Interpolant Strength in Model Checking
Based on CAV’12 work

Simone Fulvio Rollini, O. Sery, N. Sharygina

Formal Verification Lab, University of Lugano

November 4th, 2012
Outline

1 Background
   - Interpolation for Model Checking
   - Labeled Interpolation Systems
Outline

1 Background
   - Interpolation for Model Checking
   - Labeled Interpolation Systems

2 Contribution
   - Interpolant Strength in Model Checking
   - Simultaneous Abstraction: Requirements and Constraints
   - Path Interpolation: Requirements and Constraints
1 Background
  - Interpolation for Model Checking
  - Labeled Interpolation Systems

2 Contribution
  - Interpolant Strength in Model Checking
  - Simultaneous Abstraction: Requirements and Constraints
  - Path Interpolation: Requirements and Constraints

3 Conclusions
Outline

1 Background
- Interpolation for Model Checking
- Labeled Interpolation Systems

2 Contribution
- Interpolant Strength in Model Checking
- Simultaneous Abstraction: Requirements and Constraints
- Path Interpolation: Requirements and Constraints

3 Conclusions
1. Background
   - Interpolation for Model Checking
   - Labeled Interpolation Systems

2. Contribution
   - Interpolant Strength in Model Checking
   - Simultaneous Abstraction: Requirements and Constraints
   - Path Interpolation: Requirements and Constraints

3. Conclusions
• Model Checking
Symbolic Model Checking
Hardware and Software Verification

- Model Checking
  - System model vs behavioral property specification
Symbolic Model Checking
Hardware and Software Verification

- Model Checking
  - System model vs behavioral property specification
- Symbolic SAT-based approach
Symbolic Model Checking
Hardware and Software Verification

- Model Checking
  - System model vs behavioral property specification

- Symbolic SAT-based approach
  - System and properties as logic formulae
• Model Checking
  • System model vs behavioral property specification

• Symbolic SAT-based approach
  • System and properties as logic formulae
  • Problem encoding into logic (SAT)
Symbolic Model Checking
Hardware and Software Verification

- Model Checking
  - System model vs behavioral property specification

- Symbolic SAT-based approach
  - System and properties as logic formulae
  - Problem encoding into logic (SAT)
  - Problem solving by means of reasoning engines (SAT solvers)
Interpolation
Applications to Symbolic Model Checking

- Bounded model checking: approximate reachability set computation [McM03]
Interpolation
Applications to Symbolic Model Checking

- Bounded model checking: approximate reachability set computation [McM03]
- Predicate abstraction refinement based on spurious behaviors [HJRM04]
Interpolation
Applications to Symbolic Model Checking

- Bounded model checking: approximate reachability set computation [McM03]
- Predicate abstraction refinement based on spurious behaviors [HJRM04]
- Transition relation approximation [JM05]
Interpolation
Applications to Symbolic Model Checking

- Bounded model checking: approximate reachability set computation [McM03]
- Predicate abstraction refinement based on spurious behaviors [HJRM04]
- Transition relation approximation [JM05]
- Lazy abstraction [McM06]
• Bounded model checking: approximate reachability set computation [McM03]

• Predicate abstraction refinement based on spurious behaviors [HJRM04]

• Transition relation approximation [JM05]

• Lazy abstraction [McM06]

• Software upgrade checking [Pincette,SFS12]
Interpolation
Applications to Symbolic Model Checking

- Bounded model checking: approximate reachability set computation [McM03]
- Predicate abstraction refinement based on spurious behaviors [HJRM04]
- Transition relation approximation [JM05]
- Lazy abstraction [McM06]
- Software upgrade checking [Pincette, SFS12]

Property-based overapproximation
• Various applications, different interpolation requirements
Various applications, different interpolation requirements

Various interpolation systems [P97, McM04, DKPW10]
• Various applications, different interpolation requirements

• Various interpolation systems [P97, McM04, DKPW10]

• Various interpolant features
Open Issues and Contribution

- Various applications, different interpolation requirements
- Various interpolation systems [P97,McM04,DKPW10]
- Various interpolant features
  - Strength affects overapproximation coarseness
Interpolation
Open Issues and Contribution

- Various applications, different interpolation requirements
- Various interpolation systems [P97, McM04, DKPW10]
- Various interpolant features
  - Strength affects overapproximation coarseness
  - Strength empirically affects verification performance, convergence
Interpolation
Open Issues and Contribution

• Various applications, different interpolation requirements

• Various interpolation systems [P97, McM04, DKPW10]

• Various interpolant features
  • Strength affects overapproximation coarseness
  • Strength empirically affects verification performance, convergence

⇒ Formalization of requirements for simultaneous abstraction, path interpolation
Open Issues and Contribution

- Various applications, different interpolation requirements
- Various interpolation systems [P97, McM04, DKPW10]
- Various interpolant features
  - Strength affects overapproximation coarseness
  - Strength empirically affects verification performance, convergence

