The Impact of Craig’s Interpolation Theorem

in Computer Science

Cesare Tinelli

tinelli@cs.uiowa.edu

The University of Iowa
The Role of Logic in Computer Science

Mathematical logic is central to Computer Science
Mathematical logic is central to Computer Science

It provides formal foundations for

- Programming languages
- Relational databases
- Computational complexity
- Hardware design and validation
- Formal methods in software engineering
- Artificial Intelligence

...
Craig’s Interpolation Theorem in CS

- has had a strong and lasting impact in several CS areas, both at the theoretical and the practical level
Craig’s Interpolation Theorem in CS

- has had a strong and lasting impact in several CS areas, both at the theoretical and the practical level

- has been generalized to many other logics used in CS (sorted, equational, modal, intuitionistic, . . . )
Craig’s Interpolation Theorem in CS

- has had a strong and lasting impact in several CS areas, both at the theoretical and the practical level.
- has been generalized to many other logics used in CS (sorted, equational, modal, intuitionistic, . . . )
- together with compactness, is considered a crucial property of any new logic for CS.
Craig’s Interpolation Theorem in CS

- has had a strong and lasting impact in several CS areas, both at the theoretical and the practical level
- has been generalized to many other logics used in CS (sorted, equational, modal, intuitionistic, . . . )
- together with compactness, is considered a crucial property of any new logic for CS
- comes up in any formal method based on modular decomposition of complex systems
Craig’s Interpolation

Some applications:

- **Hardware/software specification** (Diaconescu et al., ’93, Rosu & Goguen, 2000, Bicarregui et al., 2000)
Craig’s Interpolation

Some applications:

- **Hardware/software specification** (Diaconescu et al., ’93, Rosu & Goguen, 2000, Bicarregui et al., 2000)

- **Reasoning with large knowledge bases** (Amir & McIlraith, 2005)
Craig’s Interpolation

Some applications:

- **Hardware/software specification** (Diaconescu et al., '93, Rosu & Goguen, 2000, Bicarregui et al., 2000)
- **Reasoning with large knowledge bases** (Amir & McIlraith, 2005)
- **Type inference** (Jhala et al., 2007)
Craig’s Interpolation

Some applications:

- **Hardware/software specification** (Diaconescu et al., '93, Rosu & Goguen, 2000, Bicarregui et al., 2000)
- **Reasoning with large knowledge bases** (Amir & McIlraith, 2005)
- **Type inference** (Jhala et al., 2007)
- **Combination of theorem provers for different theories** (Nelson & Oppen, 1979; Tinelli, 2003; Ghilardi, 2005)
Craig’s Interpolation

Some applications:

- **Hardware/software specification** (Diaconescu et al., '93, Rosu & Goguen, 2000, Bicarregui et al., 2000)
- **Reasoning with large knowledge bases** (Amir & McIlraith, 2005)
- **Type inference** (Jhala et al., 2007)
- **Combination of theorem provers for different theories** (Nelson & Oppen, 1979; Tinelli, 2003; Ghilardi, 2005)
- **Model checking of finite- and infinite-state systems** (McMillan, 2003, Henzinger et al., 2004)
Craig’s Interpolation

Some applications:

- **Hardware/software specification** (Diaconescu et al., ’93, Rosu & Goguen, 2000, Bicarregui et al., 2000)
- **Reasoning with large knowledge bases** (Amir & McIlraith, 2005)
- **Type inference** (Jhala et al., 2007)
- **Combination of theorem provers for different theories** (Nelson & Oppen, 1979; Tinelli, 2003; Ghilardi, 2005)
- **Model checking of finite- and infinite-state systems** (McMillan, 2003, Henzinger et al., 2004)
Craig’s Interpolation: If $\varphi_1$ and $\varphi_2$ are inconsistent, there is a $\varphi$ in their shared language such that

$$\varphi_1 \models \psi \text{ and } \psi \land \varphi_2 \text{ is inconsistent.}$$
Craig’s Interpolation: If \( \varphi_1 \) and \( \varphi_2 \) are inconsistent, there is a \( \varphi \) in their shared language such that

\[
\varphi_1 \models \psi \quad \text{and} \quad \psi \land \varphi_2 \text{ is inconsistent.}
\]

Intuitively,

1. \( \psi \) is an abstraction of \( \varphi_1 \) from the viewpoint of \( \varphi_2 \);
2. \( \psi \) summarizes and translates in the shared language why \( \varphi_1 \) is inconsistent with \( \varphi_2 \).
Part I: Craig Interpolation for Prover Combinations
In some areas of Computer Science, one is interested in the satisfiability in a particular theory of certain classes of formulas.
In some areas of Computer Science, one is interested in the **satisfiability in a particular theory** of certain classes of formulas.

- Microprocessors design: theory of **equality**, atoms like \( f(g(a, b), c) = g(c, a) \).
In some areas of Computer Science, one is interested in the **satisfiability in a particular theory** of certain classes of formulas.

- Microprocessors design: theory of **equality**, atoms like $f(g(a, b), c) = g(c, a)$.
In some areas of Computer Science, one is interested in the **satisfiability in a particular theory** of certain classes of formulas.

- **Microprocessors design:** theory of **equality**, atoms like $f(g(a, b), c) = g(c, a)$.
- **Timed automata, planning:** theory of **integers/reals**, atoms like $x - y < 2$.
- **Software verification/model checking:** combination of theories, atoms like $5 + \text{first}((x + 2) :: l) = a[j] + 1$. 
In some areas of Computer Science, one is interested in the satisfiability in a particular theory of certain classes of formulas.

