Work in Progress Towards Liveness Verification for Infinite Systems by Abstract Interpretation

Patrick Cousot
cims.nyu.edu/~pcousot/

Joint work with Radhia Cousot www.di.ens.fr/~rcousot/

Limitations of “abstract and model-check” for liveness

- For unbounded transition systems, finite abstractions are
  - Incomplete for termination;
  - Unsound for non-termination;

- And so the limitation is similar for liveness, no counter-example to infinite program execution

Origin of the limitations

- Model-checking is impossible because counter-examples are unbounded infinite

- We need automatic verification not checking

- This requires
  - Infinitary abstractions
  - of well-founded relations / well-orders
  - and effectively computable approximations

i.e. Abstract Interpretation

Unless ...

- One is only interested in liveness in the finite abstract (or the concrete is bounded) \(\rightarrow\) decidable

- Or, model-checking is used for checking the termination proof inductive argument (e.g. given variant functions) \(\rightarrow\) decidable


- Of very limited interest:
  - Program executions are unbounded \(\rightarrow\) undecidable
  - The hardest problem for liveness proofs is to infer the inductive argument, then the proof is “easy”
Analysis and verification with well-founded relations and well-orders

Well-founded relations / Well-orders

• Well-founded relation:

A relation \( r \in \varphi(\mathbb{X} \times \mathbb{X}) \) on a set \( \mathbb{X} \) is well-founded if and only if there is no infinite descending chain \( x_0, x_1, \ldots, x_n, \ldots \) of elements \( x_i, \ i \in \mathbb{N} \) of \( \mathbb{X} \) such that \( \forall n \in \mathbb{N} : (x_{n+1}, x_n) \in r \) (or equivalently \( (x_n, x_{n+1}) \in r^{-1} \)).

\[
\begin{array}{ccccccc}
  r^{-1} & r^{-1} & r^{-1} & r^{-1} & r^{-1} & \cdots \\
\end{array}
\]

• Well-order:

A well-order (or well-ordering) is a poset \( \langle \mathbb{X}, \sqsubseteq \rangle \), which is well-founded and total.

\[
\begin{array}{ccccccc}
  \sqsubseteq & \sqsubseteq & \sqsubseteq & \times & \sqsubseteq & \cdots \\
\end{array}
\]

*Assuming the axiom of choice in set theory.

Maximal trace operational semantics

• A transition system: \( \langle \Sigma, \tau \rangle \)

- states
- transition relation

• Maximal trace operational semantics: set of

  - Finite traces:

  \[
  \tau \tau \tau \tau \tau \tau^{-1}
  \]

  - Infinite traces:

  \[
  \tau \tau \tau \tau \tau \tau \tau^{-1} \cdots
  \]

Relevance to Termination Proof

• Program termination is

\( \langle \Sigma, \tau^{-1} \rangle \) is well-founded

i.e. no infinite execution \( ((\tau^{-1})^{-1} = \tau) \)

\[
\begin{array}{ccccccc}
  \tau & \tau & \tau & \times & \tau & \tau & \tau^{-1} \\
\end{array}
\]
**Relevance to LTL verification**

- \( P \cup Q \) for transition system \( \langle \Sigma, \tau \rangle \)

if and only if

\[ \langle \{ x \in \Sigma \mid P(x) \lor Q(x) \}, \{ (y, x) \in \tau^{-1} \mid \neg Q(x) \land \neg Q(y) \} \rangle \]

is well-founded

---

**General idea of the abstraction**

- Combine two abstractions:
  - Abstraction of a relation to its **well-founded part** (to get a necessary condition for wellfoundedness)
  - Abstraction of this well-founded part to a **well-order** (to get a sufficient condition for wellfoundedness)

\[ \langle \varphi(\mathcal{X} \times \mathcal{X}), \subseteq \rangle \overset{\text{wf}}{\longrightarrow} \langle \mathcal{W}(\mathcal{X}), \subseteq \rangle \overset{\alpha \circ}{\longrightarrow} \langle \mathcal{X} \not\ni \emptyset, \preceq \rangle \]

**Relations**

- We encode relations by a domain and a set of connections between elements of the domains (some may be unconnected)

\[ \mathcal{X}(\mathcal{X}) \triangleq \{ (D, r) \mid D \in \varphi(\mathcal{X}) \land r \in \varphi(D \times D) \} \]

\[ \mathcal{W}(\mathcal{X}) \triangleq \{ (D, r) \in \mathcal{X}(\mathcal{X}) \mid r \in \mathcal{W}(D) \} \]

\( \mathcal{W}(\mathcal{X}) \) is the set of well-founded relations on subsets of the set \( \mathcal{X} \).