⇒ Formalization of requirements for simultaneous abstraction, path interpolation

⇒ Identification of subset of interpolation systems satisfying requirements
Interpolation [Craig57, McM03]

Background

- Craig’s interpolant $I$ for unsatisfiable $A \land B$

$A \rightarrow I \land B$ is overapproximation $A$ conflicting with $B$.
Interpolation [Craig57, McM03]

Background

- Craig’s interpolant $I$ for unsatisfiable $A \land B$
  - $A \rightarrow I \quad I \land B$ unsatisfiable
Interpolation [Craig57, McM03]

Background

- Craig’s interpolant $I$ for unsatisfiable $A \land B$
  - $A \rightarrow I \quad I \land B$ unsatisfiable
  - $I$ defined over common symbols of $A$ and $B$
Interpolation [Craig57, McM03]

Background

- Craig’s interpolant $I$ for unsatisfiable $A \land B$
  - $A \rightarrow I \land B$ unsatisfiable
  - $I$ defined over common symbols of $A$ and $B$
  - $I$ as overapproximation $A$ conflicting with $B$
Interpolation [Craig57, McM03]

Background

- Craig’s interpolant $I$ for unsatisfiable $A \land B$
  - $A \rightarrow I$ \hspace{1cm} $I \land B$ unsatisfiable
  - $I$ defined over common symbols of $A$ and $B$
  - $I$ as overapproximation $A$ conflicting with $B$
Interpolant Strength
Applications to Symbolic Model Checking

- $l_1$ stronger than $l_2$  \quad $l_1 \rightarrow l_2$
- $I_1$ stronger than $I_2$ \quad $I_1 \rightarrow I_2$

- Interpolation as property-based overapproximation
• $I_1$ stronger than $I_2$ \quad $I_1 \rightarrow I_2$

• Interpolation as property-based overapproximation

• Strength affects approximation coarseness
Interpolant Strength
Applications to Symbolic Model Checking

- $I_1$ stronger than $I_2$  \quad $I_1 \rightarrow I_2$
- Interpolation as property-based overapproximation
- Strength affects approximation coarseness
• Interpolant $I$ for unsatisfiable $A \land B$
• Interpolant $I$ for unsatisfiable $A \land B$

• Different procedures [P97, McM04, DKPW10]
Interpolants Generation

- Interpolant $I$ for unsatisfiable $A \land B$
- Different procedures [P97, McM04, DKPW10]
- Standard generation approach
Interpolants Generation

- Interpolant $I$ for unsatisfiable $A \land B$
- Different procedures [P97, McM04, DPKW10]
- Standard generation approach
  - Derivation of unsatisfiability resolution proof of $A \land B$
Interpolation in SAT

Interpolants Generation

- Interpolant $I$ for unsatisfiable $A \land B$
- Different procedures [P97, McM04, DPKW10]
- Standard generation approach
  - Derivation of unsatisfiability resolution proof of $A \land B$
  - Computation of $I$ from proof structure
Proofs

SAT

- Propositional SAT $p \land (\overline{p} \lor r)$
• Propositional SAT  \[ p \land (\overline{p} \lor r) \]

• Proof of unsatisfiability
Proofs

SAT

- Propositional SAT: \( p \land (\overline{p} \lor r) \)

- Proof of unsatisfiability
  - Certificate of unsatisfiability
Proofs
SAT

- Propositional SAT \( p \land (\overline{p} \lor r) \)

- Proof of unsatisfiability
  - Certificate of unsatisfiability
  - Generated at solving time
Proofs

SAT

- Propositional SAT \( p \land (\overline{p} \lor r) \)

- Proof of unsatisfiability
  - Certificate of unsatisfiability
  - Generated at solving time

- CDCL SAT solver
Proofs

SAT

- Propositional SAT  \( p \land (\overline{p} \lor r) \)

- Proof of unsatisfiability
  - Certificate of unsatisfiability
  - Generated at solving time

- CDCL SAT solver
  - Resolution system
Resolution System
Background

- Literal \( p \), \( \overline{p} \)
Resolution System

Background

- Literal
  \[ p \quad \overline{p} \]

- Clause
  \[ p \lor \overline{q} \lor r \lor \ldots \Rightarrow p\overline{q}r\ldots \quad \text{Empty clause} \quad \bot \]
Resolution System
Background

- Literal \( p \quad \overline{p} \)

- Clause \( p \lor \overline{q} \lor r \lor \ldots \leadsto p\overline{q}r\ldots \)  Empty clause  \( \bot \)

- Input formula \( (p \lor q) \land (r \lor \overline{p}) \land \ldots \leadsto \{pq, r\overline{p}, \ldots\} \)
Resolution System

Background

- **Literal**  \( p \quad \overline{p} \)

- **Clause**  \( p \lor \overline{q} \lor r \lor \ldots \leadsto p\overline{q}r \ldots \)  Empty clause  \( \bot \)

