Software Model-Checking and Cyclic Proof Search

Takeshi Tsukada Chiba University

Based on joint work with Hiroshi Unno published in POPL 2022 and PLDI 2023

Gothenburg Cyclothon, Sep 25th 2025

This talk

Relationship between **software model checking** and **cyclic proof search**

[Ball+ 2001] [Birgmeier+ 2014]
[Cimatti&Griggio 2012] [Cimatti+ 2014]
[Henzinger+ 2004, 2002] [Hoder&Bjørner 2012]
[Komuravelli+ 2014, 2013] [McMillan 2006] …

[Brotherston and Simpson 2011]
[Sprenger and Dam 2003] …

Known **Model-checking problem** $\leftrightarrow$ **Validity/(un)satisfiability problem**

New

Model-checking algorithms

= proof search heuristics

(**Internal states of algorithms** = **partially constructed proofs**)

Our aim from the viewpoint of software model-checking

Providing a **unified account for model-checking algorithms** in terms of **logic**

- To **understand** behaviours of many algorithms using a **single common structure**

  partially constructed proofs

- To **compare** different algorithms
  - Property-directed reachability ≈ Efficient game solving algorithm

    [Bradley 2011] [Een+ 2011] [Cimatti&Griggio 2012] [Farzan&Kincaid 2017]

- To **develop** new algorithms
  - Refutationally complete variant of PDR

Our aim from the viewpoint of cyclic proof search

Importing ideas and techniques of software model-checking to cyclic proof search

- **Finding an appropriate cut formula is crucial** for cyclic proof search
  - **Cut-elimination fails** for cyclic proof systems [Kimura+ 2020] [Masuoka&Tatsuta 2021]

- Software model-checking community has developed
  highly-efficient algorithms to find an appropriate cut formula
  - Existing proof search strategies for cyclic proof system ≈ bounded model-checking + covering
    E.g. [Tellez and Brotherston 2020]

Outline

- **Background**
    - **Software model-checking**
    - Proof systems for inductive definitions
- Key observation
- Software model-checking as cyclic proof search

Software model-checking

Algorithmic analysis of programs to prove properties of their executions

[Jhara&Majumdar 2009]

Let us focus on **safety verification of a while program**

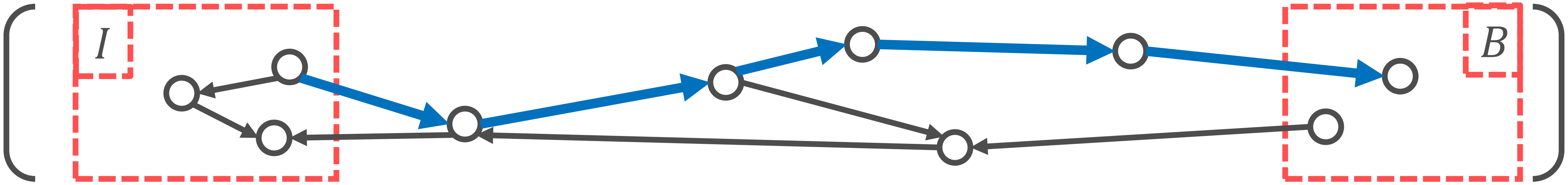
Input	Set of **states**	D
	Initial states	$I \subseteq D$
	Bad states	$B \subseteq D$
	Transition relation	$T \subseteq D \times D$

Usually **infinite**, e.g. $D = \mathbb{Z}^n$

Output Whether B is **unreachable** from I via T

- $\neg \exists s_0 s_1 \ldots s_n \in S. I(s_0) \wedge T(s_0, s_1) \wedge \cdots \wedge T(s_{n-1}, s_n) \wedge B(s_n)$

$\neg$



Inductive invariant

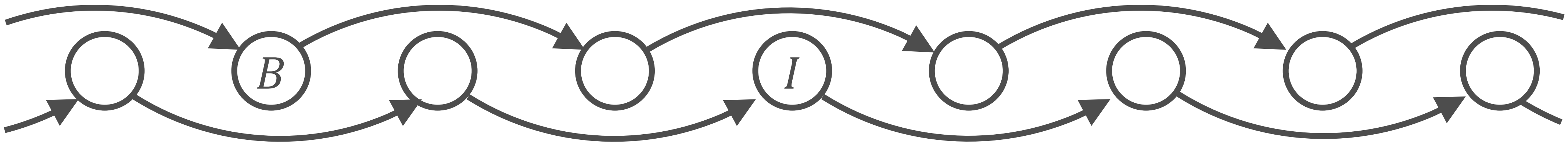
Set of states	D
Initial states	$I \subseteq D$
Bad states	$B \subseteq D$
Transition relation	$T \subseteq D \times D$

A witness of the safety of a given system

Def A subset $P \subseteq D$ is an **inductive invariant** if

- all initial states are P $\qquad I(x) \Longrightarrow P(x)$
- P contains no bad state $\qquad P(x) \Longrightarrow \neg B(x)$
- P is closed under the transition relation $\qquad P(x) \wedge T(x, y) \Longrightarrow P(y)$

Example $\quad D = \mathbb{Z},\ I = \{0\},\ B = \{-3\},\ T = \{\ (n, n+2) \mid n \in \mathbb{Z}\ \}$



- $P_1 = \{\ 2n \mid n \in \mathbb{Z}, n \geq 0\ \}$ is an inductive invariant
- $P_2 = \{\ n \in \mathbb{Z} \mid n \geq 0\ \}$ is an inductive invariant

Inductive invariant

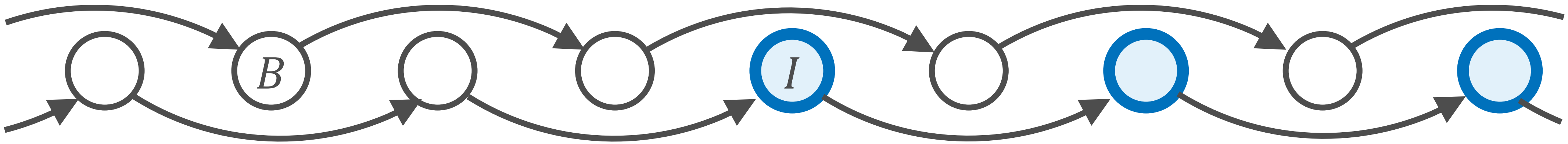
Set of states	D
Initial states	$I \subseteq D$
Bad states	$B \subseteq D$
Transition relation	$T \subseteq D \times D$

A witness of the safety of a given system

Def A subset $P \subseteq D$ is an **inductive invariant** if

- all initial states are P $\qquad I(x) \Longrightarrow P(x)$
- P contains no bad state $\qquad P(x) \Longrightarrow \neg B(x)$
- P is closed under the transition relation $\qquad P(x) \wedge T(x,y) \Longrightarrow P(y)$

Example $\quad D = \mathbb{Z},\ I = \{0\},\ B = \{-3\},\ T = \{\ (n, n+2) \mid n \in \mathbb{Z}\ \}$



- $\boldsymbol{P_1 = \{\ 2n \mid n \in \mathbb{Z}, n \geq 0\ \}}$ **is an inductive invariant**
- $P_2 = \{\ n \in \mathbb{Z} \mid n \geq 0\ \}$ is an inductive invariant

Inductive invariant

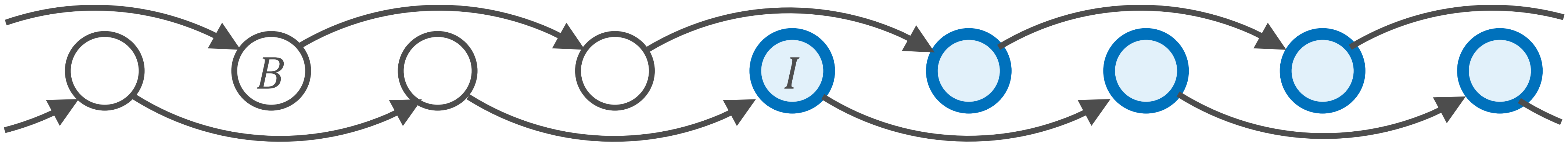
Set of states	D
Initial states	$I \subseteq D$
Bad states	$B \subseteq D$
Transition relation	$T \subseteq D \times D$