- Microprocessors design: theory of equality, atoms like \( f(g(a, b), c) = g(c, a) \).
- Timed automata, planning: theory of integers/reals, atoms like \( x - y < 2 \).
- Software verification/model checking: combination of theories, atoms like \( 5 + \text{first}((x + 2) :: l) = a[j] + 1 \).

We refer to this general problem as Satisfiability Modulo Theories, or SMT.
In some areas of Computer Science, one is interested in the **satisfiability in a particular theory** of certain classes of formulas.

- Microprocessors design: theory of equality, atoms like $f(g(a, b), c) = g(c, a)$.

- Timed automata, planning: theory of integers/reals, atoms like $x - y < 2$.

- Software verification/model checking: combination of theories, atoms like $5 + \text{first}((x + 2) :: l) = a[j] + 1$.

We refer to this general problem as **Satisfiability Modulo Theories**, or SMT.
Let $T$ be a first-order theory of signature $\Sigma$ and $\mathcal{L}^\Sigma$ a class of $\Sigma$-formulas.
Satisfiability Modulo Theories

Let $T$ be a first-order theory of signature $\Sigma$ and $\mathcal{L}^\Sigma$ a class of $\Sigma$-formulas.

The $T$-satisfiability problem for $\mathcal{L}^\Sigma$ consists in deciding for any formula $\varphi[x] \in \mathcal{L}^\Sigma$ whether $T \cup \{\exists x. \varphi\}$ is satisfiable.
Let $T$ be a first-order theory of signature $\Sigma$ and $\mathcal{L}^\Sigma$ a class of $\Sigma$-formulas.

The $T$-satisfiability problem for $\mathcal{L}^\Sigma$ consists in deciding for any formula $\varphi[x] \in \mathcal{L}^\Sigma$ whether $T \cup \{\exists x. \varphi\}$ is satisfiable.

Some relevant theories in SMT

- Equality with “Uninterpreted Function Symbols”
- Linear Arithmetic (Real and Integer)
- Arrays (i.e., updatable maps)
- Bit vectors
- Finite trees
Solving Combined SMT Problems

For many theories $T$ and some formula classes $\mathcal{L}$ there exist (efficient) decision procedures for the $T$-satisfiability problem for $\mathcal{L}^\Sigma$. 
For many theories $T$ and some formula classes $\mathcal{L}$ there exist (efficient) decision procedures for the $T$-satisfiability problem for $\mathcal{L}^\Sigma$.

**Problem:** In practice, we often need to deal with mixed formulas in $\mathcal{L}^\Sigma_1 \cup \cdots \cup \Sigma_n$ modulo a combined theory $T_1 \cup \cdots \cup T_n$. 
Solving Combined SMT Problems

For many theories $T$ and some formula classes $\mathcal{L}$ there exist (efficient) decision procedures for the $T$-satisfiability problem for $\mathcal{L}^\Sigma$.

**Problem:** In practice, we often need to deal with mixed formulas in $\mathcal{L}^{\Sigma_1 \cup \cdots \cup \Sigma_n}$ modulo a combined theory $T_1 \cup \cdots \cup T_n$.

In that case, it helps if we can

combine modularly decision procedures for the individual $T_1, \ldots, T_n$ into a decision procedure for $T_1 \cup \cdots \cup T_n$. 
The General Combined Satisfiability Problem

For $i = 1, 2$,

- let $T_i$ a first-order theory of signature $\Sigma_i$ and
- let $\mathcal{L}^{\Sigma_i}$ be a class of $\Sigma_i$-formulas

such that the $T_i$-satisfiability problem for $\mathcal{L}^{\Sigma_i}$ is decidable.
The General Combined Satisfiability Problem

For \( i = 1, 2, \)

- let \( T_i \) a first-order theory of signature \( \Sigma_i \) and
- let \( L^{\Sigma_i} \) be a class of \( \Sigma_i \)-formulas

such that the \( T_i \)-satisfiability problem for \( L^{\Sigma_i} \) is decidable.

Combination methods apply to languages \( L^{\Sigma_1 \cup \Sigma_2} \) that are effectively purifiable for \( T_1 \) and \( T_2 \)
The General Combined Satisfiability Problem

For \( i = 1, 2 \),

1. let \( T_i \) a first-order theory of signature \( \Sigma_i \) and

2. let \( \mathcal{L}^{\Sigma_i} \) be a class of \( \Sigma_i \)-formulas

such that the \( T_i \)-satisfiability problem for \( \mathcal{L}^{\Sigma_i} \) is decidable.

Combination methods apply to languages \( \mathcal{L}^{\Sigma_1 \cup \Sigma_2} \) that are effectively purifiable for \( T_1 \) and \( T_2 \), i.e., such that

the \( (T_1 \cup T_2) \)-satisfiability of a formula \( \varphi \in \mathcal{L}^{\Sigma_1 \cup \Sigma_2} \)

is effectively reducible to

the \( (T_1 \cup T_2) \)-satisfiability of formulas of the form \( \varphi_1 \land \varphi_2 \)

with \( \varphi_i \in \mathcal{L}^{\Sigma_i} \) for \( i = 1, 2 \).
The General Combined Satisfiability Problem

For $i = 1, 2$,

- let $T_i$ a first-order theory of signature $\Sigma_i$ and
- let $\mathcal{L}^{\Sigma_i}$ be a class of $\Sigma_i$-formulas

such that the $T_i$-satisfiability problem for $\mathcal{L}^{\Sigma_i}$ is decidable.