- **Input formula**  \((p \lor q) \land (r \lor \overline{p}) \land \ldots \leadsto \{pq, r\overline{p}, \ldots\}\)

- **Resolution rule**

\[
\begin{array}{c}
pC \\
\overline{p}D
\end{array} \quad \frac{C}{D} \quad \frac{D}{p}
\]

Antecedents:  \( pC \quad \overline{p}D \)  
Resolvent:  \( CD \)  
Pivot:  \( p \)
Resolution Proofs

Resolution System

- Resolution proof of unsatisfiability of a set of clauses $S$
Resolution Proofs

Resolution System

- Resolution proof of unsatisfiability of a set of clauses $S$
  - Tree
  - Leaves as clauses of $S$
  - Inner nodes as resolvents
  - Root as unique $\bot$

Set of clauses $A = \{p \lor q\}$, $B = \{p \land r\}$
Resolution Proofs
Resolution System

- Resolution proof of unsatisfiability of a set of clauses $S$
  - Tree
  - Leaves as clauses of $S$
  - Inner nodes as resolvents
  - Root as unique $\bot$

- Set of clauses $A = \{pq, r\}$
  $B = \{pr, q\}$
Resolution Proofs
Resolution System

• Resolution proof of unsatisfiability of a set of clauses $S$
  • Tree
  • Leaves as clauses of $S$
  • Inner nodes as resolvents
  • Root as unique $\bot$

• Set of clauses $A = \{p\overline{q}, r\}$ $B = \{\overline{pr}, q\}$

• Proof of unsatisfiability

\[
\begin{array}{c}
\overline{p}q \\
\overline{pr} \\
\overline{qr} \\
\overline{qr} \\
\overline{q} \\
\overline{q} \\
\bot
\end{array}
\]
1 Background
   - Interpolation for Model Checking
   - Labeled Interpolation Systems

2 Contribution
   - Interpolant Strength in Model Checking
   - Simultaneous Abstraction: Requirements and Constraints
   - Path Interpolation: Requirements and Constraints

3 Conclusions
Interpolation system parametric in labeling function [DKPW10]
• Interpolation system parametric in labeling function [DKPW10]

• Interpolant determined by proof and labeling
• Interpolation system parametric in labeling function [DKPW10]

• Interpolant determined by proof and labeling

• Generalization of [P97,McM04] \((P, M, M')\)
• Interpolation system parametric in labeling function [DKPW10]

• Interpolant determined by proof and labeling

• Generalization of [P97,McM04] \((P, M, M')\)

• Strength comparison can be reduced to labeling comparison
• Labeling $L$ for $A \land B$
Labeling $L$ for $A \land B$

- Label $\in \{a, b, ab\}$
Labeling

Labeled Interpolation Systems

- Labeling $L$ for $A \land B$
  - Label $\in \{a, b, ab\}$
  - Individual clause literals
Labeling

Labeled Interpolation Systems

- Labeling $L$ for $A \land B$
  - Label $\in \{a, b, ab\}$
  - Individual clause literals

- $A$-local $\mapsto a$, $B$-local $\mapsto b$, $AB$-common $\mapsto \{a, b, ab\}$
Labeling
Labeled Interpolation Systems

- Labeling $L$ for $A \land B$
  - Label $\in \{a, b, ab\}$
  - Individual clause literals

- $A$-local $\mapsto a$, $B$-local $\mapsto b$, $AB$-common $\mapsto \{a, b, ab\}$

- $A = (\overline{p} \lor ?q) \land (p \lor \overline{q})$
  - $B = (\overline{q} \lor r) \land (q \lor \overline{r})$
Labeling Lattice [DKPW10]
Labeled Interpolation Systems

- $b \preceq ab \preceq a \sim\rightarrow (\alpha_1, \ldots, \alpha_n) \preceq (\beta_1, \ldots, \beta_n)$
Labeling Lattice [DKPW10]

Labeled Interpolation Systems

- $b \preceq ab \preceq a \leadsto (\alpha_1, \ldots, \alpha_n) \preceq (\beta_1, \ldots, \beta_n)$

- $(\alpha_1, \ldots, \alpha_n) \preceq (\beta_1, \ldots, \beta_n) \Rightarrow l_1 \rightarrow l_2$
Labeling Lattice [DKPW10]

Labeled Interpolation Systems

- $b \preceq ab \preceq a \Rightarrow (\alpha_1, \ldots, \alpha_n) \preceq (\beta_1, \ldots, \beta_n)$
- $(\alpha_1, \ldots, \alpha_n) \preceq (\beta_1, \ldots, \beta_n) \implies I_1 \rightarrow I_2$
- Labeling lattice
1 Background
- Interpolation for Model Checking
- Labeled Interpolation Systems

2 Contribution
- Interpolant Strength in Model Checking
- Simultaneous Abstraction: Requirements and Constraints
- Path Interpolation: Requirements and Constraints

