1:Nat
W --> #2:Nat

Variant 2
Config: < s(0),0 >
R --> s(s(%1:Nat))
W --> 0

No more variants.
fvu-narrow < 0, 0 > =>* < s(N:Nat), s(M:Nat) > .  *** mutual exclusion

No solution.

fvu-narrow < 0 , 0 > =>* < N:Nat , s(s(M:Nat)) > . *** one writer

No solution.
```

Equational Abstractions for Explicit-State Model Checking

The application of equational abstraction to symbolic model checking is particularly simple.

Equational Abstractions for Explicit-State Model Checking

The application of equational abstraction to symbolic model checking is particularly simple. This is because executability conditions do not matter, since for narrowing (i.e., for symbolic execution), variant unification is enough, even when the rules R are not coherent in $\mathcal{R}/G$.

Equational Abstractions for Explicit-State Model Checking

The application of equational abstraction to symbolic model checking is particularly simple. This is because executability conditions do not matter, since for narrowing (i.e., for symbolic execution), variant unification is enough, even when the rules R are not coherent in $\mathcal{R}/G$. In fact, the rules in `R&W-ABS` are not coherent, but it did not matter at all for symbolic execution.

Equational Abstractions for Explicit-State Model Checking

The application of equational abstraction to symbolic model checking is particularly simple. This is because executability conditions do not matter, since for narrowing (i.e., for symbolic execution), variant unification is enough, even when the rules R are not coherent in $\mathcal{R}/G$. In fact, the rules in `R&W-ABS` are not coherent, but it did not matter at all for symbolic execution.

For explicit state model checking of modal logic or LTL properties, the admissibility of $\mathcal{R}/G$ is crucial.

Equational Abstractions for Explicit-State Model Checking

The application of equational abstraction to symbolic model checking is particularly simple. This is because executability conditions do not matter, since for narrowing (i.e., for symbolic execution), variant unification is enough, even when the rules R are not coherent in $\mathcal{R}/G$. In fact, the rules in `R&W-ABS` are not coherent, but it did not matter at all for symbolic execution.

For explicit state model checking of modal logic or LTL properties, the admissibility of $\mathcal{R}/G$ is crucial. Likewise, decidability by matching modulo B of state predicates u, or $u \mid \varphi$ is also crucial.

Equational Abstractions for Explicit-State Model Checking

The application of equational abstraction to symbolic model checking is particularly simple. This is because executability conditions do not matter, since for narrowing (i.e., for symbolic execution), variant unification is enough, even when the rules R are not coherent in $\mathcal{R}/G$. In fact, the rules in `R&W-ABS` are not coherent, but it did not matter at all for symbolic execution.

For explicit state model checking of modal logic or LTL properties, the admissibility of $\mathcal{R}/G$ is crucial. Likewise, decidability by matching modulo B of state predicates u, or $u \mid \varphi$ is also crucial.

For symbolic model checking the meaning of u was a subset $[\![u]\!]_{E \cup B} \subseteq T_{\Sigma/E \cup B, State}$.

Equational Abstractions for Explicit-State Model Checking

The application of equational abstraction to symbolic model checking is particularly simple. This is because executability conditions do not matter, since for narrowing (i.e., for symbolic execution), variant unification is enough, even when the rules R are not coherent in $\mathcal{R}/G$. In fact, the rules in `R&W-ABS` are not coherent, but it did not matter at all for symbolic execution.

For explicit state model checking of modal logic or LTL properties, the admissibility of $\mathcal{R}/G$ is crucial. Likewise, decidability by matching modulo B of state predicates u, or $u \mid \varphi$ is also crucial.

For symbolic model checking the meaning of u was a subset $[\![u]\!]_{E \cup B} \subseteq T_{\Sigma/E\cup B,State}$. Instead, for explicit-state model checking we need a subset $[\![u]\!]_{!_{\vec{E}/B}} \subseteq C_{\Sigma/\vec{E},B,State}$.

Equational Abstractions for Explicit-State Model Checking

The application of equational abstraction to symbolic model checking is particularly simple. This is because executability conditions do not matter, since for narrowing (i.e., for symbolic execution), variant unification is enough, even when the rules R are not coherent in $\mathcal{R}/G$. In fact, the rules in `R&W-ABS` are not coherent, but it did not matter at all for symbolic execution.

For explicit state model checking of modal logic or LTL properties, the admissibility of $\mathcal{R}/G$ is crucial. Likewise, decidability by matching modulo B of state predicates u, or $u \mid \varphi$ is also crucial.

For symbolic model checking the meaning of u was a subset $[\![u]\!]_{E \cup B} \subseteq T_{\Sigma/E \cup B, State}$. Instead, for explicit-state model checking we need a subset $[\![u]\!]_{!_{\vec{E}/B}} \subseteq C_{\Sigma/\vec{E},B,State}$. More generally, we can define $[\![u \mid \varphi]\!]_{!_{\vec{E}/B}}$ as follows:

State Predicates for Admissible Rewrite Theories

For $\mathcal{R} = (\Sigma, E \cup B, R)$ admissible with constructors Ω we require $u \in T_{\Omega}(X)_{State}$ s.t. $u = u!_{\vec{E}/B}$, and that the conjunction of Σ-equalities φ is s.t. $vars(\varphi) \subseteq vars(u)$.

State Predicates for Admissible Rewrite Theories

For $\mathcal{R} = (\Sigma, E \cup B, R)$ admissible with constructors Ω we require $u \in T_\Omega(X)_{State}$ s.t. $u = u!_{\vec{E}/B}$, and that the conjunction of Σ-equalities φ is s.t. $vars(\varphi) \subseteq vars(u)$. Then $[\![u \mid \varphi]\!]_{!_{\vec{E}/B}} = \{[v] \in C_{\Sigma/\vec{E},B,State} \mid \exists \rho \in [X \to T_\Omega] \; s.t. \; v =_B u\rho \; \wedge \; E \cup B \vdash \varphi\rho\}$.