A witness of the safety of a given system

Def A subset $P \subseteq D$ is an **inductive invariant** if

- all initial states are P $\qquad I(x) \Longrightarrow P(x)$
- P contains no bad state $\qquad P(x) \Longrightarrow \neg B(x)$
- P is closed under the transition relation $\qquad P(x) \wedge T(x, y) \Longrightarrow P(y)$

Example $D = \mathbb{Z}$, $I = \{0\}$, $B = \{-3\}$, $T = \{\, (n, n + 2) \mid n \in \mathbb{Z} \,\}$



- $P_1 = \{\ 2n \mid n \in \mathbb{Z}, n \geq 0\ \}$ is an inductive invariant
- $\boldsymbol{P_2 = \{\ n \in \mathbb{Z} \mid n \geq 0\ \}}$ **is an inductive invariant**

Inductive invariant

Set of states	D
Initial states	$I \subseteq D$
Bad states	$B \subseteq D$
Transition relation	$T \subseteq D \times D$

A witness of the safety of a given system

Def A subset $P \subseteq D$ is an **inductive invariant** if

- all initial states are P $\quad I(x) \Longrightarrow P(x)$
- P contains no bad state $\quad P(x) \Longrightarrow \neg B(x)$
- P is closed under the transition relation $\quad P(x) \wedge T(x, y) \Longrightarrow P(y)$

Prop If an inductive invariant $P \subseteq D$ exists, the system never reaches a bad state

Model-checkers search for inductive invariants in a variety of cleaver ways

- It is **relatively easy to check** if a given $P \subseteq D$ is indeed an inductive invariant

Outline

- **Background**
  - Software model-checking
  - **Proof systems for inductive definitions**
- Key observation
- Software model-checking as cyclic proof search

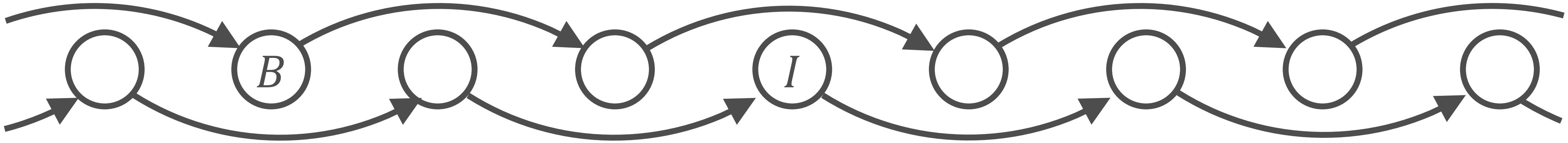
A Logical Formalisation

The **set of reachable states** is the **least solution μR** for P in

$$P(x) \quad \stackrel{\mu}{\Longleftrightarrow} \quad I(x) \vee \left(\exists y. P(y) \wedge T(y, x)\right)$$

- Defining a property as the least solution of an equation = **inductive definition**

Example $D = \mathbb{Z}, I = \{0\}, T = \{\, (n, n+2) \mid n \in \mathbb{Z} \,\}$



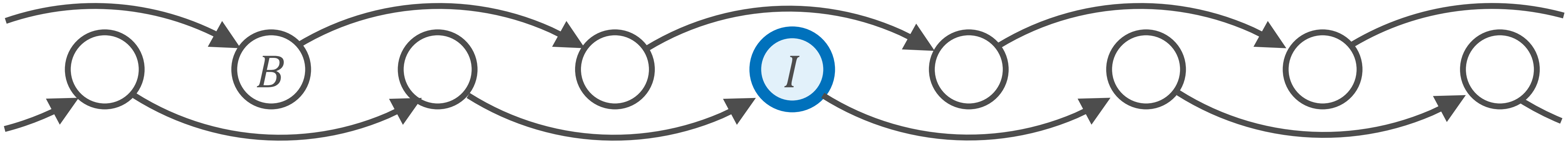
A Logical Formalisation

The **set of reachable states** is the **least solution μR** for P in

$$P(x) \quad \stackrel{\mu}{\Longleftrightarrow} \quad I(x) \vee \big(\exists y. P(y) \wedge T(y,x)\big)$$

- Defining a property as the least solution of an equation = **inductive definition**

Example $D = \mathbb{Z}$, $I = \{0\}$, $T = \{\, (n, n+2) \mid n \in \mathbb{Z} \,\}$



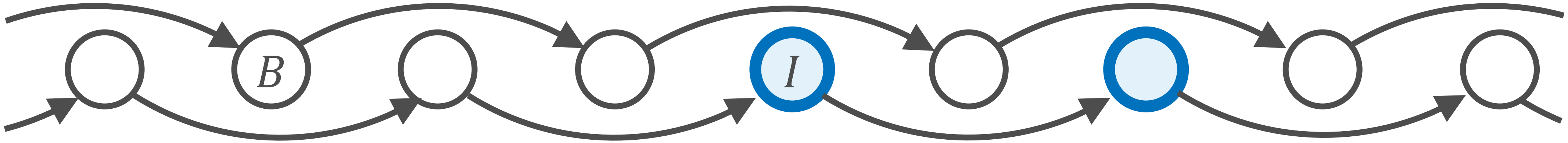
A Logical Formalisation

The **set of reachable states** is the **least solution μR** for P in

$$P(x) \quad \overset{\mu}{\Longleftrightarrow} \quad I(x) \lor \left(\exists y. P(y) \land T(y, x)\right)$$

- Defining a property as the least solution of an equation = **inductive definition**

<u>**Example**</u> $D = \mathbb{Z}$, $I = \{0\}$, $T = \{\, (n, n+2) \mid n \in \mathbb{Z} \,\}$



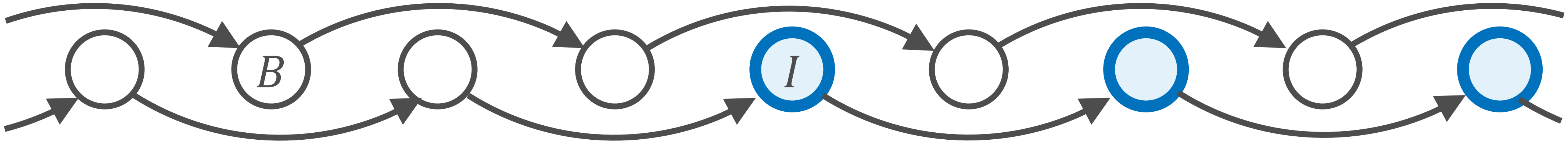
A Logical Formalisation

The **set of reachable states** is the **least solution μR** for P in

$$P(x) \quad \stackrel{\mu}{\Longleftrightarrow} \quad I(x) \vee \big(\exists y. P(y) \wedge T(y,x)\big)$$

- Defining a property as the least solution of an equation = **inductive definition**

Example $D = \mathbb{Z}$, $I = \{0\}$, $T = \{\,(n, n+2) \mid n \in \mathbb{Z}\,\}$



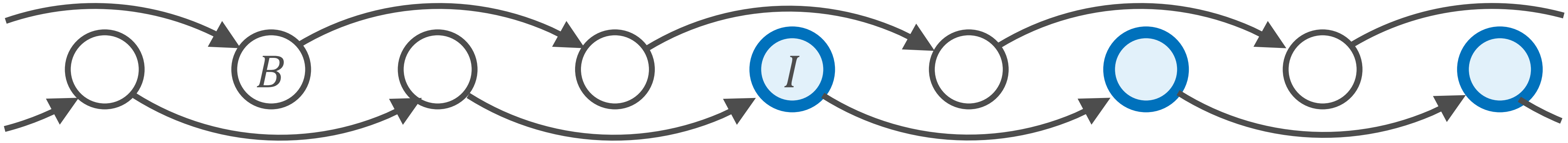
A Logical Formalisation

The **set of reachable states** is the **least solution μR** for P in

$$P(x) \quad \stackrel{\mu}{\Longleftrightarrow} \quad I(x) \vee \big(\exists y. P(y) \wedge T(y, x)\big)$$