3 Conclusions
1 Background
   ● Interpolation for Model Checking
   ● Labeled Interpolation Systems

2 Contribution
   ● Interpolant Strength in Model Checking
   ● Simultaneous Abstraction: Requirements and Constraints
   ● Path Interpolation: Requirements and Constraints

3 Conclusions
Interpolant Strength in Model Checking

Contribution

- Systematic use of interpolation in symbolic model checking
Systematic use of interpolation in symbolic model checking

Focus on **interpolant strength**
Interpolant Strength in Model Checking

Contribution

- Systematic use of interpolation in symbolic model checking

- Focus on **interpolant strength**

⇒ **Scenarios**: simultaneous abstraction, path interpolation
• Systematic use of interpolation in symbolic model checking

• Focus on \textbf{interpolant strength}

⇒ \textbf{Scenarios}: simultaneous abstraction, path interpolation

• Generation of multiple interpolants $I_1, \ldots, I_n$
• Systematic use of interpolation in symbolic model checking

• Focus on **interpolant strength**

⇒ **Scenarios**: simultaneous abstraction, path interpolation

  • Generation of multiple interpolants $I_1, \ldots, I_n$
  • Additional requirements on $I_1, \ldots, I_n$
• Systematic use of interpolation in symbolic model checking

• Focus on **interpolant strength**

⇒ **Scenarios**: simultaneous abstraction, path interpolation
  - Generation of multiple interpolants $I_1, \ldots, I_n$
  - Additional requirements on $I_1, \ldots, I_n$

⇒ **Constraints on labeled interpolation systems**
Interpolant Strength in Model Checking

Contribution

- Systematic use of interpolation in symbolic model checking
- Focus on **interpolant strength**

⇒ **Scenarios**: simultaneous abstraction, path interpolation
  - Generation of multiple interpolants $I_1, \ldots, I_n$
  - Additional requirements on $I_1, \ldots, I_n$

⇒ **Constraints on labeled interpolation systems**
  - Generation of each $I_i$ with different $L_i$
• Systematic use of interpolation in symbolic model checking

• Focus on **interpolant strength**

⇒ **Scenarios**: simultaneous abstraction, path interpolation
  
  • Generation of multiple interpolants \( I_1, \ldots, I_n \)
  
  • Additional requirements on \( I_1, \ldots, I_n \)

⇒ **Constraints on labeled interpolation systems**

  • Generation of each \( I_i \) with different \( L_i \)
  
  • Identification of constraints on \( L_1, \ldots, L_n \)
• Simultaneous abstraction
Applications to Model Checking
Labeled Interpolation Systems

• Simultaneous abstraction
  • Software upgrade checking [Pincette,SFS12]
Applications to Model Checking
Labeled Interpolation Systems

- Simultaneous abstraction
  - Software upgrade checking [Pincette, SFS12]
- Path interpolation
Applications to Model Checking
Labeled Interpolation Systems

- Simultaneous abstraction
  - Software upgrade checking [Pincette, SFS12]

- Path interpolation
  - Counterexample-guided abstraction refinement [CGJLV00]
Outline

1 Background
   - Interpolation for Model Checking
   - Labeled Interpolation Systems

2 Contribution
   - Interpolant Strength in Model Checking
   - Simultaneous Abstraction: Requirements and Constraints
   - Path Interpolation: Requirements and Constraints

3 Conclusions
Simultaneous Abstraction

- Program safe

\[
\begin{align*}
\phi_{\text{main}} &\land \phi_{f_1} \land \phi_{f_2} \land \phi_{f_3} \land \phi_{f_4} \\
\text{UNSAT} \quad &
\quad \\
\phi_{f_1} \land \phi_{f_2} \land \phi_{f_3} \land \phi_{f_4} \\
\text{UNSAT} \quad &
\quad \\
\phi_{f_2} \land \phi_{f_4} \\
\text{Check interpolants} &
\end{align*}
\]

Functions update

- \( f_2 \mapsto f_2' \)
- \( f_4 \mapsto f_4' \)

Program safe? Check

\[
\phi_{f_2'} \rightarrow I_{f_2} \\
\phi_{f_4'} \rightarrow I_{f_4}
\]
Software Upgrade Checking
Simultaneous Abstraction

- Program safe

\[ \phi_{main} \land \phi_{f1} \land \phi_{f2} \land \phi_{f3} \land \phi_{f4} \quad \text{UNSAT} \]
Software Upgrade Checking
Simultaneous Abstraction

- **Program safe**
  \[ \phi_{main} \land \phi_{f1} \land \phi_{f2} \land \phi_{f3} \land \phi_{f4} \quad \text{UNSAT} \]

- **Extract interpolants**
Software Upgrade Checking
Simultaneous Abstraction

- Program safe
  \[ \phi_{main} \land \phi_{f1} \land \phi_{f2} \land \phi_{f3} \land \phi_{f4} \quad \text{UNSAT} \]