State Predicates for Admissible Rewrite Theories

For $\mathcal{R} = (\Sigma, E \cup B, R)$ admissible with constructors Ω we require $u \in T_{\Omega}(X)_{State}$ s.t. $u = u!_{\vec{E}/B}$, and that the conjunction of Σ-equalities φ is s.t. $vars(\varphi) \subseteq vars(u)$. Then $[\![u \mid \varphi]\!]_{!_{\vec{E}/B}} = \{[v] \in C_{\Sigma/\vec{E},B,State} \mid \exists \rho \in [X \to T_{\Omega}]\ s.t.\ v =_B u\rho\ \wedge\ E \cup B \vdash \varphi\rho\}$. Since $[v] \in C_{\Sigma/\vec{E},B,State}$, this forces ρ to be a normalized substitution on $vars(u)$.

State Predicates for Admissible Rewrite Theories

For $\mathcal{R} = (\Sigma, E \cup B, R)$ admissible with constructors Ω we require $u \in T_\Omega(X)_{State}$ s.t. $u = u!_{\vec{E}/B}$, and that the conjunction of Σ-equalities φ is s.t. $vars(\varphi) \subseteq vars(u)$. Then $[\![u \mid \varphi]\!]_{!_{\vec{E}/B}} = \{[v] \in C_{\Sigma/\vec{E},B,State} \mid \exists\rho \in [X \to T_\Omega]\ s.t.\ v =_B u\rho\ \wedge\ E \cup B \vdash \varphi\rho\}$. Since $[v] \in C_{\Sigma/\vec{E},B,State}$, this forces ρ to be a normalized substitution on $vars(u)$. Note that, under these assumptions, the membership $[v] \in [\![u \mid \varphi]\!]_{!_{\vec{E}/B}}$ is decidable by B-matching and evaluation of $\varphi\rho$.

State Predicates for Admissible Rewrite Theories

For $\mathcal{R} = (\Sigma, E \cup B, R)$ admissible with constructors Ω we require $u \in T_{\Omega}(X)_{State}$ s.t. $u = u!_{\vec{E}/B}$, and that the conjunction of Σ-equalities φ is s.t. $vars(\varphi) \subseteq vars(u)$. Then $[\![u \mid \varphi]\!]_{!_{\vec{E}/B}} = \{[v] \in C_{\Sigma/\vec{E},B,State} \mid \exists \rho \in [X \to T_{\Omega}] \; s.t. \; v =_B u\rho \;\wedge\; E \cup B \vdash \varphi\rho\}$. Since $[v] \in C_{\Sigma/\vec{E},B,State}$, this forces ρ to be a normalized substitution on $vars(u)$. Note that, under these assumptions, the membership $[v] \in [\![u \mid \varphi]\!]_{!_{\vec{E}/B}}$ is decidable by B-matching and evaluation of $\varphi\rho$.

Although $(\Sigma, E \cup B)$ need not be FVP, we require that its constructor subtheory $(\Omega^{+}, E_{\Omega^{+}} \cup B_{\Omega^{+}})$ is FVP.

State Predicates for Admissible Rewrite Theories

For $\mathcal{R} = (\Sigma, E \cup B, R)$ admissible with constructors Ω we require $u \in T_\Omega(X)_{State}$ s.t. $u = u!_{\vec{E}/B}$, and that the conjunction of Σ-equalities φ is s.t. $vars(\varphi) \subseteq vars(u)$. Then $[\![u \mid \varphi]\!]_{!_{\vec{E}/B}} = \{[v] \in C_{\Sigma/\vec{E},B,State} \mid \exists \rho \in [X \to T_\Omega]\ s.t.\ v =_B u\rho\ \wedge\ E \cup B \vdash \varphi\rho\}$. Since $[v] \in C_{\Sigma/\vec{E},B,State}$, this forces ρ to be a normalized substitution on $vars(u)$. Note that, under these assumptions, the membership $[v] \in [\![u \mid \varphi]\!]_{!_{\vec{E}/B}}$ is decidable by B-matching and evaluation of $\varphi\rho$.

Although $(\Sigma, E \cup B)$ need not be FVP, we require that its constructor subtheory $(\Omega^+, E_{\Omega^+} \cup B_{\Omega^+})$ is FVP. We will the only consider equational abstractions $\mathcal{R}/G$ where $E \cup B \cup G$ is ground convergent, $G = E'_{\Omega^+} \cup B'_{\Omega^+}$ are Ω^+ equations and axioms, and $E_{\Omega^+} \cup E'_{\Omega^+} \cup B_{\Omega^+} \cup B'_{\Omega^+}$ is also FVP.

State Predicates for Admissible Rewrite Theories

For $\mathcal{R} = (\Sigma, E \cup B, R)$ admissible with constructors Ω we require $u \in T_\Omega(X)_{State}$ s.t. $u = u!_{\vec{E}/B}$, and that the conjunction of Σ-equalities φ is s.t. $vars(\varphi) \subseteq vars(u)$. Then $[\![u \mid \varphi]\!]!_{\vec{E}/B} = \{[v] \in C_{\Sigma/\vec{E},B,State} \mid \exists\rho \in [X \to T_\Omega]\ s.t.\ v =_B u\rho \ \wedge\ E \cup B \vdash \varphi\rho\}$. Since $[v] \in C_{\Sigma/\vec{E},B,State}$, this forces ρ to be a normalized substitution on $vars(u)$. Note that, under these assumptions, the membership $[v] \in [\![u \mid \varphi]\!]!_{\vec{E}/B}$ is decidable by B-matching and evaluation of $\varphi\rho$.

Although $(\Sigma, E \cup B)$ need not be FVP, we require that its constructor subtheory $(\Omega^+, E_{\Omega^+} \cup B_{\Omega^+})$ is FVP. We will the only consider equational abstractions $\mathcal{R}/G$ where $E \cup B \cup G$ is ground convergent, $G = E'_{\Omega^+} \cup B'_{\Omega^+}$ are Ω^+ equations and axioms, and $E_{\Omega^+} \cup E'_{\Omega^+} \cup B_{\Omega^+} \cup B'_{\Omega^+}$ is also FVP.

How are state predicates $[\![u \mid \varphi]\!]!_{\vec{E}/B}$ in $\mathcal{R}$ and $[\![u' \mid \varphi']\!]!_{\vec{E} \cup \vec{E'}_{\Omega^+}/B \cup B'_{\Omega^+}}$ in $\mathcal{R}/G$ related?

State Predicates for Admissible Rewrite Theories

For $\mathcal{R} = (\Sigma, E \cup B, R)$ admissible with constructors Ω we require $u \in T_{\Omega}(X)_{State}$ s.t. $u = u!_{\vec{E}/B}$, and that the conjunction of Σ-equalities φ is s.t. $vars(\varphi) \subseteq vars(u)$. Then $[\![u \mid \varphi]\!]_{!_{\vec{E}/B}} = \{[v] \in C_{\Sigma/\vec{E},B,State} \mid \exists \rho \in [X \to T_{\Omega}]\ s.t.\ v =_B u\rho \ \wedge\ E \cup B \vdash \varphi\rho\}$. Since $[v] \in C_{\Sigma/\vec{E},B,State}$, this forces ρ to be a normalized substitution on $vars(u)$. Note that, under these assumptions, the membership $[v] \in [\![u \mid \varphi]\!]_{!_{\vec{E}/B}}$ is decidable by B-matching and evaluation of $\varphi\rho$.

Although $(\Sigma, E \cup B)$ need not be FVP, we require that its constructor subtheory $(\Omega^{+}, E_{\Omega^{+}} \cup B_{\Omega^{+}})$ is FVP. We will the only consider equational abstractions $\mathcal{R}/G$ where $E \cup B \cup G$ is ground convergent, $G = E'_{\Omega^{+}} \cup B'_{\Omega^{+}}$ are Ω^{+} equations and axioms, and $E_{\Omega^{+}} \cup E'_{\Omega^{+}} \cup B_{\Omega^{+}} \cup B'_{\Omega^{+}}$ is also FVP.