Combination methods apply to languages $\mathcal{L}^{\Sigma_1 \cup \Sigma_2}$ that are effectively purifiable for $T_1$ and $T_2$

Observation: For purifiable languages, $(T_1 \cup T_2)$-satisfiability is at heart an interpolation problem.
Combined Satisfiability as Interpolation

For $i = 1, 2$, let $T_i$-be a $\Sigma_i$-theory and $\varphi_i[x_i]$ a $\Sigma_i$-formula.

$\varphi_1 \land \varphi_2$ is $(T_1 \cup T_2)$-unsatisfiable
Combined Satisfiability as Interpolation

For $i = 1, 2$, let $T_i$-be a $\Sigma_i$-theory and $\varphi_i[x_i]$ a $\Sigma_i$-formula.

$$\varphi_1 \land \varphi_2 \text{ is } (T_1 \cup T_2)\text{-unsatisfiable}$$

iff

$$T_1, \varphi_1, T_2, \varphi_2 \models \bot$$
Combined Satisfiability as Interpolation

For $i = 1, 2$, let $T_i$-be a $\Sigma_i$-theory and $\varphi_i[x_i]$ a $\Sigma_i$-formula.

$\varphi_1 \land \varphi_2$ is $(T_1 \cup T_2)$-unsatisfiable

iff

$T_1, \varphi_1, T_2, \varphi_2 \models \bot$}

iff, by an application of Craig’s interpolation theorem, there is a $(\Sigma_1 \cap \Sigma_2)$-formula $\varphi(x)$ with $x = x_1 \cap x_2$ s.t.

$T_1, \varphi_1 \models \varphi$ and $T_2, \varphi_2, \varphi \models \bot$
Combined Satisfiability as Interpolation

For $i = 1, 2$, let $T_i$-be a $\Sigma_i$-theory and $\varphi_i[x_i]$ a $\Sigma_i$-formula.

$$\varphi_1 \land \varphi_2 \text{ is } (T_1 \cup T_2)\text{-unsatisfiable}$$

iff

$$T_1, \varphi_1, T_2, \varphi_2 \models \perp$$

iff, by an application of Craig’s interpolation theorem, there is a $(\Sigma_1 \cap \Sigma_2)$-formula $\varphi(x)$ with $x = x_1 \cap x_2$ s.t.

$$T_1, \varphi_1 \models \varphi \text{ and } T_2, \varphi_2, \varphi \models \perp$$

The problem then is “just” computing the interpolant $\varphi$. 
Combined Satisfiability as Interpolation

For $i = 1, 2$, let $T_i$-be a $\Sigma_i$-theory and $\varphi_i[x_i]$ a $\Sigma_i$-formula.

$\varphi_1 \land \varphi_2$ is $(T_1 \cup T_2)$-unsatisfiable

iff

$T_1, \varphi_1, T_2, \varphi_2 \models \bot$

iff, by an application of Craig’s interpolation theorem, there is a $(\Sigma_1 \cap \Sigma_2)$-formula $\varphi(x)$ with $x = x_1 \cap x_2$ s.t.

$T_1, \varphi_1 \models \varphi$ and $T_2, \varphi_2, \varphi \models \bot$

All existing combination methods are in essence ways to compute $\varphi$, possibly incrementally, in finite time, without building a direct proof that $T_1, \varphi_1, T_2, \varphi_2 \models \bot$
For $i = 1, 2$, let $T_i$-be a $\Sigma_i$-theory and $\varphi_i[x_i]$ a $\Sigma_i$-formula.

$\varphi_1 \land \varphi_2$ is $(T_1 \cup T_2)$-unsatisfiable

iff

$T_1, \varphi_1, T_2, \varphi_2 \models \bot$

iff, by an application of Craig’s interpolation theorem, there is a $(\Sigma_1 \cap \Sigma_2)$-formula $\varphi(x)$ with $x = x_1 \cap x_2$ s.t.

$T_1, \varphi_1 \models \varphi$ and $T_2, \varphi_2, \varphi \models \bot$

Historical note: The original correctness proof of the foremost combination method for SMT (Nelson & Oppen, 1979) relies directly on Craig’s interpolation theorem.
The class of quantifier-free formulas is effectively purifiable for any $\Sigma_1$ and $\Sigma_2$
The class of quantifier-free formulas is effectively purifiable for any $\Sigma_1$ and $\Sigma_2$:

Given a quantifier-free $(\Sigma_1 \cup \Sigma_2)$-formula $\varphi$
we can compute $\Sigma_1$-qffs $\varphi_1^1 \ldots \varphi_n^1$ and $\Sigma_2$-qffs $\varphi_2^1 \ldots \varphi_n^2$ s.t.
for every $(\Sigma_1 \cup \Sigma_2)$-structure $\mathcal{A}$,
$\varphi$ is satisfiable in $\mathcal{A}$ iff $\varphi_1^j \land \varphi_2^j$ is satisfiable in $\mathcal{A}$ for some $j$. 
The class of quantifier-free formulas is effectively purifiable for any $\Sigma_1$ and $\Sigma_2$.