- Extract interpolants
  \[ l_{main} \land l_{f1} \land l_{f2} \land l_{f3} \land l_{f4} \quad \text{UNSAT} \]
Software Upgrade Checking
Simultaneous Abstraction

• Program safe
  \[ \phi_{main} \land \phi_{f1} \land \phi_{f2} \land \phi_{f3} \land \phi_{f4} \quad \text{UNSAT} \]

• Extract interpolants
  \[ I_{main} \land I_{f1} \land I_{f2} \land I_{f3} \land I_{f4} \quad \text{UNSAT} \]

• Functions update
  \[ f_2 \leadsto f'_2 \quad f_4 \leadsto f'_4 \]
Software Upgrade Checking
Simultaneous Abstraction

- Program safe: \( \phi_{\text{main}} \land \phi_{f1} \land \phi_{f2} \land \phi_{f3} \land \phi_{f4} \) \text{ UNSAT}
- Extract interpolants: \( l_{\text{main}} \land l_{f1} \land l_{f2} \land l_{f3} \land l_{f4} \) \text{ UNSAT}
- Functions update: \( f_2 \leadsto f'_2 \) \( f_4 \leadsto f'_4 \)
- Program safe?

S.F.Rollini (USI)  Interpolant Strength in Model Checking  November 4th, 2012
Software Upgrade Checking
Simultaneous Abstraction

- Program safe
  \[ \phi_{main} \land \phi_{f1} \land \phi_{f2} \land \phi_{f3} \land \phi_{f4} \land \text{UNSAT} \]

- Extract interpolants
  \[ l_{main} \land l_{f1} \land l_{f2} \land l_{f3} \land l_{f4} \land \text{UNSAT} \]

- Functions update
  \[ f_2 \leadsto f'_2 \quad f_4 \leadsto f'_4 \]

- Program safe?
  Check
  \[ \phi_{f'2} \rightarrow l_{f2} \quad \phi_{f'4} \rightarrow l_{f4} \]
Software Upgrade Checking

Simultaneous Abstraction

- Program safe
  \[ \phi_{main} \land \phi_{f1} \land \phi_{f2} \land \phi_{f3} \land \phi_{f4} \quad \text{UNSAT} \]

- Extract interpolants
  \[ I_{main} \land I_{f1} \land I_{f2} \land I_{f3} \land I_{f4} \quad \text{UNSAT} \]

- Functions update
  \[ f_2 \leadsto f'_2 \quad f_4 \leadsto f'_4 \]

- Program safe?
  \[ \text{Check} \quad \phi_{f'2} \rightarrow I_{f2} \quad \phi_{f'4} \rightarrow I_{f4} \]
• Requirement: $I_1 \land \ldots \land I_n \quad \text{UNSAT}$
Results

Simultaneous Abstraction

- Requirement: \( I_1 \land \ldots \land I_n \ \text{UNSAT} \)

- Satisfied for: \( L_1, \ldots, L_n \preceq \text{Pudlák} \)
Results
Simultaneous Abstraction

- Requirement: \( I_1 \land \ldots \land I_n \) UNSAT
- Satisfied for: \( L_1, \ldots, L_n \preceq \text{Pudlák} \)
- Not satisfied in general for: \( L_i \succ \text{Pudlák} \)
Results
Simultaneous Abstraction

- Requirement: \( I_1 \land \ldots \land I_n \) UNSAT

- Satisfied for: \( L_1, \ldots, L_n \preceq \text{Pudlák} \)

- Not satisfied in general for: \( L_i \succ \text{Pudlák} \)
Outline

1 Background
   - Interpolation for Model Checking
   - Labeled Interpolation Systems

2 Contribution
   - Interpolant Strength in Model Checking
   - Simultaneous Abstraction: Requirements and Constraints
   - Path Interpolation: Requirements and Constraints

3 Conclusions
• Counterexample-guided abstraction refinement
  • Abstract $\rightarrow$ Check $\rightarrow$ Refine
Counterexample-guided abstraction refinement
  - Abstract $\rightarrow$ Check $\rightarrow$ Refine

Spurious trace
• Counterexample-guided abstraction refinement
  • Abstract $\rightarrow$ Check $\rightarrow$ Refine

• Spurious trace $\tau_1 \land \ldots \land \tau_n$ UNSAT

\[
\begin{align*}
\text{init} & \quad \tau_1 & \quad \tau_2 & \quad \ldots & \quad \tau_i & \quad \ldots & \quad \tau_n & \quad \text{error} \\
\top & \quad & \quad & \quad & \quad & \quad & \bot
\end{align*}
\]
• Counterexample-guided abstraction refinement
  • Abstract $\rightarrow$ Check $\rightarrow$ Refine

• Spurious trace $\tau_1 \land \ldots \land \tau_n$ UNSAT

S.F. Rollini (USI) Interpolant Strength in Model Checking November 4th, 2012 26 / 30
CEGAR
Path Interpolation