- Defining a property as the least solution of an equation = **inductive definition**

Example $D = \mathbb{Z}$, $I = \{0\}$, $T = \{\, (n, n+2) \mid n \in \mathbb{Z} \,\}$



$$\mu R = \{2n \mid n \in \mathbb{Z}, n \geq 0\}$$

A Logical Formalisation

The **set of reachable states** is the **least solution μR** for P in

$$P(x) \quad \overset{\mu}{\Longleftrightarrow} \quad I(x) \vee \big(\exists y. P(y) \wedge T(y, x)\big)$$

- Defining a property as the least solution of an equation = **inductive definition**

<u>**Prop**</u> The system never reaches a bad state if and only if **$\mu R(x) \vdash \neg B(x)$ is valid**

- Simply because μR is the set of reachable states

Proof systems for inductive definitions are usable to prove $\mu R(x) \vdash \neg B(x)$

Classical proof rule for inductive definitions

Due to Martin-Löf (1972)

$$\boxed{\mu R(x) \overset{\mu}{\iff} I(x) \lor (\exists y. \mu R(y) \land T(y,x))}$$

$$\frac{I(x) \lor (\exists y. \varphi(y) \land T(y,x)) \vdash \varphi(x) \qquad \varphi(x) \vdash \neg B(x)}{\mu R(x) \vdash \neg B(x)}$$

The premises require that **$\varphi(x)$ is an inductive invariant**

- Initial states satisfy φ — $I(x) \vdash \varphi(x)$
- φ is closed under the transition — $\exists y. \varphi(y) \land T(y,x) \vdash \varphi(x)$
- φ has no bad state — $\varphi(x) \vdash \neg B(x)$

This rule cannot be used to describe processes searching for inductive invariants

- This rule is **applicable only after an inductive invariant φ is found**

Cyclic proof system

[Brotherston&Simpson 2011]
[Sprenger&Dam 2003] …

A proof system in which **proofs may have cycles**

- **Cycle ≈ use of induction hypothesis**

$$
\dfrac{
  \dfrac{
    \dfrac{\vdots}{I(x) \vdash \varphi(x)}
    \qquad
    \dfrac{
      \dfrac{
        \boxed{\mu R(y) \vdash \varphi(y)}
        \qquad
        \dfrac{\vdots}{\varphi(y) \vdash T(y,x) \Rightarrow \varphi(x)}
      }{\mu R(y) \vdash T(y,x) \Rightarrow \varphi(x)}
    }{\exists y.\mu R(y) \wedge T(y,x) \vdash \varphi(x)}
  }{
    \dfrac{I(x) \vee (\exists y.\mu R(y) \wedge T(y,x)) \vdash \varphi(x)}{\mu R(x) \vdash \varphi(x)}
  }
  \qquad
  \dfrac{\vdots}{\varphi(x) \vdash \neg B(x)}
}{\mu R(x) \vdash \neg B(x)}
$$

A rule for inductive definition just expands the definition

- **Applicable without knowing an inductive invariant**

$$\dfrac{I(x) \vee (\exists y.\mu R(y) \wedge T(y,x)) \vdash \varphi(x)}{\mu R(x) \vdash \varphi(x)}$$

$$\boxed{\mu R(x) \overset{\mu}{\Longleftrightarrow} I(x) \vee (\exists y.\mu R(y) \wedge T(y,x))}$$

Outline

- Background
    - Software model-checking
    - Proof systems for inductive definitions
- **Key observation**
- Software model-checking as cyclic proof search

Key observation

To establish a precise connection between model-checking and proof search,

- **"all reachable states are not bad"** is inappropriate,

$$\mu R(x) \vdash \neg B(x) \qquad\qquad \mu R(x) \overset{\mu}{\iff} I(x) \vee \big(\exists y. \mu R(y) \wedge T(y, x)\big)$$

  - A state x is **reachable** if $\exists y_0 y_1 \dots y_{n-1}. I(y_0) \wedge T(y_0, y_1) \wedge \cdots \wedge T(y_{n-1}, x)$
    (cf. **strongest post-condition**, **backward reachability checking**)

- but the **dual** formalisation **"all initial states are safe"** should be used

$$I(x) \vdash \nu S(x) \qquad\qquad \nu S(x) \overset{\nu}{\iff} \neg B(x) \wedge \big(\forall y. T(x, y) \Rightarrow S(y)\big)$$

greatest solution

  - A state x is **safe** if $\neg\exists y_1 \dots y_n. T(x, y_1) \wedge \cdots \wedge T(y_{n-1}, y_n) \wedge B(y_n)$
    (cf. **weakest pre-condition**, **forward reachability checking**)

Outline

- Background
  - Software model-checking
  - Proof systems for inductive definitions
- Key observation
- **Software model-checking as cyclic proof search**

Goal-oriented proof search

A bottom-up proof-search

- An intermediate state is a **proof with unproved leaves**

1. Start from the tree consisting only of the **goal sequent**

$$I(x) \overset{?}{\vdash} \nu S(x)$$

2. Choose an unproved leaf and **select an appropriate proof rule** for it

$$\frac{I(x) \overset{?}{\vdash} \neg B(x) \wedge \left(\forall y. T(x,y) \Rightarrow \nu S(y)\right)}{I(x) \vdash \nu S(x)}$$

3. Iterate this process until there are no unproved leaves

Symbolic execution

<u>**Heuristic 1**</u> Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

$$\frac{I(x) \overset{?}{\vdash} \neg B(x) \wedge \big(\forall y. T(x,y) \Rightarrow \nu S(y)\big)}{I(x) \vdash \nu S(x)}$$

Symbolic execution

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

$$\dfrac{\dfrac{I(x) \overset{?}{\vdash} \neg B(x) \qquad\qquad I(x) \overset{?}{\vdash} \forall y. T(x,y) \Rightarrow \nu S(y)}{I(x) \vdash \neg B(x) \wedge \big(\forall y. T(x,y) \Rightarrow \nu S(y)\big)}}{I(x) \vdash \nu S(x)}$$

Symbolic execution

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

An SMT solver can automatically (dis)prove

$$\dfrac{\dfrac{I(x) \overset{?}{\vdash} \neg B(x) \qquad\qquad I(x) \overset{?}{\vdash} \forall y. T(x,y) \Rightarrow \nu S(y)}{I(x) \vdash \neg B(x) \land \big(\forall y. T(x,y) \Rightarrow \nu S(y)\big)}}{I(x) \vdash \nu S(x)}$$

Symbolic execution

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

$$\dfrac{\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \overset{\textbf{?}}{I(x) \vdash \forall y. T(x,y) \Rightarrow \nu S(y)}}{I(x) \vdash \neg B(x) \wedge \big(\forall y. T(x,y) \Rightarrow \nu S(y)\big)}}{I(x) \vdash \nu S(x)}$$

Symbolic execution

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

$$\dfrac{\dfrac{\begin{array}{c}\text{SMT}\\ I(x) \vdash \neg B(x)\end{array} \qquad \dfrac{I(x) \overset{?}{\vdash} T(x,y) \Rightarrow \nu S(y)}{I(x) \vdash \forall y.T(x,y) \Rightarrow \nu S(y)}}{I(x) \vdash \neg B(x) \wedge \left(\forall y.T(x,y) \Rightarrow \nu S(y)\right)}}{I(x) \vdash \nu S(x)}$$

Symbolic execution

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

$$\dfrac{\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \dfrac{\dfrac{I(x), T(x,y) \overset{?}{\vdash} \nu S(y)}{I(x) \vdash T(x,y) \Rightarrow \nu S(y)}}{I(x) \vdash \forall y. T(x,y) \Rightarrow \nu S(y)}}{I(x) \vdash \neg B(x) \wedge \big(\forall y. T(x,y) \Rightarrow \nu S(y)\big)}}{I(x) \vdash \nu S(x)}$$