Moreover, the $T$-satisfiability problem for qffs is decidable for a very large number of theories of interest in CS.
The class of quantifier-free formulas is effectively purifiable for 
any $\Sigma_1$ and $\Sigma_2$.

Moreover, the $T$-satisfiability problem for qffs is decidable for 
a very large number of theories of interest in CS.

Let’s focus then on quantifier-free formulas.
The class of quantifier-free formulas is effectively purifiable for any $\Sigma_1$ and $\Sigma_2$.

Moreover, the $T$-satisfiability problem for qffs is decidable for a very large number of theories of interest in CS.

Let’s focus then on quantifier-free formulas.

For simplicity, but wlog, let’s consider only combined satisfiability problems of the form

$$\Gamma_1 \cup \Gamma_2$$

where each $\Gamma_i$ is a finite set of $\Sigma_i$-literals (i.e., atomic formulas and negated atomic formulas)
The Combined Satisfiability Problem for QFFs

For $i = 1, 2$, let $T_i$-be a $\Sigma_i$-theory and $\Gamma_i[x_i]$ a set of $\Sigma_i$-litersals.
The Combined Satisfiability Problem for QFFs

For $i = 1, 2$, let $T_i$-be a $\Sigma_i$-theory and $\Gamma_i[x_i]$ a set of $\Sigma_i$-literals.
Let $\psi_1, \ldots, \psi_n$ be $(\Sigma_1 \cap \Sigma_2)$-formulas over $x = x_1 \cap x_2$. 
For $i = 1, 2$, let $T_i$ be a $\Sigma_i$-theory and $\Gamma_i[x_i]$ a set of $\Sigma_i$-literals.

Let $\psi_1, \ldots, \psi_n$ be $(\Sigma_1 \cap \Sigma_2)$-formulas over $x = x_1 \cap x_2$.

$\psi_1, \ldots, \psi_n$ is an interpolation chain if for each $k = 1, \ldots, m$ there is an $i \in \{1, 2\}$ s.t.

$$T_i, \Gamma_i, \psi_1, \ldots, \psi_{k-1} \models \psi_k$$
The Combined Satisfiability Problem for QFFs

For $i = 1, 2$, let $T_i$-be a $\Sigma_i$-theory and $\Gamma_i[x_i]$ a set of $\Sigma_i$-literals.

Let $\psi_1, \ldots, \psi_n$ be $(\Sigma_1 \cap \Sigma_2)$-formulas over $x = x_1 \cap x_2$.

$\psi_1, \ldots, \psi_n$ is an interpolation chain if for each $k = 1, \ldots, m$ there is an $i \in \{1, 2\}$ s.t.

$$T_i, \Gamma_i, \psi_1, \ldots, \psi_{k-1} \models \psi_k$$

Under the right conditions:

1. $\Gamma_1 \cup \Gamma_2$ is $(T_1 \cup T_2)$-unsatisfiable iff there is an interpolation chain $\psi_1, \ldots, \psi_m$ with $\psi_n = \bot$, and

2. each $\psi_i$ is a disjunction of atoms and is computable using one of the decision procedures for $T_1$ and $T_2$. 
The Combined Satisfiability Problem for QFFs

Sufficient conditions on $T_1$ and $T_2$ (Ghilardi, 2005)

Where $\Sigma_0 = \Sigma_1 \cap \Sigma_2$, there is a universal $\Sigma_0$-theory $T_0$ that is:
The Combined Satisfiability Problem for QFFs

Sufficient conditions on $T_1$ and $T_2$ (Ghilardi, 2005)

Where $\Sigma_0 = \Sigma_1 \cap \Sigma_2$, there is a universal $\Sigma_0$-theory $T_0$ that is:

1. $T_i$-compatible for $i = 1, 2$:
   
   (a) is enclosed in $T_i$
   
   (b) admits a model completion $T_0^*$
   
   (c) every model of $T_i$ embeds into a model of $T_i \cup T_0^*$
The Combined Satisfiability Problem for QFFs

Sufficient conditions on \( T_1 \) and \( T_2 \) (Ghilardi, 2005)

Where \( \Sigma_0 = \Sigma_1 \cap \Sigma_2 \), there is a universal \( \Sigma_0 \)-theory \( T_0 \) that is:

1. \( T_i \)-compatible for \( i = 1, 2 \):
   (a) is enclosed in \( T_i \)
   (b) admits a model completion \( T_0^* \)
   (c) every model of \( T_i \) embeds into a model of \( T_i \cup T_0^* \)

2. effectively locally finite:
   For any \( x \) we can compute a set \( \{ t_1, \ldots t_n \} \) of \( \Sigma_0 \)-terms over \( x \) s.t. every \( \Sigma_0 \)-term \( t[x] \) is \( T_0 \)-equivalent to some \( t_i \)
The Combined Satisfiability Problem for QFFs

Sufficient conditions on $T_1$ and $T_2$ (Ghilardi, 2005)

Where $\Sigma_0 = \Sigma_1 \cap \Sigma_2$, there is a universal $\Sigma_0$-theory $T_0$ that is:

1. $T_i$-compatible for $i = 1, 2$:
   (a) is enclosed in $T_i$
   (b) admits a model completion $T_0^*$
   (c) every model of $T_i$ embeds into a model of $T_i \cup T_0^*$

2. effectively locally finite:
   For any $x$ we can compute a set $\{t_1, \ldots, t_n\}$ of $\Sigma_0$-terms over $x$ s.t. every $\Sigma_0$-term $t[x]$ is $T_0$-equivalent to some $t_i$

Nelson-Oppen Method: $\Sigma_0 = \emptyset$ and each $T_i$ is stably infinite.
A $\Sigma$-theory $T$ is **stably infinite** iff every quantifier-free $T$-satisfiable formula is satisfiable in an infinite model of $T$. 
Stably Infinite Theories

A $\Sigma$-theory $T$ is stably infinite iff every quantifier-free $T$-satisfiable formula is satisfiable in an infinite model of $T$.