How are state predicates $[\![u \mid \varphi]\!]_{!_{\vec{E}/B}}$ in $\mathcal{R}$ and $[\![u' \mid \varphi']\!]_{!_{\vec{E}\cup\vec{E'}_{\Omega^{+}}/B\cup B'_{\Omega^{+}}}}$ in $\mathcal{R}/G$ related? This can be answered as follows:

G-Abstractable State Predicates

Call a state predicate $u \mid \varphi$ in $\mathcal{R}$ G-abstractable

G-Abstractable State Predicates

Call a state predicate $u \mid \varphi$ in $\mathcal{R}$ G-abstractable if for $(u'_1, \gamma_1), \ldots, (u'_k, \gamma_k)$ the $E_{\Omega^+} \cup B_{\Omega^+}$-variants of u, we have $vars((\varphi\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+} / B \cup B'_{\Omega+}}) \subseteq vars(u_i)$ $1 \leq i \leq k$.

G-Abstractable State Predicates

Call a state predicate $u \mid \varphi$ in $\mathcal{R}$ G-abstractable if for $(u'_1, \gamma_1), \ldots, (u'_k, \gamma_k)$ the $E_{\Omega^+} \cup B_{\Omega^+}$-variants of u, we have $vars((\varphi\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega^+} / B \cup B'_{\Omega^+}}) \subseteq vars(u_i)$ $1 \leq i \leq k$. Abbreviate $(\varphi\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega^+} / B \cup B'_{\Omega^+}}$ to φ'_i and call $u'_1 \mid \varphi'_1 \vee \ldots \vee u'_k \mid \varphi'_k$ the G-abstraction of $u \mid \varphi$ in $\mathcal{R}/G$.

G-Abstractable State Predicates

Call a state predicate $u \mid \varphi$ in $\mathcal{R}$ G-abstractable if for $(u'_1, \gamma_1), \ldots, (u'_k, \gamma_k)$ the $E_{\Omega^+} \cup B_{\Omega^+}$-variants of u, we have $vars((\varphi\gamma_i)!_{\vec{E}\cup\vec{E'}_{\Omega+}/B\cup B'_{\Omega+}}) \subseteq vars(u_i)$ $1 \leq i \leq k$. Abbreviate $(\varphi\gamma_i)!_{\vec{E}\cup\vec{E'}_{\Omega+}/B\cup B'_{\Omega+}}$ to φ'_i and call $u'_1 \mid \varphi'_1 \vee \ldots \vee u'_k \mid \varphi'_k$ the G-abstraction of $u \mid \varphi$ in $\mathcal{R}/G$.

Consider now the unique surjective Σ-homomorphism:

$$[\_!_{\vec{E}\cup\vec{E'}_{\Omega+}/B\cup B'_{\Omega+}}] : \mathbb{C}_{\Sigma/\vec{E},B} \to \mathbb{C}_{\Sigma/\vec{E},\vec{E'}_{\Omega+}/B\cup B'_{\Omega+}}$$

G-Abstractable State Predicates

Call a state predicate $u \mid \varphi$ in $\mathcal{R}$ G-abstractable if for $(u'_1, \gamma_1), \ldots, (u'_k, \gamma_k)$ the $E_{\Omega+} \cup B_{\Omega+}$-variants of u, we have $vars((\varphi\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}) \subseteq vars(u_i)$ $1 \leq i \leq k$. Abbreviate $(\varphi\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}$ to φ'_i and call $u'_1 \mid \varphi'_1 \vee \ldots \vee u'_k \mid \varphi'_k$ the G-abstraction of $u \mid \varphi$ in $\mathcal{R}/G$.

Consider now the unique surjective Σ-homomorphism:

$$[\_!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}] : \mathbb{C}_{\Sigma/\vec{E},B} \rightarrow \mathbb{C}_{\Sigma/\vec{E},\vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}$$

A key theorem, proved in the Appendix, is:

G-Abstractable State Predicates

Call a state predicate $u \mid \varphi$ in $\mathcal{R}$ G-abstractable if for $(u'_1, \gamma_1), \ldots, (u'_k, \gamma_k)$ the $E_{\Omega+} \cup B_{\Omega+}$-variants of u, we have $vars((\varphi\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}) \subseteq vars(u_i)$ $1 \leq i \leq k$. Abbreviate $(\varphi\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}$ to φ'_i and call $u'_1 \mid \varphi'_1 \vee \ldots \vee u'_k \mid \varphi'_k$ the G-abstraction of $u \mid \varphi$ in $\mathcal{R}/G$.

Consider now the unique surjective Σ-homomorphism:

$$[\_!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}] : \mathbb{C}_{\Sigma/\vec{E},B} \to \mathbb{C}_{\Sigma/\vec{E},\vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}$$

A key theorem, proved in the Appendix, is:

Theorem. The image of the set $[\![u \mid \varphi]\!]_{!_{\vec{E}/B}}$ under the above homomorphism is contained in the set $[\![(u'_1 \mid \varphi'_1 \vee \ldots \vee u'_k \mid \varphi'_k)]\!]_{!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}}$.

G-Abstractable State Predicates

Call a state predicate $u \mid \varphi$ in $\mathcal{R}$ G-abstractable if for $(u'_1, \gamma_1), \ldots, (u'_k, \gamma_k)$ the $E_{\Omega+} \cup B_{\Omega+}$-variants of u, we have $vars((\varphi\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}) \subseteq vars(u_i)$ $1 \leq i \leq k$. Abbreviate $(\varphi\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}$ to φ'_i and call $u'_1 \mid \varphi'_1 \vee \ldots \vee u'_k \mid \varphi'_k$ the G-abstraction of $u \mid \varphi$ in $\mathcal{R}/G$.

Consider now the unique surjective Σ-homomorphism:

$$[\_!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}] : \mathbb{C}_{\Sigma/\vec{E},B} \rightarrow \mathbb{C}_{\Sigma/\vec{E},\vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}$$

A key theorem, proved in the Appendix, is:

Theorem. The image of the set $[\![u \mid \varphi]\!]_{!_{\vec{E}/B}}$ under the above homomorphism is contained in the set $[\![(u'_1 \mid \varphi'_1 \vee \ldots \vee u'_k \mid \varphi'_k)]\!]_{!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}}$.

Let us see an example.

Abstractable State Predicates for R&W

In R&W, state predicates for the complements of the mutual exclusion and one writer invariants are, respectively, `< s(N:Nat) , s(M:Nat) >` and `< N:Nat , s(s(M:Nat)) >`. What are their corresponding *G*-abstractions in R&W-ABS?

Abstractable State Predicates for R&W

In R&W, state predicates for the complements of the mutual exclusion and one writer invariants are, respectively, `< s(N:Nat), s(M:Nat) >` and `< N:Nat , s(s(M:Nat)) >`. What are their corresponding *G*-abstractions in R&W-ABS?