Symbolic execution

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

$$\dfrac{\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \dfrac{\dfrac{\dfrac{\overset{?}{\exists x. I(x) \wedge T(x,y) \vdash \nu S(y)}}{I(x), T(x,y) \vdash \nu S(y)}}{I(x) \vdash T(x,y) \Rightarrow \nu S(y)}}{I(x) \vdash \forall y. T(x,y) \Rightarrow \nu S(y)}}{I(x) \vdash \neg B(x) \wedge \big(\forall y. T(x,y) \Rightarrow \nu S(y)\big)}}{I(x) \vdash \nu S(x)}$$

Symbolic execution

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

One-step transition

$$
\dfrac{
\begin{array}{c}\text{SMT}\\ I(x) \vdash \neg B(x)\end{array}
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\boxed{\exists x. I(x) \wedge T(x,y)}^{\,?} \vdash \nu S(y)}{I(x), T(x,y) \vdash \nu S(y)}}{I(x) \vdash T(x,y) \Rightarrow \nu S(y)}}{I(x) \vdash \forall y. T(x,y) \Rightarrow \nu S(y)}}
{\dfrac{I(x) \vdash \neg B(x) \wedge \left(\forall y. T(x,y) \Rightarrow \nu S(y)\right)}{\boxed{I(x)} \vdash \nu S(x)}}
$$

Symbolic execution

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

Next states of I are safe

Initial states are not bad

One-step transition

?

$$\dfrac{\dfrac{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \dfrac{\dfrac{\dfrac{\boxed{\exists x. I(x) \wedge T(x,y)} \vdash \nu S(y)}{I(x), T(x,y) \vdash \nu S(y)}}{I(x) \vdash T(x,y) \Rightarrow \nu S(y)}}{I(x) \vdash \forall y. T(x,y) \Rightarrow \nu S(y)}}{\dfrac{I(x) \vdash \neg B(x) \wedge (\forall y. T(x,y) \Rightarrow \nu S(y))}{\boxed{I(x)} \vdash \nu S(x)}}$$

Symbolic execution

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

Next states of φ are safe

Current states are not bad

One-step transition

$$\dfrac{\dfrac{\mathrm{SMT}}{\varphi(x) \vdash \neg B(x)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\overset{?}{\boxed{\exists x.\varphi(x) \land T(x,y)}} \vdash \nu S(y)}{\varphi(x), T(x,y) \vdash \nu S(y)}}{\varphi(x) \vdash T(x,y) \Rightarrow \nu S(y)}}{\varphi(x) \vdash \forall y.T(x,y) \Rightarrow \nu S(y)}}{\dfrac{\varphi(x) \vdash \neg B(x) \land (\forall y.T(x,y) \Rightarrow \nu S(y))}{\boxed{\varphi(x)} \vdash \nu S(x)}}$$

Generalise to arbitrary set φ

Symbolic execution

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

$$\dfrac{\dfrac{\text{SMT}}{\varphi(x) \vdash \neg B(x)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\exists x.\varphi(x) \land T(x,y) \overset{?}{\vdash} \nu S(y)}{\varphi(x), T(x,y) \vdash \nu S(y)}}{\varphi(x) \vdash T(x,y) \Rightarrow \nu S(y)}}{\varphi(x) \vdash \forall y.T(x,y) \Rightarrow \nu S(y)}}{\dfrac{\varphi(x) \vdash \neg B(x) \land \left(\forall y.T(x,y) \Rightarrow \nu S(y)\right)}{\varphi(x) \vdash \nu S(x)}}$$

A derived rule: $$\frac{\varphi(x) \vdash \neg B(x) \qquad \exists x.\varphi(x) \land T(x,y) \vdash \nu S(y)}{\varphi(x) \vdash \nu S(x)}\ (\textsc{SymbolicExecution})$$

Symbolic execution

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

$$
\dfrac{
\dfrac{
\overset{\text{SMT}}{\varphi(x) \vdash \neg B(x)}
\qquad
\dfrac{
\dfrac{
\dfrac{
\exists x.\varphi(x) \wedge T(x,y) \overset{?}{\vdash} \nu S(y)
}{
\overline{\varphi(x), T(x,y) \vdash \nu S(y)}
}
}{
\varphi(x) \vdash T(x,y) \Rightarrow \nu S(y)
}
}{
\varphi(x) \vdash \forall y.T(x,y) \Rightarrow \nu S(y)
}
}{
\varphi(x) \vdash \neg B(x) \wedge \big(\forall y.T(x,y) \Rightarrow \nu S(y)\big)
}
}{
\varphi(x) \vdash \nu S(x)
}
$$

A derived rule:

$$
\dfrac{\varphi(x) \vdash \neg B(x) \qquad \exists x.\varphi(x) \wedge T(x,y) \vdash \nu S(y)}{\varphi(x) \vdash \nu S(x)}\,(\mathrm{SE})
$$

Bounded model-checking [Biere+ 1999]

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

***k*-th iteration of (SE)** rule coincides with **model-checking within *k* steps**

$$\dfrac{\overset{\text{SMT}}{\varphi_k(x) \vdash \neg B(x)} \qquad \overset{\textbf{?}}{\varphi_{k+1}(x) \vdash \nu S(x)}}{\varphi_k(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

$$\vdots$$

$$\dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \vdots}{\varphi_2(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

$$\dfrac{\overset{\text{SMT}}{\varphi_1(x) \vdash \neg B(x)} \qquad \varphi_2(x) \vdash \nu S(x)}{\varphi_1(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

$$\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \varphi_1(x) \vdash \nu S(x)}{I(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

Forward criterion [Sheeran+ 2000]

<u>Heuristic 1</u> Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

Trying to make cycles after k-th iteration of (SE) rule

$$\dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \varphi_{k+1}(x) \overset{?}{\vdash} \nu S(x)}{\varphi_k(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

$$\vdots$$

$$\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \dfrac{\overset{\text{SMT}}{\varphi_1(x) \vdash \neg B(x)} \qquad \dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \vdots}{\varphi_2(x) \vdash \nu S(x)}\ (\mathrm{SE})}{\varphi_1(x) \vdash \nu S(x)}\ (\mathrm{SE})}{I(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

Forward criterion [Sheeran+ 2000]

<u>Heuristic 1</u> Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

Trying to make cycles after k-th iteration of (SE) rule

if $\varphi_{k+1}(x) = \varphi_2(x)$

$$\dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \varphi_{k+1}(x) \overset{?}{\vdash} \nu S(x)}{\varphi_k(x) \vdash \nu S(x)}\ (\text{SE})$$

$$\vdots$$

$$\dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \vdots}{\varphi_2(x) \vdash \nu S(x)}\ (\text{SE})$$

$$\dfrac{\overset{\text{SMT}}{\varphi_1(x) \vdash \neg B(x)} \qquad \varphi_2(x) \vdash \nu S(x)}{\varphi_1(x) \vdash \nu S(x)}\ (\text{SE})$$

$$\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \varphi_1(x) \vdash \nu S(x)}{I(x) \vdash \nu S(x)}\ (\text{SE})$$

Forward criterion [Sheeran+ 2000]

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

Trying to make cycles after k-th iteration of (SE) rule

if $\varphi_{k+1}(x) = \varphi_2(x)$

$$\dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \varphi_{k+1}(x) \vdash \nu S(x)}{\varphi_k(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

$$\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \dfrac{\overset{\text{SMT}}{\varphi_1(x) \vdash \neg B(x)} \qquad \dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \vdots}{\varphi_2(x) \vdash \nu S(x)}\ (\mathrm{SE})}{\varphi_1(x) \vdash \nu S(x)}\ (\mathrm{SE})}{I(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

Forward criterion [Sheeran+ 2000]

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

Trying to make cycles after k-th iteration of (SE) rule

$$\dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \varphi_{k+1}(x) \overset{?}{\vdash} \nu S(x)}{\varphi_k(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

$$\vdots$$

$$\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \dfrac{\overset{\text{SMT}}{\varphi_1(x) \vdash \neg B(x)} \qquad \dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \vdots}{\varphi_2(x) \vdash \nu S(x)}\ (\mathrm{SE})}{\varphi_1(x) \vdash \nu S(x)}\ (\mathrm{SE})}{I(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

Induction hypothesis

Induction hypothesis

Induction hypothesis

Forward criterion [Sheeran+ 2000]