Many *interesting* theories are stably infinite:

- Theories of an infinite structure.
- Complete theories with an infinite model.
- Convex theories with no trivial models.
A $\Sigma$-theory $T$ is stably infinite iff every quantifier-free $T$-satisfiable formula is satisfiable in an infinite model of $T$.

Many interesting theories are stably infinite:

- Theories of an infinite structure.
- Complete theories with an infinite model.
- Convex theories with no trivial models.

But others are not:

- Theories of a finite structure.
- Theories with models of bounded cardinality.
- Some equational/Horn theories.
Recent extensions of the Nelson-Oppen method partially lift the stable infiniteness requirement (Tinelli & Zarba, 2004; Ranise et al., 2005)
Recent extensions of the Nelson-Oppen method partially lift the stable infiniteness requirement (Tinelli & Zarba, 2004; Ranise et al., 2005)

The trick is to require the decision procedures to also exchange finite-cardinality constraints.
Recent extensions of the Nelson-Oppen method partially lift the stable infiniteness requirement (Tinelli & Zarba, 2004; Ranise et al., 2005)

The trick is to require the decision procedures to also exchange finite-cardinality constraints.

These extensions are still instances of Craig interpolation.
Recent extensions of the Nelson-Oppen method partially
lift the stable infiniteness requirement
(Tinelli & Zarba, 2004; Ranise et al., 2005)

The trick is to require the decision procedures to also
exchange finite-cardinality constraints.

These extensions are still instances of Craig interpolation.

However, they now consider interpolation chains that also
include quantified formulas like

$$\forall x, y, z. \ x = y \lor x = z$$
SMT provers based on some variant of the Nelson-Oppen method are widely used in academia and industry.
The Combined Satisfiability Problem for QFFs

- SMT provers based on some variant of the Nelson-Oppen method are widely used in academia and industry.

- The generalized results by Ghilardi have several additional applications.
  For instance, they can be used in the combination of modal logics.
Part II: Craig Interpolation in Model Checking
Software or hardware systems can be often modeled as state transition systems $\mathcal{M} = (S, I, R, L)$ where

- $S$ is a set of states
- $I \subseteq S$ is a set of initial states
- $R \subseteq S \times S$ is a total transition relation
- $L : S \rightarrow 2^{At}$ is a labelling function into sets of atomic formulas in some base logic
Software or hardware systems can be often modeled as *state transition systems* $\mathcal{M} = (S, I, R, L)$ where

- $S$ is a set of *states*
- $I \subseteq S$ is a set of *initial states*
- $R \subseteq S \times S$ is a total *transition relation*
- $L : S \rightarrow 2^{At}$ is a *labelling function* into sets of atomic formulas in some base logic

**Note:** $\mathcal{M}$ is a Kripke model (in the sense modal logic).
Software or hardware systems can be often modeled as state transition systems, or model, $\mathcal{M} = (S, I, R, L)$.
Software or hardware systems can be often modeled as state transition systems, or model, $\mathcal{M} = (S, I, R, L)$.

Most system correctness properties can be expressed as a safety property for a suitable model $\mathcal{M}$:

$\mathcal{M}$ is safe wrt a property $\psi$ if no state $R$-reachable from an initial state satisfies $\psi$. 
Software or hardware systems can be often modeled as *state transition systems*, or *model*, \( \mathcal{M} = (S, I, R, L) \)

Most system correctness properties can be expressed as a *safety* property for a suitable model \( \mathcal{M} \):

\[ \mathcal{M} \text{ is safe wrt a property } \psi \text{ if no state } R\text{-reachable from an initial state satisfies } \psi. \]

Model checking is one of the most successful areas of formal verification.

Model checking technologies are now routinely used in industry.
A model $\mathcal{M} = (S, I, R, L: S \to 2^{At})$ can be expressed symbolically by fixing a set $X$ of variables and a first-order $\Sigma$-structure $\mathcal{A}$ with universe $A$. 
Symbolic Model Checking

A model \( M = (S, I, R, L:S \rightarrow 2^{At}) \) can be expressed symbolically by fixing a set \( X \) of variables and a first-order \( \Sigma \)-structure \( A \) with universe \( A \).