```
get variants < s(N:Nat), s(M:Nat) > .
Variant 1
Config: < s(#1:Nat),s(#2:Nat) >
N --> #1:Nat
M --> #2:Nat

No more variants.

get variants  < N:Nat , s(s(M:Nat)) > .
Variant 1
Config: < #1:Nat,s(s(#2:Nat)) >
N --> #1:Nat
M --> #2:Nat

No more variants.
```

Abstractable State Predicates for R&W

In R&W, state predicates for the complements of the mutual exclusion and one writer invariants are, respectively, `< s(N:Nat), s(M:Nat) >` and `< N:Nat , s(s(M:Nat)) >`. What are their corresponding *G*-abstractions in R&W-ABS?

```
get variants < s(N:Nat), s(M:Nat) > .
Variant 1
Config: < s(#1:Nat),s(#2:Nat) >
N --> #1:Nat
M --> #2:Nat

No more variants.

get variants  < N:Nat , s(s(M:Nat)) > .
Variant 1
Config: < #1:Nat,s(s(#2:Nat)) >
N --> #1:Nat
M --> #2:Nat

No more variants.
```

Up to renaming of variables, they are the same.

G-Abstractable Rewrite Rules

Even though equational abstraction can be used for any admissible rewrite theory $\mathcal{R}$, executability of $\mathcal{R}/G$ is easier to achieve when $\mathcal{R}$ is topmost, for which making $\mathcal{R}/G$ executable is closely connected with the notion of a rule in $\mathcal{R}$ being G-abstractable.

G-Abstractable Rewrite Rules

Even though equational abstraction can be used for any admissible rewrite theory $\mathcal{R}$, executability of $\mathcal{R}/G$ is easier to achieve when $\mathcal{R}$ is topmost, for which making $\mathcal{R}/G$ executable is closely connected with the notion of a rule in $\mathcal{R}$ being G-abstractable.

Under the same assumptions on G, call a rule $l \to r \;\; if \;\; \varphi$ in $\mathcal{R}$ (where we assume $vars(r) \cup vars(\varphi) \subseteq vars(l)$) G-abstractable iff

G-Abstractable Rewrite Rules

Even though equational abstraction can be used for any admissible rewrite theory $\mathcal{R}$, executability of $\mathcal{R}/G$ is easier to achieve when $\mathcal{R}$ is topmost, for which making $\mathcal{R}/G$ executable is closely connected with the notion of a rule in $\mathcal{R}$ being G-abstractable.

Under the same assumptions on G, call a rule $l \rightarrow r \;\; if \;\; \varphi$ in $\mathcal{R}$ (where we assume $vars(r) \cup vars(\varphi) \subseteq vars(l)$) G-abstractable iff for $(l'_1, \gamma_1), \ldots, (l'_k, \gamma_k)$ the $E_{\Omega+} \cup B_{\Omega+}$-variants of l, we have $vars((r\gamma_i)!_{\vec{E}\cup\vec{E'}_{\Omega+}/B\cup B'_{\Omega+}}) \cup vars((\varphi\gamma_i)!_{\vec{E}\cup\vec{E'}_{\Omega+}/B\cup B'_{\Omega+}}) \subseteq vars(l'_i)$ $1 \leq i \leq k$.

G-Abstractable Rewrite Rules

Even though equational abstraction can be used for any admissible rewrite theory $\mathcal{R}$, executability of $\mathcal{R}/G$ is easier to achieve when $\mathcal{R}$ is topmost, for which making $\mathcal{R}/G$ executable is closely connected with the notion of a rule in $\mathcal{R}$ being G-abstractable.

Under the same assumptions on G, call a rule $l \to r \;\; if \;\; \varphi$ in $\mathcal{R}$ (where we assume $vars(r) \cup vars(\varphi) \subseteq vars(l)$) G-abstractable iff for $(l'_1, \gamma_1), \ldots, (l'_k, \gamma_k)$ the $E_{\Omega^+} \cup B_{\Omega^+}$-variants of l, we have $vars((r\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}) \cup vars((\varphi\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}) \subseteq vars(l'_i)$ $1 \leq i \leq k$. Call $\{l'_i \to r'_i \;\; if \;\; \varphi'_i\}_{1 \leq i \leq k}$ the G-abstraction of $l \to r \;\; if \;\; \varphi$, where $r'_i =_{def} (r\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}$, and

$\varphi'_i =_{def} (\varphi\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}$.

G-Abstractable Rewrite Rules

Even though equational abstraction can be used for any admissible rewrite theory $\mathcal{R}$, executability of $\mathcal{R}/G$ is easier to achieve when $\mathcal{R}$ is topmost, for which making $\mathcal{R}/G$ executable is closely connected with the notion of a rule in $\mathcal{R}$ being G-abstractable.

Under the same assumptions on G, call a rule $l \rightarrow r \;\; if \;\; \varphi$ in $\mathcal{R}$ (where we assume $vars(r) \cup vars(\varphi) \subseteq vars(l)$) G-abstractable iff for $(l'_1, \gamma_1), \ldots, (l'_k, \gamma_k)$ the $E_{\Omega^+} \cup B_{\Omega^+}$-variants of l, we have $vars((r\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}) \cup vars((\varphi\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}) \subseteq vars(l'_i)$ $1 \leq i \leq k$. Call $\{l'_i \rightarrow r'_i \;\; if \;\; \varphi'_i\}_{1 \leq i \leq k}$ the G-abstraction of $l \rightarrow r \;\; if \;\; \varphi$, where $r'_i =_{def} (r\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}$, and $\varphi'_i =_{def} (\varphi\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}$. Let $\widehat{\mathcal{R}/G}$ have rules $\widehat{R}$ replacing each $l \rightarrow r \;\; if \;\; \varphi$ in $\mathcal{R}/G$ by its G-abstraction.

G-Abstractable Rewrite Rules

Even though equational abstraction can be used for any admissible rewrite theory $\mathcal{R}$, executability of $\mathcal{R}/G$ is easier to achieve when $\mathcal{R}$ is topmost, for which making $\mathcal{R}/G$ executable is closely connected with the notion of a rule in $\mathcal{R}$ being G-abstractable.

Under the same assumptions on G, call a rule $l \to r \;\; if \;\; \varphi$ in $\mathcal{R}$ (where we assume $vars(r) \cup vars(\varphi) \subseteq vars(l)$) G-abstractable iff for $(l'_1, \gamma_1), \ldots, (l'_k, \gamma_k)$ the $E_{\Omega^+} \cup B_{\Omega^+}$-variants of l, we have $vars((r\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}) \cup vars((\varphi\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}) \subseteq vars(l'_i)$ $1 \leq i \leq k$. Call $\{l'_i \to r'_i \;\; if \;\; \varphi'_i\}_{1 \leq i \leq k}$ the G-abstraction of $l \to r \;\; if \;\; \varphi$, where $r'_i =_{def} (r\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}$, and $\varphi'_i =_{def} (\varphi\gamma_i)!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}$. Let $\widehat{\mathcal{R}/G}$ have rules $\widehat{R}$ replacing each $l \to r \;\; if \;\; \varphi$ in $\mathcal{R}/G$ by its G-abstraction. Then (see Appendix):

Theorem. If all rules in $\mathcal{R}$ are G-abstractable, $\widehat{\mathcal{R}/G}$ is admissible.

G-Abstraction of Rules for `R&W`

Let us compute the G-variants of all lefthand sides of rules `R&W` in the theory `R&W-ABS`:

G-Abstraction of Rules for R&W

Let us compute the G-variants of all lefthand sides of rules R&W in the theory R&W-ABS:

```
get variants < 0, 0 > .  *** For rule  rl < 0, 0 > => < 0, s(0) > .

Variant 1
Config: < 0,0 >

No more variants.

*** Its G-abstraction is itself.

get variants < R, s(W) > . *** For rule rl < R, s(W) > => < R, W > .

Variant 1
Config: < #1:Nat,s(#2:Nat) >
R --> #1:Nat
W --> #2:Nat

No more variants.

*** Its G-abstraction is itself
```

G-Abstraction of Rules for R&W (II)

```
Maude> get variants < R, 0 > . *** For rule rl < R, s(W) > => < R, W > .

Variant 1
Config: < #1:Nat,0 >
R --> #1:Nat

Variant 2
Config: < s(0),0 >
R --> s(s(%1:Nat))

No more variants.

*** G-abstraction: itself and < s(0) , 0 > => < s(s(R)), 0 >! = < s(0) , 0 > .

get variants < s(R),W > . ***  For rule rl < s(R), W > => < R, W > .

Variant 1
Config: < s(#1:Nat),#2:Nat >
R --> #1:Nat
W --> #2:Nat
```

G-Abstraction of Rules for R&W (III)

```
Variant 2
Config: < s(0),0 >
R --> s(%1:Nat)
W --> 0

*** Its G-abstraction includes itself, but rule
***
***  < s(0), 0 > => < s(N), 0 > .
***
*** is NOT EXECUTABLE. However, in R&W-ABS we can prove the inductive theorem:
***
***   < s(N), 0 > = < s(0) , 0 >   using as generator set {0,s(x)}
***
*** so we get the semantically equivalent EXECUTABLE rule:
***
***                           < s(0), 0 > => < s(0), 0 > .
*** making R&W-ABS ADMISSIBLE.
```

G-Abstraction of Rules for R&W (III)

```
Variant 2
Config: < s(0),0 >
R --> s(%1:Nat)
W --> 0

*** Its G-abstraction includes itself, but rule
***
***  < s(0), 0 > => < s(N), 0 > .
***
*** is NOT EXECUTABLE. However, in R&W-ABS we can prove the inductive theorem:
***
***   < s(N), 0 > = < s(0) , 0 >   using as generator set {0,s(x)}
***
*** so we get the semantically equivalent EXECUTABLE rule:
***
***                           < s(0), 0 > => < s(0), 0 > .
*** making R&W-ABS ADMISSIBLE.
```

Since we have made R&W-ABS admissible as the system module:

G-Abstraction of Rules for R&W (IV)

```
mod R&W-ABS-ADMISSIBLE is
    including R&W .
    vars N M R W : Nat .

    eq < s(s(N)),0 > = < s(0),0 > [variant] .
    rl < s(0) , 0 > => < s(0) , 0 > .
endm
```

G-Abstraction of Rules for R&W (IV)

```
mod R&W-ABS-ADMISSIBLE is
    including R&W .
    vars N M R W : Nat .

    eq < s(s(N)),0 > = < s(0),0 > [variant] .
    rl < s(0) , 0 > => < s(0) , 0 > .
endm
```

we can use it to verify properties of R&W by search:

G-Abstraction of Rules for R&W (IV)

```
mod R&W-ABS-ADMISSIBLE is
    including R&W .
    vars N M R W : Nat .

    eq < s(s(N)),0 > = < s(0),0 > [variant] .
    rl < s(0) , 0 > => < s(0) , 0 > .
endm
```

we can use it to verify properties of R&W by search:

```
search < 0, 0 > =>* < s(N), s(M) > .

No solution.

search < 0, 0 > =>* < N, s(s(M)) > .