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

Trying to make cycles after k-th iteration of (SE) rule

$$\dfrac{\boxed{\varphi_{k+1}(x) \overset{?}{\vdash} \bigvee_{i=1}^{k} \varphi_i(x)} \qquad ?}{\varphi_{k+1}(x) \vdash \nu S(x)}$$

$$\dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \varphi_{k+1}(x) \vdash \nu S(x)}{\varphi_k(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

Induction hypothesis

$$\vdots$$

$$\dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \vdots}{\varphi_2(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

Induction hypothesis

$$\dfrac{\overset{\text{SMT}}{\varphi_1(x) \vdash \neg B(x)} \qquad \varphi_2(x) \vdash \nu S(x)}{\varphi_1(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

Induction hypothesis

$$\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \varphi_1(x) \vdash \nu S(x)}{I(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

Forward criterion [Sheeran+ 2000]

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

Trying to make cycles after k-th iteration of (SE) rule

$$\dfrac{\boxed{\overset{\text{SMT}}{\varphi_{k+1}(x) \vdash \bigvee_{i=1}^{k} \varphi_i(x)}} \qquad \textbf{?}}{\varphi_{k+1}(x) \vdash \nu S(x)}$$

$$\dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \varphi_{k+1}(x) \vdash \nu S(x)}{\varphi_k(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

Induction hypothesis

$$\vdots$$

$$\dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \vdots}{\varphi_2(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

Induction hypothesis

$$\dfrac{\overset{\text{SMT}}{\varphi_1(x) \vdash \neg B(x)} \qquad \varphi_2(x) \vdash \nu S(x)}{\varphi_1(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

Induction hypothesis

$$\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \varphi_1(x) \vdash \nu S(x)}{I(x) \vdash \nu S(x)}\ (\mathrm{SE})$$

Forward criterion [Sheeran+ 2000]

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

Trying to make cycles after k-th iteration of (SE) rule

$$
\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \dfrac{\overset{\text{SMT}}{\varphi_1(x) \vdash \neg B(x)} \qquad \dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \dfrac{\dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \dfrac{\boxed{\overset{\text{SMT}}{\varphi_{k+1}(x) \vdash \bigvee_{i=1}^{k} \varphi_i(x)}} \qquad \bigvee_{i=1}^{k} \varphi_i(x) \overset{?}{\vdash} \nu S(x)}{\varphi_{k+1}(x) \vdash \nu S(x)}\,(\textsc{Cut})}{\varphi_k(x) \vdash \nu S(x)}\,(\mathrm{SE})}{\vdots}}{\varphi_2(x) \vdash \nu S(x)}\,(\mathrm{SE})}{\varphi_1(x) \vdash \nu S(x)}\,(\mathrm{SE})}{I(x) \vdash \nu S(x)}\,(\mathrm{SE})
$$

Induction hypothesis

Induction hypothesis

Induction hypothesis

Forward criterion [Sheeran+ 2000]

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

Trying to make cycles after k-th iteration of (SE) rule

$$
\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad
\dfrac{\overset{\text{SMT}}{\varphi_1(x) \vdash \neg B(x)} \qquad
\dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad
\begin{array}{c}
\dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad
\dfrac{\boxed{\overset{\text{SMT}}{\varphi_{k+1}(x) \vdash \bigvee_{i=1}^{k} \varphi_i(x)}} \qquad
\dfrac{\varphi_k(x) \overset{?}{\vdash} \nu S(x) \quad \cdots \quad \varphi_1(x) \overset{?}{\vdash} \nu S(x)}{\bigvee_{i=1}^{k} \varphi_i(x) \vdash \nu S(x)}}{\varphi_{k+1}(x) \vdash \nu S(x)}\,(\textsc{Cut})}{\varphi_k(x) \vdash \nu S(x)}\,(\text{SE}) \\
\vdots
\end{array}}{\varphi_2(x) \vdash \nu S(x)}\,(\text{SE})}{\varphi_1(x) \vdash \nu S(x)}\,(\text{SE})}{I(x) \vdash \nu S(x)}\,(\text{SE})
$$

Induction hypothesis

Induction hypothesis

Induction hypothesis

Forward criterion [Sheeran+ 2000]

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

Trying to make cycles after k-th iteration of (SE) rule

$$
\dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \dfrac{\boxed{\overset{\text{SMT}}{\varphi_{k+1}(x) \vdash \bigvee_{i=1}^{k} \varphi_i(x)}} \qquad \dfrac{\varphi_k(x) \vdash \nu S(x) \quad \dots \quad \varphi_1(x) \overset{?}{\vdash} \nu S(x)}{\bigvee_{i=1}^{k} \varphi_i(x) \vdash \nu S(x)}}{\varphi_{k+1}(x) \vdash \nu S(x)}\,(\textsc{Cut})}{\varphi_k(x) \vdash \nu S(x)}\,(\mathrm{SE})
$$

Induction hypothesis

$$
\vdots
$$

$$
\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \dfrac{\overset{\text{SMT}}{\varphi_1(x) \vdash \neg B(x)} \qquad \dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \vdots}{\varphi_2(x) \vdash \nu S(x)}\,(\mathrm{SE})}{\varphi_1(x) \vdash \nu S(x)}\,(\mathrm{SE})}{I(x) \vdash \nu S(x)}\,(\mathrm{SE})
$$

Induction hypothesis

Induction hypothesis

Forward criterion [Sheeran+ 2000]

Heuristic 1 Try to fit the shape of unproved sequents into the form $\varphi(x) \vdash \nu S(x)$

Trying to make cycles after k-th iteration of (SE) rule

$$
\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \dfrac{\overset{\text{SMT}}{\varphi_1(x) \vdash \neg B(x)} \qquad \dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \vdots}{\varphi_2(x) \vdash \nu S(x)}\,(\mathrm{SE})}{\varphi_1(x) \vdash \nu S(x)}\,(\mathrm{SE})}{I(x) \vdash \nu S(x)}\,(\mathrm{SE})
$$

$$
\vdots
$$

$$
\dfrac{\overset{\text{SMT}}{\varphi_2(x) \vdash \neg B(x)} \qquad \dfrac{\boxed{\overset{\text{SMT}}{\varphi_{k+1}(x) \vdash \bigvee_{i=1}^{k} \varphi_i(x)}} \qquad \dfrac{\varphi_k(x) \vdash \nu S(x) \quad \cdots \quad \varphi_1(x) \vdash \nu S(x)}{\bigvee_{i=1}^{k} \varphi_i(x) \vdash \nu S(x)}}{\varphi_{k+1}(x) \vdash \nu S(x)}\,(\textsc{Cut})}{\varphi_k(x) \vdash \nu S(x)}\,(\mathrm{SE})
$$

Induction hypothesis

Induction hypothesis

Induction hypothesis

More aggressive use of (Cut)

$$\dfrac{\varphi(x) \vdash \neg B(x) \qquad \exists x.\varphi(x) \wedge T(x,y) \vdash \psi(y) \qquad \psi(y) \vdash \nu S(y)}{\varphi(x) \vdash \nu S(x)}\,(\mathrm{SE{+}C{\scriptsize UT}})$$

$$\left(\dfrac{\varphi(x) \vdash \neg B(x) \qquad \dfrac{\exists x.\varphi(x) \wedge T(x,y) \vdash \psi(y) \qquad \psi(y) \vdash \nu S(y)}{\exists x.\varphi(x) \wedge T(x,y) \vdash \nu S(y)}\,(\mathrm{C{\scriptsize UT}})}{\varphi(x) \vdash \nu S(x)}\,(\mathrm{SE})\right)$$

Question **How to select the cut formula ψ?**

Let Ξ be a finite set of formulas (closed under certain logical operations)

Heuristic 2 Let the cut formula be the **strongest** $\psi \in \Xi$ s.t. $\exists x.\, \varphi(x) \wedge T(x, y) \vdash \psi(y)$

⇒ **Predicate abstraction** [Ball+2001] [Graf&Saïdi 1997]

IMPACT [McMillan 2006]

Heuristic 3 Tentatively choose $\top$ **as the cut formula**

Heuristic 4 **When the proof attempt fails, strengthen the cut formulas** as follows