Then:

- Every state \( \sigma \in S \) is a mapping in \([X \rightarrow A]\)
- \( At \) is a set of atomic \( \Sigma \)-formulas over \( X \)
- \( I \) is characterized by a qff \( \varphi_I[x] \) s.t. \( \sigma \in I \) iff \( A \models \varphi_I[\sigma] \)
- \( R \) is characterized by a qff \( \varphi_R[x, x'] \) such that \( (\sigma, \sigma') \in R \) iff \( A \models \varphi_R[\sigma, \sigma'] \)

**Notation:** if \( x = x_1, \ldots, x_n \) then \( \psi[\sigma] = \psi[\sigma(x_1), \ldots, \sigma(x_n)] \)
A state $\sigma$ is *reachable (in $k$ steps)* iff there is a sequence of states $\sigma_0, \ldots, \sigma_k = \sigma$ such that

$$A \models \varphi_I[\sigma_0] \land \varphi_R[\sigma_0, \sigma_1] \land \cdots \land \varphi_R[\sigma_{k-1}, \sigma_k]$$

A formula $\psi[x]$ is *reachable (in $k$ steps)* from a formula $\varphi[x]$ iff there is a sequence of states $\sigma_0, \ldots, \sigma_k = \sigma$ s.t.

$$A \models \varphi[\sigma_0] \land \varphi_R[\sigma_0, \sigma_1] \land \cdots \land \varphi_R[\sigma_{k-1}, \sigma_k] \land \psi[\sigma_k]$$
A state $\sigma$ is \textit{reachable (in $k$ steps)} iff there is a sequence of states $\sigma_0, \ldots, \sigma_k = \sigma$ such that

$$\mathcal{A} \models \varphi_I[\sigma_0] \land \varphi_R[\sigma_0, \sigma_1] \land \cdots \land \varphi_R[\sigma_{k-1}, \sigma_k]$$

A formula $\psi[x]$ is \textit{reachable (in $k$ steps)} from a formula $\varphi[x]$ iff there is a sequence of states $\sigma_0, \ldots, \sigma_k = \sigma$ s.t.

$$\mathcal{A} \models \varphi[\sigma_0] \land \varphi_R[\sigma_0, \sigma_1] \land \cdots \land \varphi_R[\sigma_{k-1}, \sigma_k] \land \psi[\sigma_k]$$

\textbf{Observation:} $\mathcal{M}$ is safe wrt $\psi$ iff $\psi$ is not reachable from $\varphi_I$ iff

$$\varphi_I[x_0] \land \varphi_R[x_0, x_1] \land \cdots \land \varphi_R[x_{k-1}, x_k] \land \psi[x_k]$$

is unsatisfiable in $\mathcal{A}$ for all $k \geq 0$. 
For a large class of systems $\mathcal{M}$, we can compute from $\varphi_I$ and $\varphi_R$ the strongest inductive invariant $\varphi_{IR}$ for $\mathcal{M}$:
Strongest Inductive Invariant

For a large class of systems $\mathcal{M}$, we can compute from $\varphi_I$ and $\varphi_R$ the strongest inductive invariant $\varphi_{IR}$ for $\mathcal{M}$:

for all $\sigma \in S$, $\mathcal{A} \models \varphi_{IR}[\sigma]$ exactly when $\sigma$ is reachable.
For a large class of systems $\mathcal{M}$, we can compute from $\varphi_I$ and $\varphi_R$ the strongest inductive invariant $\varphi_{IR}$ for $\mathcal{M}$:

for all $\sigma \in S$, $\mathcal{A} \models \varphi_{IR}[\sigma]$ exactly when $\sigma$ is reachable.

Then, to check that $\mathcal{M}$ is safe wrt to a property $\psi$ it suffices to check that $\varphi_{IR}[x] \land \psi[x]$ is unsatisfiable in $\mathcal{A}$.
For a large class of systems $\mathcal{M}$, we can compute from $\varphi_I$ and $\varphi_R$ the strongest inductive invariant $\varphi_{IR}$ for $\mathcal{M}$:

for all $\sigma \in S$, $A \models \varphi_{IR}[\sigma]$ exactly when $\sigma$ is reachable.

Then, to check that $\mathcal{M}$ is safe wrt to a property $\psi$ it suffices to check that $\varphi_{IR}[x] \land \psi[x]$ is unsatisfiable in $A$.

This can be completely automated if the satisfiability in $A$ of qffs is decidable.
For a large class of systems $\mathcal{M}$, we can compute from $\varphi_I$ and $\varphi_R$ the strongest inductive invariant $\varphi_{IR}$ for $\mathcal{M}$:

$$\text{for all } \sigma \in S, \mathcal{A} \models \varphi_{IR}[\sigma] \text{ exactly when } \sigma \text{ is reachable.}$$

Then, to check that $\mathcal{M}$ is safe wrt to a property $\psi$ it suffices to check that $\varphi_{IR}[x] \land \psi[x]$ is unsatisfiable in $\mathcal{A}$.

This can be completely automated if the satisfiability in $\mathcal{A}$ of qffs is decidable.

**Problem:** Computing $\varphi_{IR}$ can be very expensive.
Strongest Inductive Invariant

For a large class of systems $\mathcal{M}$, we can compute from $\varphi_I$ and $\varphi_R$ the strongest inductive invariant $\varphi_{IR}$ for $\mathcal{M}$:

for all $\sigma \in S$, $\mathcal{A} \models \varphi_{IR}[\sigma]$ exactly when $\sigma$ is reachable.

Then, to check that $\mathcal{M}$ is safe wrt to a property $\psi$ it suffices to check that $\varphi_{IR}[x] \land \psi[x]$ is unsatisfiable in $\mathcal{A}$.

This can be completely automated if the satisfiability in $\mathcal{A}$ of qffs is decidable.

Problem: Computing $\varphi_{IR}$ can be very expensive.

