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
Unless ...

• One is only interested in **liveness in the finite abstract** (or the concrete is bounded) → decidable

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


• Of **very** limited interest:
  
  • Program executions are unbounded → **undecidable**

  • The hardest problem for liveness proofs is to infer the inductive argument, then the proof is “easy”
Origin of the limitations

• Model-checking is impossible because counter-examples are unbounded infinite
  \[ \ldots \] versus \[ \ldots \]

• We need \textit{automatic verification} not \textit{checking}

• This requires
  
  • \textit{Infinitary abstractions}
  
  • \textit{of well-founded relations / well-orders}
  
  • \textit{and effectively computable approximations}

  i.e. \textit{Abstract Interpretation}
Analysis and verification with well-founded relations and well-orders
Maximal trace operational semantics

- A transition system: $\langle \Sigma, \tau \rangle$

- Maximal trace operational semantics: set of
  - Finite traces:
  - Infinite traces:
Well-founded relations / Well-orders

- Well-founded relation:

A relation \( r \in \varnothing(\mathcal{X} \times \mathcal{X}) \) on a set \( \mathcal{X} \) is well-founded if and only if\(^3\) there is no infinite descending chain \( x_0, x_1, \ldots, x_n, \ldots \) of elements \( x_i, \ i \in \mathbb{N} \) of \( \mathcal{X} \) such that \( \forall n \in \mathbb{N} : \langle x_{n+1}, x_n \rangle \in r \) (or equivalently \( \langle x_n, x_{n+1} \rangle \in r^{-1} \)).

- Well-order:

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

---

\(^3\)Assuming the axiom of choice in set theory.
Relevance to Termination Proof

• Program termination is

\[ \langle \Sigma, \tau^{-1} \rangle \text{ is well-founded} \]

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

\[ \tau \cdot \tau \cdot \tau \cdot \tau \cdot \tau \cdot \tau \cdot \tau \cdot \tau \ldots \]
Relevance to LTL verification

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

if and only if

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

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)

\[ \langle \gamma \circ \alpha \rangle_{\text{wf}}: \langle \wp(\mathcal{X} \times \mathcal{X}), \subseteq \rangle \xrightarrow{\alpha_{\text{wf}}} \langle \wp(\mathcal{X}), \sqsubseteq \rangle \xrightarrow{\gamma_{\text{wf}}} \langle \mathcal{X} \not\rightarrow \emptyset, \preceq \rangle \]

- **relation**
- **well-founded part**
- **well-order on founded part**
Abstraction of relations to their well-founded part
Relations

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

\[
\begin{align*}
\mathcal{R}(\mathcal{X}) & \triangleq \{ \langle D, r \rangle \mid D \in \wp(\mathcal{X}) \land r \in \wp(D \times D) \} \\
\mathcal{W}(\mathcal{X}) & \triangleq \{ \langle D, r \rangle \in \mathcal{R}(\mathcal{X}) \mid r \in \mathcal{W}_f(D) \}
\end{align*}
\]

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

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

\[
\begin{align*}
\begin{array}{ccc}
& a & \\
b & & c \\
& & \\
\end{array}
\end{align*}
\cup
\begin{align*}
\begin{array}{ccc}
& a & \\
b & & c \\
& & \\
\end{array}
\end{align*}
=
\begin{align*}
\begin{array}{ccc}
& a & \\
b & & c \\
& & \circlearrowright \\
\end{array}
\end{align*}
\]
Well-founded part of a relation

- Example of well-founded part of a relation:

\[(\alpha^{\text{wf}}(r))^{-1}\quad \text{where}\quad \delta(r) = \{a, b\}\]