- **Replace cut formula φ_i with $\varphi_i \land Q_i$** and **solve the constraints on Q_i**

$$I(x) \overset{?}{\vdash} \nu S(x)$$

IMPACT [McMillan 2006]

Heuristic 3 Tentatively choose ⊤ **as the cut formula**

Heuristic 4 **When the proof attempt fails, strengthen the cut formulas** as follows

- **Replace cut formula φ_i with $\varphi_i \land Q_i$** and **solve the constraints on Q_i**

$$\dfrac{I(x) \overset{?}{\vdash} \neg B(x) \qquad \exists x. I(x) \land T(x,y) \overset{?}{\vdash} \boxed{\top} \qquad \boxed{\top} \overset{?}{\vdash} \nu S(y)}{I(x) \vdash \nu S(x)} \;(\text{SE+Cut})$$

IMPACT [McMillan 2006]

Heuristic 3 Tentatively choose ⊤ **as the cut formula**

Heuristic 4 **When the proof attempt fails, strengthen the cut formulas** as follows

- **Replace cut formula φ_i with $\varphi_i \land Q_i$** and **solve the constraints on Q_i**

$$\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \exists x. I(x) \land T(x,y) \vdash^{?} \boxed{\top} \qquad \boxed{\top} \vdash^{?} \nu S(y)}{I(x) \vdash \nu S(x)}\ (\text{SE+Cut})$$

IMPACT [McMillan 2006]

Heuristic 3 Tentatively choose ⊤ **as the cut formula**

Heuristic 4 **When the proof attempt fails, strengthen the cut formulas** as follows

- **Replace cut formula φ_i with $\varphi_i \wedge Q_i$** and **solve the constraints on Q_i**

$$\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \overset{\text{SMT}}{\exists x. I(x) \wedge T(x,y) \vdash \boxed{\top}} \qquad \boxed{\top} \overset{?}{\vdash} \nu S(y)}{I(x) \vdash \nu S(x)} \;(\text{SE+}\textsc{Cut})$$

IMPACT [McMillan 2006]

Heuristic 3 Tentatively choose $\top$ **as the cut formula**

Heuristic 4 **When the proof attempt fails, strengthen the cut formulas** as follows

- **Replace cut formula φ_i with $\varphi_i \wedge Q_i$** and **solve the constraints on Q_i**

$$\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \overset{\text{SMT}}{\exists x. I(x) \wedge T(x,y) \vdash \top} \qquad \overset{?}{\top \vdash \nu S(y)}}{I(x) \vdash \nu S(x)} \ (\text{SE+Cut})$$

IMPACT [McMillan 2006]

Heuristic 3 Tentatively choose $\top$ **as the cut formula**

Heuristic 4 **When the proof attempt fails, strengthen the cut formulas** as follows

- **Replace cut formula φ_i with $\varphi_i \wedge Q_i$** and **solve the constraints on Q_i**

$$\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \overset{\text{SMT}}{\exists x. I(x) \wedge T(x,y) \vdash \top} \qquad \dfrac{\overset{\textbf{False}}{\top \vdash \neg B(y)} \qquad \cdots\cdots}{\top \vdash \nu S(y)}\,(\mathrm{SE{+}C{\small UT}})}{I(x) \vdash \nu S(x)}\,(\mathrm{SE{+}C{\small UT}})$$

IMPACT [McMillan 2006]

Heuristic 3 Tentatively choose $\top$ **as the cut formula**

Heuristic 4 **When the proof attempt fails, strengthen the cut formulas** as follows

- **Replace cut formula φ_i with $\varphi_i \wedge Q_i$** and **solve the constraints on Q_i**

$$
\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \quad \exists x. I(x) \wedge T(x,y) \overset{?}{\vdash} \boxed{\top \wedge Q_1(y)} \quad \dfrac{\boxed{\top \wedge Q_1(y)} \overset{?}{\vdash} \neg B(y) \quad \cdots\cdots}{\boxed{\top \wedge Q_1(y)} \vdash \nu S(y)}\,(\text{SE+Cut})}{I(x) \vdash \nu S(x)}\,(\text{SE+Cut})
$$

IMPACT [McMillan 2006]

Heuristic 3 Tentatively choose ⊤ **as the cut formula**

Heuristic 4 **When the proof attempt fails, strengthen the cut formulas** as follows

- **Replace cut formula φ_i with $\varphi_i \wedge Q_i$** and **solve the constraints on Q_i**

$$\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \exists x. I(x) \wedge T(x,y) \overset{?}{\vdash} \boxed{\top \wedge Q_1(y)} \qquad \dfrac{\boxed{\top \wedge Q_1(y)} \overset{?}{\vdash} \neg B(y) \quad \cdots\cdots}{\boxed{\top \wedge Q_1(y)} \vdash \nu S(y)}\,(\textsc{SE+Cut})}{I(x) \vdash \nu S(x)}\,(\textsc{SE+Cut})$$

$$\text{Constraints: } \{\exists x. I(x) \wedge T(x,y) \vdash \top \wedge Q_1(y), \quad \top \wedge Q_1(y) \vdash \neg B(y)\}$$

IMPACT [McMillan 2006]

Heuristic 3 Tentatively choose ⊤ **as the cut formula**

Heuristic 4 **When the proof attempt fails, strengthen the cut formulas** as follows

- **Replace cut formula φ_i with $\varphi_i \land Q_i$** and **solve the constraints on Q_i**

$$
\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \exists x. I(x) \land T(x,y) \overset{?}{\vdash} \boxed{\top \land Q_1(y)} \qquad \dfrac{\boxed{\top \land Q_1(y)} \overset{?}{\vdash} \neg B(y) \quad \cdots\cdots}{\boxed{\top \land Q_1(y)} \vdash \nu S(y)}\,(\text{SE+C}\textsc{ut})}{I(x) \vdash \nu S(x)}\,(\text{SE+C}\textsc{ut})
$$

$$
\text{Constraints: } \{\exists x. I(x) \land T(x,y) \vdash \top \land Q_1(y), \quad \top \land Q_1(y) \vdash \neg B(y)\}
$$

A solution of this constraint set is called an **interpolant**

Property-directed reachability

[Bradley 2011] [Een+ 2011] [Cimatti&Griggio 2012] ...

Heuristic 5 In strengthening, **keep cut formulas unchanged as many as possible**

$$\frac{\overset{\textbf{fail}}{\phi_n(x) \vdash \neg B(x)} \qquad \exists x.\varphi_n(x) \wedge T(x,y) \overset{?}{\vdash} \top \qquad \top \overset{?}{\vdash} \nu S(y)}{\varphi_n(y) \vdash \nu S(y)}\ (\text{SE+Cut})$$

$$\frac{\overset{\text{SMT}}{\varphi_{n-1}(x) \vdash \neg B(x)} \qquad \overset{\text{SMT}}{\exists x.\varphi_{n-1}(x) \wedge T(x,y) \vdash \varphi_n(y)} \qquad \varphi_n(y) \vdash \nu S(y)}{\varphi_{n-1}(y) \vdash \nu S(y)}\ (\text{SE+Cut})$$

$$\vdots$$

$$\frac{\overset{\text{SMT}}{\varphi_1(x) \vdash \neg B(x)} \qquad \overset{\text{SMT}}{\exists x.\varphi_1(x) \wedge T(x,y) \vdash \varphi_2(y)} \qquad \varphi_2(y) \vdash \nu S(y)}{\varphi_1(y) \vdash \nu S(y)}\ (\text{SE+Cut})$$

$$\frac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \overset{\text{SMT}}{\exists x.I(x) \wedge T(x,y) \vdash \varphi_1(y)} \qquad \varphi_1(y) \vdash \nu S(y)}{I(x) \vdash \nu S(x)}\ (\text{SE+Cut})$$

Property-directed reachability

[Bradley 2011] [Een+ 2011]
[Cimatti&Griggio 2012] ...

<u>Heuristic 5</u> In strengthening, **keep cut formulas unchanged as many as possible**

fail

?