- Counterexample-guided abstraction refinement
  - Abstract $\rightarrow$ Check $\rightarrow$ Refine

- Spurious trace $\tau_1 \land \ldots \land \tau_n$ UNSAT

- Extract interpolants
  $\top \land \tau_1 \rightarrow I_1$
  $I_i \land \tau_{i+1} \rightarrow I_{i+1}$
  $I_{n-1} \land \tau_n \rightarrow \bot$
• Counterexample-guided abstraction refinement
  • Abstract $\rightarrow$ Check $\rightarrow$ Refine

• Spurious trace $\tau_1 \land \ldots \land \tau_n$ UNSAT

$\begin{array}{c}
\text{init} \quad \tau_1 \quad \tau_2 \quad \ldots \quad \tau_i \quad \ldots \quad \tau_n \quad \text{error}
\end{array}$

$\begin{array}{c}
\top \quad \tau_1 \quad \tau_2 \quad \ldots \quad \tau_i \quad \ldots \quad \tau_n \quad \bot
\end{array}$

• Extract interpolants
  $\top \land \tau_1 \rightarrow l_1$  $l_i \land \tau_{i+1} \rightarrow l_{i+1}$  $l_{n-1} \land \tau_n \rightarrow \bot$

$\begin{array}{c}
\text{init} \quad \tau_1 \quad \tau_2 \quad \ldots \quad \tau_i \quad \ldots \quad \tau_n \quad \text{error}
\end{array}$

$\begin{array}{c}
\top \quad l_1 \quad l_2 \quad \ldots \quad l_{i-1} \quad l_i \quad l_{n-1} \quad \bot
\end{array}$
- Counterexample-guided abstraction refinement
  - Abstract $\rightarrow$ Check $\rightarrow$ Refine

- Spurious trace $\tau_1 \land \ldots \land \tau_n$ UNSAT

- Extract interpolants

$$\top \land \tau_1 \rightarrow I_1 \quad I_i \land \tau_{i+1} \rightarrow I_{i+1} \quad I_{n-1} \land \tau_n \rightarrow \bot$$

\[ \begin{array}{cccccccc}
\text{init} & \tau_1 & \tau_2 & \ldots & \tau_i & \ldots & \tau_n & \text{error} \\
\top & l_1 & l_2 & \ldots & l_{i-1} & l_i & l_{n-1} & \bot 
\end{array} \]
Results
Path Interpolation

- Requirement:  \( \tau_1 \rightarrow l_1 \quad l_i \land \tau_{i+1} \rightarrow l_{i+1} \quad l_{n-1} \land \tau_n \rightarrow \bot \)
• Requirement: \( \tau_1 \rightarrow l_1 \quad l_i \land \tau_{i+1} \rightarrow l_{i+1} \quad l_{n-1} \land \tau_n \rightarrow \bot \)

• Satisfied for: \( L_1 \preceq \ldots \preceq L_n \)
Results
Path Interpolation

- Requirement: \( \tau_1 \rightarrow I_1 \quad I_i \land \tau_{i+1} \rightarrow I_{i+1} \quad I_{n-1} \land \tau_n \rightarrow \perp \)

- Satisfied for: \( L_1 \preceq \ldots \preceq L_n \)
1 Background
   - Interpolation for Model Checking
   - Labeled Interpolation Systems

2 Contribution
   - Interpolant Strength in Model Checking
   - Simultaneous Abstraction: Requirements and Constraints
   - Path Interpolation: Requirements and Constraints

3 Conclusions
• Interpolant strength in symbolic model checking
• Interpolant strength in symbolic model checking
• Simultaneous abstraction, path interpolation
• Interpolant strength in symbolic model checking
• Simultaneous abstraction, path interpolation
  • Generation of multiple interpolants, additional requirements
Summary

- Interpolant strength in symbolic model checking
- Simultaneous abstraction, path interpolation
  - Generation of multiple interpolants, additional requirements
- Constraints on labeled interpolation systems
Summary

- Interpolant strength in symbolic model checking
- Simultaneous abstraction, path interpolation
  - Generation of multiple interpolants, additional requirements
- Constraints on labeled interpolation systems
Ongoing Work

- Necessary and sufficient conditions for labeled interpolation systems

Thanks for your attention!

verify.inf.usi.ch

S.F.Rollini (USI)
Ongoing Work

- Necessary and sufficient conditions for labeled interpolation systems
  - Path interpolation, simultaneous abstraction, state-transition interpolation, tree interpolation

Thanks for your attention!
Ongoing Work

- Necessary and sufficient conditions for labeled interpolation systems
  - Path interpolation, simultaneous abstraction, state-transition interpolation, tree interpolation
- Labeled interpolation systems w.r.t. semantical/syntactical features of interpolants
Ongoing Work

- Necessary and sufficient conditions for labeled interpolation systems
  - Path interpolation, simultaneous abstraction, state-transition interpolation, tree interpolation