- Formally

\[
\begin{align*}
\alpha^{\text{wf}}(r) &\triangleq \langle \delta(r), \ r \cap (\mathcal{X} \times \delta(r)) \rangle \quad \text{where} \\
\delta(r) &\triangleq \{x \in \mathcal{X} \mid \forall \langle x_i \in \mathcal{X}, \ i \in \mathbb{N} \rangle : x = x_0 \land \forall i \in \mathbb{N} : x_i \ r^{-1} \ x_{i+1} \} \\
\gamma^{\text{wf}}(\langle D, w \rangle) &\triangleq w \cup (\mathcal{X} \times \neg D)
\end{align*}
\]
Partial order on relations

- Formalize the intuition of over-approximation of well-founded relations in \( w(x) \)

\[
(a_{\text{wf}}(r))^{-1} \quad \langle D, w \rangle
\]

\[
\mathbb{H} \subseteq \mathbb{G}
\]

\[
\alpha \leftarrow \mathbb{H} \subseteq \mathbb{G}
\]

\[
\gamma \leftarrow \langle D, w \rangle \subseteq \langle D', w' \rangle
\]

- Formal definition:

\[
\langle D, w \rangle \preceq \langle D', w' \rangle \triangleq \gamma_{\text{wf}}(\langle D, w \rangle) \subseteq \gamma_{\text{wf}}(\langle D', w' \rangle)
\]

\[
= D' \subseteq D \land w \cap (D' \times D') \subseteq w' \land w \cap (\neg D' \times D') = \emptyset
\]
Best abstraction of the well-founded part

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

\[
\langle \varnothing(\mathcal{X} \times \mathcal{X}), \subseteq \rangle \xleftrightarrow{\gamma_{\text{wf}}} \langle \mathcal{W}(\mathcal{X}), \preceq \rangle
\]

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

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

if \(\alpha_{\text{wf}}(r) \preceq \langle D, w \rangle\) then \(r\) is well-founded on \(D\)
Fixpoint characterization of the well-founded part of a relation

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

where

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

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

- By abstraction \( \alpha(\langle D, w \rangle) = D \), we get a fixpoint characterization of the wellfoundedness domain.

- \( \delta(r) = \text{lfp} \subseteq \lambda X \cdot \min_r(\mathcal{X}) \cup \text{pre}[r] X \)

- 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

• The same infinitary *under-approximation* techniques do work for the inference of sufficient conditions for well-foundedness
A screenshot of the error reporting with the precondition inference.

• **Implemented in Visual Studio contract checker**

---

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 (\(\langle \mathbb{N}, < \rangle, \langle \mathbb{O}, < \rangle, \text{etc} \)).

• Well-orders are easy to represent in a computer (while arbitrary well-founded relations may not be).
Well-order abstraction of a well-founded relation

- Abstraction to a ranking function:

\[
\langle D, w^{-1} \rangle \in \mathcal{W}(\mathcal{X})
\]

\[
\nu_1 = \alpha^\circ(w)
\]

\[
\gamma^\circ(\nu_1) \supseteq w
\]

- Formally

\[
\alpha^\circ \in \mathcal{WF}(D) \leftrightarrow (D \mapsto \emptyset)
\]

\[
\alpha^\circ(w) \triangleq \lambda y \in D \cdot \bigcup \{ \alpha^\circ(w)x + 1 \mid \langle x, y \rangle \in w \}
\]

\[
\gamma^\circ \in (D \mapsto \emptyset) \mapsto \mathcal{WF}(D)
\]

\[
\gamma^\circ(\nu) \triangleq \{ \langle x, y \rangle \in D \times D \mid \nu(x) < \nu(y) \}
\]
Partial order on well-orders

- The length of maximal decreasing chains is over-approximated

- Formally

\[ f \preceq g \triangleq \gamma^o(f) \subseteq \gamma^o(g) \]
Best abstraction

• Any well-founded relation can be abstracted to a most precise well-order

\[ \langle \mathcal{Wf}(D), \subseteq \rangle \xleftarrow{\gamma \circ} \langle D \mapsto \emptyset, \preceq \rangle \]