?

$$\dfrac{\phi_n(x) \vdash \neg B(x) \qquad \exists x.\varphi_n(x) \wedge T(x,y) \vdash \top \qquad \top \vdash \nu S(y)}{\varphi_n(y) \vdash \nu S(y)}\ (\textsc{SE+Cut})$$

SMT

SMT

$$\dfrac{\varphi_{n-1}(x) \vdash \neg B(x) \qquad \exists x.\varphi_{n-1}(x) \wedge T(x,y) \vdash \boxed{\varphi_n(y)} \qquad \boxed{\varphi_n(y)} \vdash \nu S(y)}{\varphi_{n-1}(y) \vdash \nu S(y)}\ (\textsc{SE+Cut})$$

$$\vdots$$

SMT

SMT

$$\dfrac{\varphi_1(x) \vdash \neg B(x) \qquad \exists x.\varphi_1(x) \wedge T(x,y) \vdash \varphi_2(y) \qquad \varphi_2(y) \vdash \nu S(y)}{\varphi_1(y) \vdash \nu S(y)}\ (\textsc{SE+Cut})$$

SMT

SMT

$$\dfrac{I(x) \vdash \neg B(x) \qquad \exists x.I(x) \wedge T(x,y) \vdash \varphi_1(y) \qquad \varphi_1(y) \vdash \nu S(y)}{I(x) \vdash \nu S(x)}\ (\textsc{SE+Cut})$$

Property-directed reachability

[Bradley 2011] [Een+ 2011] [Cimatti&Griggio 2012] …

Heuristic 5 In strengthening, **keep cut formulas unchanged as many as possible**

fail

$$\phi_n(x) \vdash \neg B(x) \qquad \exists x.\varphi_n(x) \wedge T(x,y) \overset{?}{\vdash} \top \qquad \top \overset{?}{\vdash} \nu S(y)$$

(SE+Cut)

SMT

$$\varphi_{n-1}(x) \vdash \neg B(x) \qquad \exists x.\varphi_{n-1}(x) \wedge T(x,y) \vdash \boxed{\varphi_n(y)} \qquad \boxed{\varphi_n(y)} \vdash \nu S(y)$$

SMT

(SE+Cut)

$$\boxed{\varphi_{n-1}(y)} \vdash \nu S(y)$$

$$\vdots$$

SMT

$$\varphi_1(x) \vdash \neg B(x) \qquad \exists x.\varphi_1(x) \wedge T(x,y) \vdash \varphi_2(y) \qquad \varphi_2(y) \vdash \nu S(y)$$

SMT

(SE+Cut)

SMT

$$I(x) \vdash \neg B(x) \qquad \exists x.I(x) \wedge T(x,y) \vdash \varphi_1(y) \qquad \varphi_1(y) \vdash \nu S(y)$$

SMT

(SE+Cut)

$$I(x) \vdash \nu S(x)$$

Property-directed reachability

[Bradley 2011] [Een+ 2011] [Cimatti&Griggio 2012] …

Heuristic 5 In strengthening, **keep cut formulas unchanged as many as possible**

fail

$\phi_n(x) \vdash \neg B(x) \qquad \exists x.\varphi_n(x) \wedge T(x,y) \overset{?}{\vdash} \top \qquad \top \overset{?}{\vdash} \nu S(y)$

(SE+CUT)

$$\varphi_n(y) \vdash \nu S(y)$$

SMT SMT

$\varphi_{n-1}(x) \vdash \neg B(x) \qquad \exists x.\varphi_{n-1}(x) \wedge T(x,y) \vdash \boxed{\varphi_n(y)} \qquad \boxed{\varphi_n(y)} \vdash \nu S(y)$

(SE+CUT)

$$\boxed{\varphi_{n-1}(y)} \vdash \nu S(y)$$

$$\vdots$$

SMT SMT

$\varphi_1(x) \vdash \neg B(x) \qquad \exists x.\varphi_1(x) \wedge T(x,y) \vdash \boxed{\varphi_2(y)} \qquad \boxed{\varphi_2(y)} \vdash \nu S(y)$

(SE+CUT)

SMT SMT

$I(x) \vdash \neg B(x) \qquad \exists x.I(x) \wedge T(x,y) \vdash \varphi_1(y) \qquad \varphi_1(y) \vdash \nu S(y)$

(SE+CUT)

$$I(x) \vdash \nu S(x)$$

Property-directed reachability

[Bradley 2011] [Een+ 2011] [Cimatti&Griggio 2012] …

Heuristic 5 In strengthening, **keep cut formulas unchanged as many as possible**

$$\dfrac{\overset{\textbf{fail}}{\phi_n(x) \vdash \neg B(x)} \qquad \exists x.\varphi_n(x) \wedge T(x,y) \overset{?}{\vdash} \top \qquad \top \overset{?}{\vdash} \nu S(y)}{\boxed{\varphi_n(y)} \vdash \nu S(y)}\ (\mathrm{SE{+}Cut})$$

$$\dfrac{\overset{\text{SMT}}{\varphi_{n-1}(x) \vdash \neg B(x)} \qquad \exists x.\varphi_{n-1}(x) \wedge T(x,y) \overset{\text{SMT}}{\vdash} \boxed{\varphi_n(y)} \qquad \boxed{\varphi_n(y)} \vdash \nu S(y)}{\boxed{\varphi_{n-1}(y)} \vdash \nu S(y)}\ (\mathrm{SE{+}Cut})$$

$$\vdots$$

$$\dfrac{\overset{\text{SMT}}{\varphi_1(x) \vdash \neg B(x)} \qquad \exists x.\varphi_1(x) \wedge T(x,y) \overset{\text{SMT}}{\vdash} \boxed{\varphi_2(y)} \qquad \boxed{\varphi_2(y)} \vdash \nu S(y)}{\boxed{\varphi_1(y)} \vdash \nu S(y)}\ (\mathrm{SE{+}Cut})$$

$$\dfrac{\overset{\text{SMT}}{I(x) \vdash \neg B(x)} \qquad \exists x.I(x) \wedge T(x,y) \overset{\text{SMT}}{\vdash} \boxed{\varphi_1(y)} \qquad \boxed{\varphi_1(y)} \vdash \nu S(y)}{I(x) \vdash \nu S(x)}\ (\mathrm{SE{+}Cut})$$

Property-directed reachability

[Bradley 2011] [Een+ 2011] [Cimatti&Griggio 2012] ...

Heuristic 5 In strengthening, **keep cut formulas unchanged as many as possible**

In terms of constraints, **Heuristic 5** requires us to find a solution σ such that

$$\sigma(Q_1) = \cdots = \sigma(Q_k) = \top$$

for the largest possible k

$$\left\{ \begin{array}{ll} \exists x. I(x) \land T(x,y) \vdash \varphi_1(y) \land Q_1(y), & \varphi_1(y) \land Q_1(y) \vdash \neg B(y) \\ \exists x. \varphi_1(x) \land Q_1(x) \land T(x,y) \vdash \varphi_2(y) \land Q_2(y), & \varphi_2(y) \land Q_2(y) \vdash \neg B(y) \\ \qquad \vdots & \\ \exists x. \varphi_{n-1}(x) \land Q_{n-1}(x) \land T(x,y) \vdash \varphi_n(y) \land Q_n(y), & \varphi_n(y) \land Q_n(y) \vdash \neg B(y) \end{array} \right\}$$

Property-directed reachability

[Bradley 2011] [Een+ 2011] [Cimatti&Griggio 2012] …

Heuristic 5 In strengthening, **keep cut formulas unchanged as many as possible**

Property-directed reachability

[Bradley 2011] [Een+ 2011] [Cimatti&Griggio 2012] …

Heuristic 5 In strengthening, **keep cut formulas unchanged as many as possible**

≈ Keep as many parts as possible unchanged

Property-directed reachability [Bradley 2011] [Een+ 2011] [Cimatti&Griggio 2012] ...