- Labeled interpolation systems w.r.t. semantical/syntactical features of interpolants
  - Experimentation with FunFrog, eVolCheck, SAFARI model checkers
Ongoing Work

• Necessary and sufficient conditions for labeled interpolation systems
  • Path interpolation, simultaneous abstraction, state-transition interpolation, tree interpolation

• Labeled interpolation systems w.r.t. semantical/syntactical features of interpolants
  • Experimentation with FunFrog, eVolCheck, SAFARI model checkers

Thanks for your attention!
verify.inf.usi.ch
S.F. Rollini, O. Sery and N. Sharygina

*Leveraging Interpolant Strength in Model Checking.*

CAV 2012.
Example
Labeled Interpolation Systems

- \( A = \{pq, r\} \)
- \( B = \{pr, q\} \)
Example

Labeled Interpolation Systems

- \( A = \{pq, r\} \quad B = \{pr, q\}\)

\[
p, q, r \mapsto (b, b, b)
\]

\[
\begin{array}{c}
pq \\
pr \\
r
\end{array}
\]

\[
\begin{array}{c}
q r \\
r
\end{array}
\]

\[
\begin{array}{c}
q \\
q
\end{array}
\]

\[
\bot [(p \lor q) \land r]
\]
Example
Labeled Interpolation Systems

• $A = \{pq, r\}$  \hspace{1cm}  $B = \{pr, q\}$

\[
p, q, r \mapsto (b, b, b) \quad \begin{array}{cc}
pq & pr \\
\hline 
\bar{q}r & r \\
\hline 
\bar{q} & q \\
\hline 
\bot \left[ (p \lor q) \land r \right]
\end{array}
\quad \begin{array}{cc}
pq & pr \\
\hline 
\bar{q}r & r \\
\hline 
\bar{q} & q \\
\hline 
\bot \left[ (p \land r) \lor \bar{q} \right]
\end{array}
\quad \begin{array}{cc}
p, q, r \mapsto (a, a, a)
\end{array}
\]
Example

Labeled Interpolation Systems

- \( A = \{ pq, r \} \quad B = \{ pr, q \} \)

\( p, q, r \mapsto (b, b, b) \)

\[
\begin{array}{c c c}
pq & pr \\
\hline
qr & r \\
\hline
\bar{q} & q \\
\hline
\bot & [(p \lor \bar{q}) \land r] \\
\end{array}
\]

\( p, q, r \mapsto (a, a, a) \)

\[
\begin{array}{c c c}
pq & pr \\
\hline
qr & r \\
\hline
\bar{q} & q \\
\hline
\bot & [(p \land r) \lor \bar{q}] \\
\end{array}
\]

- \((b, b, b) \preceq (a, a, a)\)
Example
Labeled Interpolation Systems

• \( A = \{ pq, r \} \quad B = \{ pr, q \} \)

\[
\begin{align*}
\text{p, q, r} & \mapsto (b, b, b) & \text{p, q, r} & \mapsto (a, a, a) \\
\text{pq} & \quad \text{pr} \\
\hline \\
\text{qr} & \quad r \\
\hline \\
\text{q} & \quad q \\
\hline \\
\perp \quad [(p \lor \overline{q}) \land r] & \\
\hline \\
\end{align*}
\]

• \((b, b, b) \preceq (a, a, a) \quad \implies \quad (p \lor \overline{q}) \land r \rightarrow (p \land r) \lor \overline{q}\)
Transition Relation Approximation
Simultaneous Abstraction

- BMC: iterative analysis \( k \)-length traces
Transition Relation Approximation
Simultaneous Abstraction

- BMC: iterative analysis $k$-length traces
  - Initial states $S$, abstract transition relation $\hat{T}$, error states $E$
• BMC: iterative analysis $k$-length traces
  • Initial states $S$, abstract transition relation $\hat{T}$, error states $E$

• Check $S \land \hat{T}^0 \land \hat{T}^1 \land \ldots \land \hat{T}^{k-1} \land E^k$
Transition Relation Approximation
Simultaneous Abstraction

- BMC: iterative analysis \(k\)-length traces
  - Initial states \(S\), abstract transition relation \(\hat{T}\), error states \(E\)

- Check
  \[ S \land \hat{T}^0 \land \hat{T}^1 \land \ldots \land \hat{T}^{k-1} \land E^k \land \text{SAT?} \]
Transition Relation Approximation

Simultaneous Abstraction

- BMC: iterative analysis $k$-length traces
  - Initial states $S$, abstract transition relation $\hat{T}$, error states $E$

- Check $S \land \hat{T}^0 \land \hat{T}^1 \land \ldots \land \hat{T}^{k-1} \land E^k$ SAT?