- Well-founded relations do not form a lattice for \( \preceq \):

\[ \begin{array}{c}
\text{a} \\
\text{b} \\
\text{c}
\end{array} \cup
\begin{array}{c}
\text{a} \\
\text{b} \\
\text{c}
\end{array} =
\begin{array}{c}
\text{a} \\
\text{b} \\
\text{c}
\end{array} \]
Well-founded part of a relation

- Example of well-founded part of a relation:
  \[(a^{\text{wf}}(r))^{-1}\] where \(\mathcal{B}(r) = \{a, b\}\)

- Formally

\[
\begin{align*}
a^{\text{wf}}(r) & \triangleq \langle \mathcal{B}(r), r \cap (\mathcal{X} \times \mathcal{B}(r)) \rangle \\
\mathcal{B}(r) & \triangleq \{x \in \mathcal{X} \mid \exists (x_i \in \mathcal{X}, i \in \mathbb{N}) : x = x_0 \land \forall i < n : x_i < x_{i+1}\} \\
\gamma^{\text{wf}}((D, w)) & \triangleq w \cup (\mathcal{X} \times \mathcal{D})
\end{align*}
\]

Partial order on relations

- Formalize the intuition of over-approximation of well-founded relations in \(w(\mathcal{X})\)

\[
\begin{align*}
\langle x, y \rangle & \triangleq \langle x', y' \rangle \\
\subseteq & \triangleq \gamma^{\text{wf}}(\langle x', y' \rangle) \subseteq \gamma^{\text{wf}}(\langle x', y' \rangle) \\
& = D' \subseteq D \wedge (D' \times D') \subseteq w' \wedge w' \wedge (\neg D' \times D') = \emptyset
\end{align*}
\]

Best abstraction of the well-founded part

- Any relation can be abstracted to its most precise well-founded part

\[
\langle \mathcal{B}(\mathcal{X} \times \mathcal{X}), \subseteq \rangle \prec \prec \gamma^{\text{wf}} \prec \prec \langle \mathcal{W}(\mathcal{X}), \subseteq \rangle
\]

- The best abstraction provides a necessary and sufficient condition for well-foundedness

- An \(\subseteq\)-over-approximation of this best abstraction yields a sufficient condition for well-foundedness

  if \(a^{\text{wf}}(r) \subseteq \langle D, w \rangle\) then \(r\) is well-founded on \(D\)

Fixpoint characterization of the well-founded part of a relation

- \(a^{\text{wf}}(r) = \text{lfp} \bigtriangledown \lambda \langle D, w \rangle \cdot (\text{min}_v(\mathcal{X}) \cup \text{pre}[r]D, w \cup \{x, y \in r \mid x \in \text{pre}[r]D\})\)

  where

  \[
  \text{pre}[r]X = \{x \in \mathcal{X} \mid \forall y \in \mathcal{X} : r(x, y) \Rightarrow y \in \mathcal{X}\}
  \]

  and \(\langle D, w \rangle \subseteq \langle D', w' \rangle\) if and only if \(D \subseteq D' \wedge w \subseteq w'\).

- By abstraction \(a((D, w)) = D\), we get a fixpoint characterization of the well-foundedness domain.

\[
\mathcal{B}(r) = \text{lfp} \bigtriangledown \lambda X \cdot (\text{min}_v(\mathcal{X}) \cup \text{pre}[r]X)
\]

- We have recent results on under-approximating such fixpoint equations by Abstract Interpretation using abstraction and convergence acceleration by widening/narrowing.
Recent results

- We have studied in
  
  
  Patrick Cousot, Radhia Cousot, Francesco Logozzo: Precondition Inference from Intermittent Assertions and Application to Contracts on Collections. VMCAI 2011: 150-168

- The static inference of such under-approximations do work for the inference of sufficient conditions for well-foundedness

The same infinitary under-approximation techniques do work for the inference of sufficient conditions for well-foundedness

Abstraction of a relation’s well-founded part to a well-order

Why well-orders?

- It is always possible to prove that a relation is well-founded by abstraction to a well order (\(\langle \mathbb{N}, <\rangle\), \(\langle \mathbb{O}, <\rangle\), etc).