Heuristic 5 In strengthening, **keep cut formulas unchanged as many as possible**

maximally conservative

≈ Keep as many parts as possible unchanged

Property-directed reachability [Bradley 2011] [Een+ 2011] [Cimatti&Griggio 2012] …

Heuristic 5 In strengthening, **keep cut formulas unchanged as many as possible**

maximally conservative

≈ Keep as many parts as possible unchanged

Similar ideas can be found in

- a procedure for game solving [Farzan&Kincaid 2017]
- **Spacer**, a state-of-the-art solver for non-linear CHCs [Komuravelli+, 2014]

Game solving [Farzan&Kincaid 2017]

Heuristic 5' In strengthening, **keep as many parts as possible unchanged**

They developed a validity checker for **first-order real arithmetic with ν**

- Corresponding to possibly infinite games with trivial condition

To prove $\vdash \nu X.\lambda x.\phi$, it constructs proofs of approximations $\vdash \nu^{(n)} X.\lambda x.\phi$ for $n = 1,2,\ldots$

- where $\nu^{(0)} X.\lambda x.\phi \coloneqq \top$ and $\nu^{(n+1)} X.\lambda x.\phi \coloneqq \phi[X \mapsto \nu^{(n)} X.\lambda x.\phi]$
- a proof of $\vdash \nu^{(n)} X.\lambda x.\phi$ can be seen as a partial proof of $\vdash \nu X.\lambda x.\phi$

The proof of $\vdash \nu^{(n+1)} X.\lambda x.\phi$ is adapted from the proof of Π of $\vdash \nu^{(n)} X.\lambda x.\phi$

- Replace every $\psi_0 \vdash \nu^{(0)} X.\lambda x.\phi$ in Π with a proof of $\psi_0 \vdash \nu^{(1)} X.\lambda x.\phi$
- If it fails, replace every $\psi_1 \vdash \nu^{(1)} X.\lambda x.\phi$ in Π with a proof of $\psi_1 \vdash \nu^{(2)} X.\lambda x.\phi$
- If it fails, ...

Spacer [Komuravelli+, 2014]

Heuristic 5' In strengthening, **keep as many parts as possible unchanged**

A solver for **non-linear CHCs**

$$\{I(x) \implies P(x), \quad P(x) \land P(y) \land T(x,y,z) \implies P(z), \quad P(z) \implies \neg B(z)\}$$

It has a solution iff $\left(\mu X. \lambda z. I(z) \lor \left(\exists xy. X(x) \land X(y) \land T(x,y,z)\right)\right)(z) \vDash \neg B(z)$

It constructs proofs of approximations $(\mu^{(n)} X. \ldots)(z) \vdash \neg B(z)$ for $n = 1,2, \ldots$

The proof of $(\mu^{(n+1)} X. \ldots)(z) \vdash \neg B(z)$ is adapted from $(\mu^{(n)} X. \ldots)(z) \vdash \neg B(z)$

- Construct the following proof, and try to strengthen the conclusion to $\ldots \vdash \neg B(z)$ by the "**smallest**" change

$$\dfrac{\begin{array}{c}\vdots\\ (\mu^{(n)} X. \ldots)(x) \vdash \neg B(x)\end{array} \qquad \begin{array}{c}\vdots\\ (\mu^{(n)} X. \ldots)(y) \vdash \neg B(y)\end{array} \qquad \neg B(x) \land \neg B(y) \land T(x,y,z) \vdash \top}{(\mu^{(n+1)} X. \ldots)(x) \vdash \top}$$

Spacer [Komuravelli+, 2014]

<u>Heuristic 5'</u> In strengthening, **keep as many parts as possible unchanged**

A solver for **non-linear CHCs**

$$\{I(x) \Longrightarrow P(x), \quad P(x) \wedge P(y) \wedge T(x,y,z) \Longrightarrow P(z), \quad P(z) \Longrightarrow \neg B(z)\}$$

It has a solution iff $\left(\mu X.\lambda z.I(z) \vee \left(\exists xy.X(x) \wedge X(y) \wedge T(x,y,z)\right)\right)(z) \vDash \neg B(z)$

It constructs proofs of approximations $(\mu^{(n)}X. \dots)(z) \vdash \neg B(z)$ for $n = 1,2, \dots$

The proof of $(\mu^{(n+1)}X. \dots)(z) \vdash \neg B(z)$ is adapted from $(\mu^{(n)}X. \dots)(z) \vdash \neg B(z)$

- Construct the following proof, and try to strengthen the conclusion to … $\vdash \neg B(z)$ by the "**smallest**" change

$$\dfrac{\begin{array}{c}\vdots\\(\mu^{(n)}X.\dots)(x) \vdash \neg B(x)\end{array} \qquad \boxed{\begin{array}{c}\vdots\\(\mu^{(n)}X.\dots)(y) \vdash \neg B(y)\end{array}} \qquad \neg B(x) \wedge \neg B(y) \wedge T(x,y,z) \vdash \top}{(\mu^{(n+1)}X.\dots)(x) \vdash \top}$$

Spacer [Komuravelli+, 2014]

Heuristic 5' In strengthening, **keep as many parts as possible unchanged**

A solver for **non-linear CHCs**

$$\{I(x) \implies P(x), \quad P(x) \wedge P(y) \wedge T(x,y,z) \implies P(z), \quad P(z) \implies \neg B(z)\}$$

It has a solution iff $\big(\mu X. \lambda z. I(z) \vee \big(\exists xy. X(x) \wedge X(y) \wedge T(x,y,z)\big)\big)(z) \vDash \neg B(z)$

It constructs proofs of approximations $(\mu^{(n)} X. \dots)(z) \vdash \neg B(z)$ for $n = 1,2, \dots$

The proof of $(\mu^{(n+1)} X. \dots)(z) \vdash \neg B(z)$ is adapted from $(\mu^{(n)} X. \dots)(z) \vdash \neg B(z)$

- Construct the following proof, and try to strengthen the conclusion to $\dots \vdash \neg B(z)$ by the "**smallest**" change

$$\frac{\boxed{\begin{array}{c}\vdots\\ (\mu^{(n)} X. \dots)(x) \vdash \neg B(x)\end{array}} \quad \boxed{\begin{array}{c}\vdots\\ (\mu^{(n)} X. \dots)(y) \vdash \neg B(y)\end{array}} \quad \neg B(x) \wedge \neg B(y) \wedge T(x,y,z) \vdash \top}{(\mu^{(n+1)} X. \dots)(x) \vdash \top}$$

Note on IC3/PDR and Spacer

Concrete methods for finding maximally conservative modifications are important

There are simple methods using **quantified formulas / quantifier elimination**

but **methods with quantifiers / QE are inefficient**

Both IC3/PDR and Spacer do not treat QE as a black box,
but **interleaving operations inside QE with those of the main procedure**

- During the computation of QE, one may obtain $\boldsymbol{QE(\exists x.\phi) = \psi_0} \lor ???$
- The detail of **the unknown part ??? may be irrelevant** to the main procedure
- So **we return to the main proc., freezing the computation of ???**
  - If it later turns out that ??? is important, we will resume that computation.

Future work

Other ideas in the verification community

- k-induction
- Splitter predicate and its generalizations
- Relational verification

Beyond the standard Hoare triples

- Total correctness, ω-regular, angelic nondeterminism, incorrectness
  - They have natural fixed-point encoding
- Verification of a procedural language (e.g., with angelic nondeterminism)
  - It does not seem to have natural encoding in first-order fixed-point logic

Dealing with ranking functions / disjunctively well-founded relations

Future work

Leveraging the characteristics of cyclic proofs

- Simpler invariants (cf. [Das 2020])
- Natural appearance of disjunction

Integrating proof search with interpolating theorem prover or other subprocedures

- Cf. Spacer (integration of search with QE)

Conclusion

Software model-checking algorithms can be seen as **cyclic proof search strategies**

- The connection is rather straightforward

  once the goal sequence is appropriately set

  "All initial states are safe"

  $$I(x) \vdash \nu S(x)$$

  where $\nu S(x) \overset{\nu}{\Leftrightarrow} \neg B(x) \land \left(\forall y. T(x, y) \Rightarrow \nu S(y)\right)$

- Several algorithms can be **reconstructed from simple proof-search heuristics**
- The usefulness of the connection is demonstrated by
  - revealing an unexpected connection: **PDR $\approx$ an efficient game solving algorithm**