- Check $S \land T^0 \land T^1 \land \ldots \land T^{k-1} \land E^k$
Transition Relation Approximation
Simultaneous Abstraction

- BMC: iterative analysis $k$-length traces
  - Initial states $S$, abstract transition relation $\hat{T}$, error states $E$

- Check $S \land \hat{T}^0 \land \hat{T}^1 \land \ldots \land \hat{T}^{k-1} \land E^k$ SAT?
- Check $S \land T^0 \land T^1 \land \ldots \land T^{k-1} \land E^k$ UNSAT?
Transition Relation Approximation
Simultaneous Abstraction

- **BMC**: iterative analysis $k$-length traces
  - Initial states $S$, abstract transition relation $\hat{T}$, error states $E$

- **Check** $S \land \hat{T}^0 \land \hat{T}^1 \land \ldots \land \hat{T}^{k-1} \land E^k$ SAT?

- **Check** $S \land T^0 \land T^1 \land \ldots \land T^{k-1} \land E^k$ UNSAT?

- Extract interpolants
Transition Relation Approximation
Simultaneous Abstraction

- **BMC:** iterative analysis $k$-length traces
  - Initial states $S$, abstract transition relation $\hat{T}$, error states $E$

- Check
  \[ S \land \hat{T}^0 \land \hat{T}^1 \land \ldots \land \hat{T}^{k-1} \land E^k \leq \text{SAT?} \]

- Check
  \[ S \land T^0 \land T^1 \land \ldots \land T^{k-1} \land E^k \leq \text{UNSAT?} \]

- Extract interpolants
  \[ S \land I_0 \land I_1 \land \ldots \land I_{k-1} \land E^k \leq \text{UNSAT} \]
Transition Relation Approximation
Simultaneous Abstraction

- BMC: iterative analysis $k$-length traces
  - Initial states $S$, abstract transition relation $\hat{T}$, error states $E$

- Check
  \[
  S \land \hat{T}^0 \land \hat{T}^1 \land \ldots \land \hat{T}^{k-1} \land E^k \\
  \text{SAT?}
  \]

- Check
  \[
  S \land T^0 \land T^1 \land \ldots \land T^{k-1} \land E^k \\
  \text{UNSAT?}
  \]

- Extract interpolants
  \[
  S \land l_0 \land l_1 \land \ldots \land l_{k-1} \land E^k \\
  \text{UNSAT}
  \]

- Strengthen $\hat{T}$
Transition Relation Approximation
Simultaneous Abstraction

- BMC: iterative analysis $k$-length traces
  - Initial states $S$, abstract transition relation $\hat{T}$, error states $E$

- Check
  
  $S \land \hat{T}^0 \land \hat{T}^1 \land \ldots \land \hat{T}^{k-1} \land E^k$ SAT?

- Check
  
  $S \land T^0 \land T^1 \land \ldots \land T^{k-1} \land E^k$ UNSAT?

- Extract interpolants
  
  $S \land I_0 \land I_1 \land \ldots \land I_{k-1} \land E^k$ UNSAT

- Strengthen $\hat{T}$
  
  $\hat{T} \land I_0 \land I_1 \land \ldots \land I_{k-1} \leadsto \hat{T}$
Transition Relation Approximation
Simultaneous Abstraction

- BMC: iterative analysis $k$-length traces
  - Initial states $S$, abstract transition relation $\hat{T}$, error states $E$

- Check
  $$S \land \hat{T}^0 \land \hat{T}^1 \land \ldots \land \hat{T}^{k-1} \land E^k \text{ SAT?}$$

- Check
  $$S \land T^0 \land T^1 \land \ldots \land T^{k-1} \land E^k \text{ UNSAT?}$$

- Extract interpolants
  $$S \land l_0 \land l_1 \land \ldots \land l_{k-1} \land E^k \text{ UNSAT}$$

- Strengthen $\hat{T}$
  $$\hat{T} \land l_0 \land l_1 \land \ldots \land l_{k-1} \leadsto \hat{T}$$

- $$S \land \hat{T}^0 \land \hat{T}^1 \land \ldots \land \hat{T}^{k-1} \land E^k \text{ UNSAT!}$$
Transition Relation Approximation

Simultaneous Abstraction

- BMC: iterative analysis $k$-length traces
  - Initial states $S$, abstract transition relation $\hat{T}$, error states $E$

- Check
  $$S \land \hat{T}^0 \land \hat{T}^1 \land \ldots \land \hat{T}^{k-1} \land E^k \text{ SAT?}$$

- Check
  $$S \land T^0 \land T^1 \land \ldots \land T^{k-1} \land E^k \text{ UNSAT?}$$

- Extract interpolants
  $$S \land I_0 \land I_1 \land \ldots \land I_{k-1} \land E^k \text{ UNSAT}$$

- Strengthen $\hat{T}$
  $$\hat{T} \land I_0 \land I_1 \land \ldots \land I_{k-1} \leadsto \hat{T}$$

- $$S \land \hat{T}^0 \land \hat{T}^1 \land \ldots \land \hat{T}^{k-1} \land E^k \text{ UNSAT!}$$