No solution.
```

thanks to the following Main Theorem (proof in the Appendix):

Main Theorem on Equational Abstractions

Main Theorem (Explicit-State Model Checking with Equational Abstractions). For $\mathcal{R}$ topmost and admissible with all its rules G-abstractable and $(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m)$ such that each $v_i \mid \varphi_i$ is abstractable as $v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i}$. The following holds for any initial states $[u] \in \mathbb{C}_{\mathcal{R}}$, $[u!] = [u!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}] \in \mathbb{C}_{\mathcal{R}/G}$:

Main Theorem on Equational Abstractions

Main Theorem (Explicit-State Model Checking with Equational Abstractions). For $\mathcal{R}$ topmost and admissible with all its rules G-abstractable and $(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m)$ such that each $v_i \mid \varphi_i$ is abstractable as $v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i}$. The following holds for any initial states $[u] \in \mathbb{C}_{\mathcal{R}}$, $[u!] = [u!_{\vec{E} \cup \vec{E}'_{\Omega+}/B \cup B'_{\Omega+}}] \in \mathbb{C}_{\mathcal{R}/G}$:

$$\mathbb{C}_{\mathcal{R}}, [u] \models_{S4} \diamond(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m) \;\Rightarrow\; \mathbb{C}_{\widehat{\mathcal{R}/G}}, [u!] \models_{S4} \diamond \bigvee_{1 \leq i \leq m} (v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i})$$

Main Theorem on Equational Abstractions

Main Theorem (Explicit-State Model Checking with Equational Abstractions). For $\mathcal{R}$ topmost and admissible with all its rules G-abstractable and $(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m)$ such that each $v_i \mid \varphi_i$ is abstractable as $v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i}$. The following holds for any initial states $[u] \in \mathbb{C}_{\mathcal{R}}$, $[u!] = [u!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}] \in \mathbb{C}_{\mathcal{R}/G}$:

$$\mathbb{C}_{\mathcal{R}}, [u] \models_{S4} \Diamond(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m) \quad \Rightarrow \quad \mathbb{C}_{\widehat{\mathcal{R}/G}}, [u!] \models_{S4} \Diamond \bigvee_{1 \leq i \leq m} (v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i})$$

and therefore the dual, contrapositive implication also holds:

Main Theorem on Equational Abstractions

Main Theorem (Explicit-State Model Checking with Equational Abstractions). For $\mathcal{R}$ topmost and admissible with all its rules G-abstractable and $(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m)$ such that each $v_i \mid \varphi_i$ is abstractable as $v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i}$. The following holds for any initial states $[u] \in \mathbb{C}_{\mathcal{R}}$, $[u!] = [u!_{\vec{E} \cup \vec{E'}_{\Omega+} / B \cup B'_{\Omega+}}] \in \mathbb{C}_{\mathcal{R}/G}$:

$$\mathbb{C}_{\mathcal{R}}, [u] \models_{S4} \Diamond(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m) \quad \Rightarrow \quad \mathbb{C}_{\widehat{\mathcal{R}/G}}, [u!] \models_{S4} \Diamond \bigvee_{1 \leq i \leq m} (v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i})$$

and therefore the dual, contrapositive implication also holds:

$$\mathbb{C}_{\widehat{\mathcal{R}/G}}, [u!] \models_{S4} \Box(\bigvee_{1 \leq i \leq m} (v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i}))^c \Rightarrow \mathbb{C}_{\mathcal{R}}, [u] \models_{S4} \Box(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m)^c$$

Main Theorem on Equational Abstractions

Main Theorem (Explicit-State Model Checking with Equational Abstractions). For $\mathcal{R}$ topmost and admissible with all its rules G-abstractable and $(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m)$ such that each $v_i \mid \varphi_i$ is abstractable as $v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i}$. The following holds for any initial states $[u] \in \mathbb{C}_{\mathcal{R}}$, $[u!] = [u!_{\vec{E} \cup \vec{E'}_{\Omega^+}/B \cup B'_{\Omega^+}}] \in \mathbb{C}_{\mathcal{R}/G}$:

$$\mathbb{C}_{\mathcal{R}}, [u] \models_{S4} \diamond(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m) \;\; \Rightarrow \;\; \mathbb{C}_{\widehat{\mathcal{R}/G}}, [u!] \models_{S4} \diamond \bigvee_{1 \leq i \leq m} (v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i})$$

and therefore the dual, contrapositive implication also holds:

$$\mathbb{C}_{\widehat{\mathcal{R}/G}}, [u!] \models_{S4} \Box(\bigvee_{1 \leq i \leq m} (v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i}))^c \Rightarrow \mathbb{C}_{\mathcal{R}}, [u] \models_{S4} \Box(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m)^c$$

Therefore,

Main Theorem on Equational Abstractions

Main Theorem (Explicit-State Model Checking with Equational Abstractions). For $\mathcal{R}$ topmost and admissible with all its rules G-abstractable and $(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m)$ such that each $v_i \mid \varphi_i$ is abstractable as $v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i}$. The following holds for any initial states $[u] \in \mathbb{C}_{\mathcal{R}}$, $[u!] = [u!_{\vec{E}\cup\vec{E'}_{\Omega+}/B\cup B'_{\Omega+}}] \in \mathbb{C}_{\mathcal{R}/G}$:

$$\mathbb{C}_{\mathcal{R}}, [u] \models_{S4} \Diamond(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m) \;\Rightarrow\; \mathbb{C}_{\widehat{\mathcal{R}/G}}, [u!] \models_{S4} \Diamond \bigvee_{1\leq i\leq m} (v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i})$$

and therefore the dual, contrapositive implication also holds:

$$\mathbb{C}_{\widehat{\mathcal{R}/G}}, [u!] \models_{S4} \Box(\bigvee_{1\leq i\leq m} (v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i}))^c \Rightarrow \mathbb{C}_{\mathcal{R}}, [u] \models_{S4} \Box(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m)^c$$

Therefore,

$$\mathbb{C}_{\widehat{\mathcal{R}/G}}, [u!] \not\models_{S4} \Diamond \bigvee_{1\leq i\leq m} (v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i})$$

Main Theorem on Equational Abstractions