Good news: Craig interpolation can be used to reduce this cost.
Computing Strongest Inductive Invariants

When $\varphi_{IR}$ is computable it is because it is the least fix point of an *image* operator $Img : QFF \rightarrow QFF$ where

- $Img(\varphi[x])$ is the strongest (wrt $\models_\mathcal{A}$, entailment in $\mathcal{A}$) qff $\varphi_p[x]$ such that
  \[ \varphi[x] \land \varphi_R[x, x'] \models_\mathcal{A} \varphi_p[x'] \]

- $\varphi_{IR} = \bigwedge_{i \geq 0} \varphi^i$ with $\varphi^0 = \varphi_I$ and $\varphi^{i+1} = \varphi^i \lor Img(\varphi^i)$
Computing Strongest Inductive Invariants

When $\varphi_{IR}$ is computable it is because it is the least fix point of an *image* operator $Img : \text{QFF} \rightarrow \text{QFF}$ where

1. $Img(\varphi[x])$ is the strongest (wrt $\models_A$, entailment in $A$) qff $\varphi_p[x]$ such that

   $$\varphi[x] \land \varphi_R[x, x'] \models_A \varphi_p[x']$$

2. $\varphi_{IR} = \bigwedge_{i \geq 0} \varphi^i$ with $\varphi^0 = \varphi_I$ and $\varphi^{i+1} = \varphi^i \lor Img(\varphi^i)$

Computing $Img$, and so $\varphi_{IR}$, is expensive because it involves quantifier elimination.
Computing Strongest Inductive Invariants

When $\varphi_{IR}$ is computable it is because it is the least fix point of an image operator $\text{Img} : \text{QFF} \rightarrow \text{QFF}$ where

- $\text{Img}(\varphi[x])$ is the strongest (wrt $\models_{\mathcal{A}}$, entailment in $\mathcal{A}$) qff $\varphi_p[x]$ such that

\[
\varphi[x] \land \varphi_R[x, x'] \models_{\mathcal{A}} \varphi_p[x']
\]

- $\varphi_{IR} = \bigwedge_{i \geq 0} \varphi^i$ with $\varphi^0 = \varphi_I$ and $\varphi^{i+1} = \varphi^i \lor \text{Img}(\varphi^i)$

Computing $\text{Img}$, and so $\varphi_{IR}$, is expensive because it involves quantifier elimination.

However, $\text{Img}$ might be much stronger than needed for proving that a property $\psi$ is unreachable.
Computing Strongest Inductive Invariants

When \( \varphi_{IR} \) is computable it is because it is the least fix point of an *image* operator \( \text{Img} : QFF \rightarrow QFF \) where

\[
\text{Img}(\varphi[x]) \text{ is the strongest (wrt } \models_A, \text{ entailment in } A) \text{ qff } \\
\varphi_p[x] \text{ such that } \\
\varphi[x] \land \varphi_R[x, x'] \models_A \varphi_p[x']
\]

\[
\varphi_{IR} = \bigwedge_{i \geq 0} \varphi^i \text{ with } \varphi^0 = \varphi_I \text{ and } \varphi^{i+1} = \varphi^i \lor \text{Img}(\varphi^i)
\]

Computing \( \text{Img} \), and so \( \varphi_{IR} \), is expensive because it involves quantifier elimination.

**Idea** (McMillan, 2003):

use interpolation to compute for each \( i \geq 0 \) an *adequate over-approximation* \( \hat{\varphi}^i \) of \( \varphi^i \) wrt \( \psi \)
How to compute $\hat{\phi}_I R$ for $\psi$ incrementally

Let $k > 0$, $\hat{\phi}^0 = \phi_I[x]$

Base Case) Let:

$$\Gamma_1 = \hat{\phi}[x_0] \land \varphi_R[x_0, x_1]$$
$$\Gamma_2 = \varphi_R[x_1, x_2] \land \cdots \land \varphi_R[x_{k-1}, x_k] \land (\psi[x_1] \lor \cdots \lor \psi[x_k])$$
How to compute $\hat{\varphi}_{IR}$ for $\psi$ incrementally

Let $k > 0$, $\hat{\varphi}^0 = \varphi_I[x]$

Base Case) Let:

\[
\begin{align*}
\Gamma_1 &= \hat{\varphi}^0[x_0] \land \varphi_R[x_0, x_1] \\
\Gamma_2 &= \varphi_R[x_1, x_2] \land \cdots \land \varphi_R[x_{k-1}, x_k] \land (\psi[x_1] \lor \cdots \lor \psi[x_k])
\end{align*}
\]

If $\Gamma_1 \land \Gamma_2$ is satisfiable in $A$, we are done:

$\psi$ is reachable from $\varphi_I$ in 1 to $k$ steps.
How to compute $\hat{\varphi}_{IR}$ for $\psi$ incrementally

Let $k > 0$, $\hat{\varphi}^0 = \varphi_I[x]$

**Base Case**) Let:

$$\Gamma_1 = \hat{\varphi}^0[x_0] \land \varphi_R[x_0, x_1]$$
$$\Gamma_2 = \varphi_R[x_1, x_2] \land \cdots \land \varphi_R[x_{k-1}, x_k] \land (\psi[x_1] \lor \cdots \lor \psi[x_k])$$

If $\Gamma_1 \land \Gamma_2$ is unsatisfiable in $\mathcal{A}$, compute an interpolant $\Gamma[x_1]$ (wrt to $\models_{\mathcal{A}}$).
How to compute $\hat{\phi}_{IR}$ for $\psi$ incrementally

Let $k > 0$, $\hat{\phi}^0 = \varphi_I[x]$

**Base Case)** Let:

\[
\Gamma_1 = \hat{\phi}^0[x_0] \land \varphi_R[x_0, x_1]
\]
\[
\Gamma_2 = \varphi_R[x_1, x_2] \land \cdots \land \varphi_R[x_{k-1}, x_k] \land (\psi[x_1] \lor \cdots \lor \psi[x_k])
\]

If $\Gamma_1 \land \Gamma_2$ is unsatisfiable in $\mathcal{A}$, compute an interpolant $\Gamma[x_1]$ (wrt to $|=\mathcal{A}$).