- Well-orders are easy to represent in a computer (while arbitrary well-founded relations may not be)
CHAPTER 30. TERMINATION

By combining the abstractions of sections...

\[ D \]

\[ g \]

\[ Y \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]

\[ g \]

\[ Y \]

\[ 8 \]

\[ x \]

\[ l'on peut tester à l'exécution. \]

\[ c \]

\[ 2 \]

\[ c \]

\[ 2 \]

\[ 2 \]
**Fixpoint characterization of the ranking function**

- The best/most precise ranking function is

\[ \text{Lfp}^{\leq} \lambda X. \{ \langle x, 0 \rangle \mid x \in \Sigma \land \forall y \in \Sigma: \langle x, y \rangle \not\in \tau \} \cup \{ \langle x, \bigcup \{ \delta + 1 \mid \exists (y, \delta) \in X: \langle x, y \rangle \in \tau \} \mid x \in \Sigma \land \exists (y, \delta) \in X: \langle x, y \rangle \in \tau \land \forall y \in \Sigma: \langle x, y \rangle \in \tau \implies \exists \delta \in : \langle y, \delta \rangle \in X \} \]

- Examples:

```
0 1 2 3 ...
\rho^{0} \rho \rho
```

**Examples**

- **Segmented ranking function abstract domain:**

\[
\text{while } 1(x \geq 0) \text{ do } f \in \mathbb{Z} \mapsto \mathbb{N} \quad \text{(at point 1)}
\]

\[ 2x := -2x + 10 \]

\[ \text{od}^3 f(x) = 0 \]

**No widening:**

<table>
<thead>
<tr>
<th>( x )</th>
<th>( f(x) )</th>
<th>( f(x) )</th>
<th>( f(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x &lt; 0 )</td>
<td>( 1 )</td>
<td>( 1 )</td>
<td>( 1 )</td>
</tr>
<tr>
<td>( 1 )</td>
<td>( 1 )</td>
<td>( 2 )</td>
<td>( 2 )</td>
</tr>
<tr>
<td>( 2 )</td>
<td>( 1 )</td>
<td>( 2 )</td>
<td>( 2 )</td>
</tr>
<tr>
<td>( x \geq 0 )</td>
<td>( 1 )</td>
<td>( 1 )</td>
<td>( 1 )</td>
</tr>
</tbody>
</table>

**Recent results**

- We have recent results on approximating such fixpoint equations by **Abstract Interpretation** using abstraction and convergence acceleration by widening/narrowing.

Patrick Cousot, Radhia Cousot: An abstract interpretation framework for termination. POPL 2012: 245-258

- **Combined with segmentation**


these techniques have been successfully implemented for termination proofs.


- The same techniques do work for the inference of ranking functions in any other contexts.

**Widening**

- Example of widening of abstract piecewise-defined ranking functions. The result of widening \( v_1^\# \) (shown in (a)) with \( v_2^\# \) (shown in (b)) is shown in (c).

- **Widenings enforce convergence** (at the cost of loss of precision on the termination domain and maximal number of steps before termination)

Widening (cont’d)

- Example of loss of precision by widening on the termination domain \( x \in \mathbb{Q} \)

\[
\text{while } 1(x < 10) \text{ do } \\
\text{2 } x := 2x \\
\text{od}
\]

\( f(x) = \begin{cases} 
3 & 5 \leq x < 10 \\
1 & 10 \leq x 
\end{cases} \)

(terminates iff \( x > 0 \), at least a partial result!

- But with \( x \in \mathbb{Z} \),

\[
f(x) = \begin{cases} 
9 & x = 1 \\
7 & x = 2 \\
5 & 3 \leq x \leq 4 \\
3 & 5 \leq x < 9 \\
1 & 10 \leq x 
\end{cases}
\]

What Next?

- Verification of LTL specifications for infinite unbounded transition systems (including software)

- Full automatic verification not debugging/bounded checking/etc (there are no counter-examples for infinite unbounded non-wellfoundedness)

Conclusion

- For well-foundedness/liveness, Abstract interpretation with infinitary abstractions and convergence acceleration 

- The well-foundedness/liveness analysis:
  - requires no given satisfaction precondition [1],
  - requires no special form of loops (e.g. linear, no test in [1])
  - is not restricted to linear ranking functions [1],
  - always terminate thanks to the widening (which is not the case of ad-hoc methods à la Terminator and its numerous derivators based on the search of lasso counter-examples along a single path at a time) [2]