Main Theorem (Explicit-State Model Checking with Equational Abstractions). For $\mathcal{R}$ topmost and admissible with all its rules G-abstractable and $(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m)$ such that each $v_i \mid \varphi_i$ is abstractable as $v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i}$. The following holds for any initial states $[u] \in \mathbb{C}_{\mathcal{R}}$, $[u!] = [u!_{\vec{E} \cup \vec{E'}_{\Omega+}/B \cup B'_{\Omega+}}] \in \mathbb{C}_{\mathcal{R}/G}$:

$$\mathbb{C}_{\mathcal{R}}, [u] \models_{S4} \Diamond(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m) \;\Rightarrow\; \mathbb{C}_{\widehat{\mathcal{R}/G}}, [u!] \models_{S4} \Diamond \bigvee_{1 \leq i \leq m} (v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i})$$

and therefore the dual, contrapositive implication also holds:

$$\mathbb{C}_{\widehat{\mathcal{R}/G}}, [u!] \models_{S4} \Box(\bigvee_{1 \leq i \leq m} (v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i}))^c \Rightarrow \mathbb{C}_{\mathcal{R}}, [u] \models_{S4} \Box(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m)^c$$

Therefore,

$$\mathbb{C}_{\widehat{\mathcal{R}/G}}, [u!] \not\models_{S4} \Diamond \bigvee_{1 \leq i \leq m} (v'_{i,1} \mid \varphi'_{i,1} \vee \ldots \vee v'_{i,k_i} \mid \varphi'_{i,k_i})$$

proves that $(v_1 \mid \varphi_1 \vee \ldots \vee v_m \mid \varphi_m)^c$ is an invariant from $[u]$ in $\mathbb{C}_{\mathcal{R}}$.

Equational Abstractions for Explicit-State Model Checking: the LTL Case

Equational abstractions can also be used for explicit-state LTL model checking.

Equational Abstractions for Explicit-State Model Checking: the LTL Case

Equational abstractions can also be used for explicit-state LTL model checking. The requirements are:

Equational Abstractions for Explicit-State Model Checking: the LTL Case

Equational abstractions can also be used for explicit-state LTL model checking. The requirements are:

1. those for model checking modal logic properties of a topmost $\mathcal{R}$ using $\widehat{\mathcal{R}/G}$ and `search`, as explained above, plus:

Equational Abstractions for Explicit-State Model Checking: the LTL Case

Equational abstractions can also be used for explicit-state LTL model checking. The requirements are:

1. those for model checking modal logic properties of a topmost $\mathcal{R}$ using $\widehat{\mathcal{R}/G}$ and `search`, as explained above, plus:
2. $\mathcal{R}$ (or at least the set of states reachable from the initial state(s)) must be deadlock-free, or made so by adding an extra, conditional rule to loop on deadlock states (always possible, and easy for topmost rewrite theories), and

Equational Abstractions for Explicit-State Model Checking: the LTL Case

Equational abstractions can also be used for explicit-state LTL model checking. The requirements are:

1. those for model checking modal logic properties of a topmost $\mathcal{R}$ using $\widehat{\mathcal{R}/G}$ and `search`, as explained above, plus:
2. $\mathcal{R}$ (or at least the set of states reachable from the initial state(s)) must be deadlock-free, or made so by adding an extra, conditional rule to loop on deadlock states (always possible, and easy for topmost rewrite theories), and
3. (i) specifying state predicates in both the `true` and `false` cases in $\mathcal{R}$-PREDS,

Equational Abstractions for Explicit-State Model Checking: the LTL Case

Equational abstractions can also be used for explicit-state LTL model checking. The requirements are:

1. those for model checking modal logic properties of a topmost $\mathcal{R}$ using $\widehat{\mathcal{R}/G}$ and `search`, as explained above, plus:
2. $\mathcal{R}$ (or at least the set of states reachable from the initial state(s)) must be deadlock-free, or made so by adding an extra, conditional rule to loop on deadlock states (always possible, and easy for topmost rewrite theories), and
3. (i) specifying state predicates in both the `true` and `false` cases in $\mathcal{R}$-PREDS, (ii) using their G-abstractions in $\mathcal{R}/G$-PREDS, and

Equational Abstractions for Explicit-State Model Checking: the LTL Case

Equational abstractions can also be used for explicit-state LTL model checking. The requirements are:

1. those for model checking modal logic properties of a topmost $\mathcal{R}$ using $\widehat{\mathcal{R}/G}$ and `search`, as explained above, plus:
2. $\mathcal{R}$ (or at least the set of states reachable from the initial state(s)) must be deadlock-free, or made so by adding an extra, conditional rule to loop on deadlock states (always possible, and easy for topmost rewrite theories), and
3. (i) specifying state predicates in both the `true` and `false` cases in $\mathcal{R}$-PREDS, (ii) using their G-abstractions in $\mathcal{R}/G$-PREDS, and (iii) $\mathcal{R}/G$-PREDS must protect BOOL.

Equational Abstractions for Explicit-State Model Checking: the LTL Case

Equational abstractions can also be used for explicit-state LTL model checking. The requirements are:

1. those for model checking modal logic properties of a topmost $\mathcal{R}$ using $\widehat{\mathcal{R}/G}$ and `search`, as explained above, plus:
2. $\mathcal{R}$ (or at least the set of states reachable from the initial state(s)) must be deadlock-free, or made so by adding an extra, conditional rule to loop on deadlock states (always possible, and easy for topmost rewrite theories), and
3. (i) specifying state predicates in both the `true` and `false` cases in $\mathcal{R}$-PREDS, (ii) using their G-abstractions in $\mathcal{R}/G$-PREDS, and (iii) $\mathcal{R}/G$-PREDS must protect BOOL.

Main Theorem. Under requirements (1)–(3), if $\widehat{\mathcal{R}/G}, [u!] \models_{LTL} \varphi$, then $\mathcal{R}, [u] \models_{LTL} \varphi$ for any $\varphi \in LTL(\Pi)$. (Proof in Appendix).

Explicit-State LTL Model Checking of R&W

For R&W requirement (1) is fulfilled by R&W-ABS-ADMISSIBLE and requirement (2) by R&W is deadlock free. Consider the predicates:

Explicit-State LTL Model Checking of R&W

For R&W requirement (1) is fulfilled by R&W-ABS-ADMISSIBLE and requirement (2) by R&W is deadlock free. Consider the predicates:

```
in model-checker.maude

mod R&W-PREDS is protecting R&W .  extending SATISFACTION .
   subsort Config < State .
   ops mutex one-writer reads writes : -> Prop .
   eq < s(N:Nat),s(M:Nat) > |= mutex = false .
   eq < 0,N:Nat > |= mutex = true .
   eq < N:Nat,0 > |= mutex = true .
   eq < N:Nat,s(s(M:Nat)) > |= one-writer = false .
   eq < N:Nat,0 > |= one-writer = true .
   eq < N:Nat,s(0) > |= one-writer = true .
   eq < s(N:Nat), M:Nat > |= reads = true .
   eq < 0, M:Nat > |= reads = false .
   eq < M:Nat, s(N:Nat) > |= writes = true .
   eq < N:Nat, 0 > |= writes = false .
endm
```

Explicit-State LTL Model Checking of R&W

For R&W requirement (1) is fulfilled by R&W-ABS-ADMISSIBLE and requirement (2) by R&W is deadlock free. Consider the predicates:

```
in model-checker.maude

mod R&W-PREDS is protecting R&W .  extending SATISFACTION .
   subsort Config < State .
   ops mutex one-writer reads writes : -> Prop .
   eq < s(N:Nat),s(M:Nat) > |= mutex = false .
   eq < 0,N:Nat > |= mutex = true .
   eq < N:Nat,0 > |= mutex = true .
   eq < N:Nat,s(s(M:Nat)) > |= one-writer = false .
   eq < N:Nat,0 > |= one-writer = true .
   eq < N:Nat,s(0) > |= one-writer = true .
   eq < s(N:Nat), M:Nat > |= reads = true .
   eq < 0, M:Nat > |= reads = false .
   eq < M:Nat, s(N:Nat) > |= writes = true .
   eq < N:Nat, 0 > |= writes = false .
endm
```

In the negative cases of `mutex` and `one-writer` we checked that their G-abstractions are themselves. For all other cases we get:

Explicit-State LTL Model Checking of R&W (II)

```
get variants < 0,N:Nat > . *** For eq < 0,N:Nat > |= mutex = true .

Variant 1
Config: < 0,#1:Nat >
N --> #1:Nat

No more variants.
*** The G-abstraction is itself

get variants < N:Nat,0 > .  *** For eq < N:Nat,0 > |= mutex = true .

Variant 1
Config: < #1:Nat,0 >
N --> #1:Nat

Variant 2
Config: < s(0),0 >
N --> s(s(%1:Nat))

No more variants.
*** The G-abstraction adds the equation < s(0),0 > |= mutex = true .
```

Explicit-State LTL Model Checking of R&W (III)

```
get variants < N:Nat,0 > .  *** For eq < N:Nat,0 > |= one-writer = true .
                            *** has already been computed for mutex

*** The G-abstraction adds the equation < s(0),0 > |= one-writer = true .

get variants < N:Nat,s(0) > .  *** For eq < N:Nat,s(0) > |= one-writer = true .

Variant 1
Config: < #1:Nat,s(0) >
N --> #1:Nat

No more variants.
*** The G-abstraction is itself

get variants < s(N:Nat), M:Nat > . *** For < s(N:Nat), M:Nat > |= reads = true

Variant 1
Config: < s(#1:Nat),#2:Nat >
N --> #1:Nat
M --> #2:Nat
```

Explicit-State LTL Model Checking of R&W (IV)

```
Variant 2
Config: < s(0),0 >
N --> s(%1:Nat)
M --> 0

No more variants.
*** The G-abstraction adds < s(0),0 > |= reads = true .

get variants < 0, M:Nat > .  *** For < 0, M:Nat > |= reads = false .

Variant 1
Config: < 0,#1:Nat >
M --> #1:Nat

No more variants.
*** The G-abstraction is itself
```

Explicit-State LTL Model Checking of R&W (V)

```
get variants < M:Nat, s(N:Nat) > . *** For < M:Nat, s(N:Nat) > |= writes = true

Variant 1
rewrites: 0 in 0ms cpu (0ms real) (0 rewrites/second)
Config: < #1:Nat,s(#2:Nat) >
M:Nat --> #1:Nat
N:Nat --> #2:Nat

No more variants.
*** The G-abstraction is itself

   < N:Nat, 0 > |= writes = false .

get variants < N:Nat, 0 > *** For < N:Nat, 0 > |= writes = false .
                          *** same variants as for eq mutex(< N:Nat,0 >) = true

*** The G-abstraction adds the equation < s(0),0 > |= writes = false .
```

Explicit-State LTL Model Checking of R&W (V)

```
get variants < M:Nat, s(N:Nat) > . *** For < M:Nat, s(N:Nat) > |= writes = true

Variant 1
rewrites: 0 in 0ms cpu (0ms real) (0 rewrites/second)
Config: < #1:Nat,s(#2:Nat) >
M:Nat --> #1:Nat
N:Nat --> #2:Nat

No more variants.
*** The G-abstraction is itself

   < N:Nat, 0 > |= writes = false .

get variants < N:Nat, 0 > *** For < N:Nat, 0 > |= writes = false .
                          *** same variants as for eq mutex(< N:Nat,0 >) = true

*** The G-abstraction adds the equation < s(0),0 > |= writes = false .
```

Therefore, we get the following modules `R&W-ABS-ADMISSIBLE-PREDS` and `R&W-ABS-ADMISSIBLE-CHECK`:

Explicit-State LTL Model Checking of R&W (VI)

```
mod R&W-ABS-ADMISSIBLE-PREDS is protecting R&W-ABS-ADMISSIBLE .
    including R&W-PREDS .
    eq < s(0),0 > |= mutex = true .
    eq < s(0),0 > |= one-writer = true .
    eq < s(0),0 > |= reads = true .
    eq < s(0),0 > |= writes = false .
endm

mod R&W-ABS-ADMISSIBLE-CHECK is protecting R&W-ABS-ADMISSIBLE-PREDS .
    including MODEL-CHECKER .
endm

red modelCheck(< 0,0 >,[] mutex) .

result Bool: true

red modelCheck(< 0,0 >,[] one-writer) .

result Bool: true
```

Explicit-State LTL Model Checking of R&W (VII)

```
red modelCheck(< 0,0 >,[] <> reads) .

result ModelCheckResult:
counterexample(nil, {< 0,0 >,unlabeled} {< 0,s(0) >,unlabeled})

red modelCheck(< 0,0 >,[] <> writes) .

result ModelCheckResult:
counterexample({< 0,0 >,unlabeled}, {< s(0),0 >,unlabeled})

red modelCheck(< 0,0 >,[] <> (reads \/ writes)) .

result Bool: true
```