$\Gamma[x]$ is an adequate over-approximation of $Img(\phi^0)$:

$\Gamma_1 |= \mathcal{A} \Gamma[x_1] \implies$ every state reachable from $\varphi_I$ is in $\Gamma$

$\Gamma \land \Gamma_2 |= \mathcal{A} \bot \implies$ no state in $\Gamma$ leads to $\psi$ within $k$ steps.
How to compute $\hat{\varphi}_{IR}$ for $\psi$ incrementally

Let $k > 0$, $\hat{\varphi}^0 = \varphi_I[x]$

**Base Case** Let:

\[
\Gamma_1 = \hat{\varphi}^0[x_0] \land \varphi_R[x_0, x_1] \\
\Gamma_2 = \varphi_R[x_1, x_2] \land \cdots \land \varphi_R[x_{k-1}, x_k] \land (\psi[x_1] \lor \cdots \lor \psi[x_k])
\]

- If $\Gamma_1 \land \Gamma_2$ is unsatisfiable in $\mathcal{A}$, compute an interpolant $\Gamma[x_1]$ (wrt to $|=\mathcal{A}$).

- $\Gamma[x]$ is an adequate over-approximation of $Img(\varphi^0)$:
  - $\Gamma_1 |=_{\mathcal{A}} \Gamma[x_1] \implies$ every state reachable from $\varphi_I$ is in $\Gamma$
  - $\Gamma \land \Gamma_2 |=_{\mathcal{A}} \bot \implies$ no state in $\Gamma$ leads to $\psi$ within $k$ steps

- Set $\hat{\varphi}^1 = \hat{\varphi}^0[x] \lor \Gamma[x]$
How to compute $\hat{\varphi}_{IR}$ for $\psi$ incrementally

Assume we have computed $\hat{\varphi}^i$ for $i > 0$.

**Step case** Let

\[
\Gamma_1 = \hat{\varphi}^i[x_0] \land \varphi_R[x_0, x_1]
\]
\[
\Gamma_2 = \varphi_R[x_1, x_2] \land \cdots \land \varphi_R[x_{k-1}, x_k] \land (\psi[x_1] \lor \cdots \lor \psi[x_k])
\]
How to compute $\hat{\varphi}_{IR}$ for $\psi$ incrementally

Assume we have computed $\hat{\varphi}^i$ for $i > 0$.

Step case) Let

$$
\Gamma_1 = \hat{\varphi}^i[x_0] \land \varphi_R[x_0, x_1] \\
\Gamma_2 = \varphi_R[x_1, x_2] \land \cdots \land \varphi_R[x_{k-1}, x_k] \land (\psi[x_1] \lor \cdots \lor \psi[x_k])
$$

- If $\Gamma_1 \land \Gamma_2$ is unsatisfiable in $A$, compute an interpolant $\Gamma$ as before
- Let $\hat{\varphi}^{i+1} = \hat{\varphi}^i[x] \lor \Gamma[x]$
How to compute $\hat{\varphi}_{IR}$ for $\psi$ incrementally

Assume we have computed $\hat{\varphi}^i$ for $i > 0$.

Step case) Let

$$\Gamma_1 = \hat{\varphi}^i[x_0] \land \varphi_R[x_0, x_1]$$
$$\Gamma_2 = \varphi_R[x_1, x_2] \land \cdots \land \varphi_R[x_{k-1}, x_k] \land (\psi[x_1] \lor \cdots \lor \psi[x_k])$$

If $\Gamma_1 \land \Gamma_2$ is satisfiable in $A$, $\psi$ is reachable from $\varphi_I$ in $i + 1$ to $i + k$ steps in the overapproximated closure of $\varphi_R$

So, the satisfying paths of states might not be paths in the original system $\mathcal{M}$.
**How to compute \( \hat{\varphi}_{IR} \) for \( \psi \) incrementally**

Assume we have computed \( \hat{\varphi}^i \) for \( i > 0 \).

**Step case** Let

\[
\Gamma_1 = \hat{\varphi}^i[x_0] \land \varphi_R[x_0, x_1]
\]

\[
\Gamma_2 = \varphi_R[x_1, x_2] \land \cdots \land \varphi_R[x_{k-1}, x_k] \land (\psi[x_1] \lor \cdots \lor \psi[x_k])
\]

- If \( \Gamma_1 \land \Gamma_2 \) is satisfiable in \( \mathcal{A} \), \( \psi \) is reachable from \( \varphi_I \) in \( i + 1 \) to \( i + k \) steps in the overapproximated closure of \( \varphi_R \).

So, the satisfying paths of states might not be paths in the original system \( \mathcal{M} \).

- Then, increase \( k \) by 1 and restart the whole process.
Thank you