• An over-approximation of this best abstraction yields over estimates of the (transfinite) lengths of maximal decreasing chains

• The generalized Turing-Floyd method is sound for any such well-order and complete for the best one.
Generalized Turing/Floyd Proof method

- \( \langle \Sigma, \tau^{-1} \rangle \) is well-founded if and only if there exists a ranking function

\[ \nu \in \Sigma \not\rightarrow \mathcal{O} \]

\( \not\rightarrow \) is for partial functions, the class \( \mathcal{O} \) of ordinals is a canonical representative of all well-orders) such that

\[ \forall x \in \text{dom}(\nu) : \forall y \in \Sigma : \]

\[ \langle x, y \rangle \in \tau \implies \nu(y) < \nu(x) \land y \in \text{dom}(\nu) \]

- \( \text{dom}(\nu) \) determines the domain of well-foundedness of \( \tau^{-1} \) on \( \Sigma \)
Fixpoint characterization of the ranking function

- The best/most precise ranking function is

\[
\text{Lfp} \subseteq \lambda X \cdot \{ \langle x, 0 \rangle \mid x \in \Sigma \land \forall y \in \Sigma: \langle x, y \rangle \notin \tau \} \cup \\
\{ \langle x, \bigcup \{ \delta + 1 \mid \exists \langle y, \delta \rangle \in X: \langle x, y \rangle \in \tau \} \rangle \mid x \in \Sigma \land \\
\exists \langle y, \delta \rangle \in X: \langle x, y \rangle \in \tau \land \forall y \in \Sigma: \langle x, y \rangle \in \tau \implies \exists \delta \in \\
: \langle y, \delta \rangle \in X \}
\]

- Examples:
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*

  Patrick Cousot, Radhia Cousot, Francesco Logozzo: A parametric segmentation functor for fully automatic and scalable array content analysis. POPL 2011: 105-118

  \[
  \begin{array}{c|c|c|c}
  0 & [0,100] & [-100,100] & [-100,-1] \\
  \uparrow & \uparrow a & \uparrow b & \uparrow A.length \\
  \end{array}
  \]

  these techniques have been successfully implemented for termination proofs


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

- Segmented ranking function abstract domain:

\[
\text{while } 1(x \geq 0) \text{ do } \\
2x := -2x + 10 \\
\text{od} \quad f(x) = 0
\]

No widening:

<table>
<thead>
<tr>
<th>1st iteration</th>
<th>2nd iteration</th>
<th>\ldots</th>
<th>5th/6th iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 [x &lt; 0]</td>
<td>\bot \text{ } f(x) = \begin{cases} 1 &amp; x &lt; 0 \ 1 &amp; x \geq 0 \end{cases}</td>
<td>\text{ }\text{ } f(x) = \begin{cases} 1 &amp; x &lt; 0 \ 1 &amp; x \geq 0 \end{cases}</td>
<td>\ldots</td>
</tr>
<tr>
<td>1 \quad \bot f(x) = \begin{cases} 1 &amp; x &lt; 0 \ 1 &amp; x \geq 0 \end{cases}</td>
<td>f(x) = \begin{cases} 1 &amp; x &lt; 0 \ 1 &amp; 0 \leq x \leq 5 \ 3 &amp; x &gt; 5 \end{cases}</td>
<td>\ldots</td>
<td>f(x) = \begin{cases} 1 &amp; x &lt; 0 \ 5 &amp; 0 \leq x \leq 2 \ 9 &amp; x = 3 \ 7 &amp; 4 \leq x \leq 5 \ 3 &amp; x &gt; 5 \end{cases}</td>
</tr>
<tr>
<td>2 \quad \bot f(x) = \begin{cases} \bot &amp; x \leq 5 \ 2 &amp; x &gt; 5 \end{cases}</td>
<td>f(x) = \begin{cases} 4 &amp; x \leq 2 \ 1 &amp; 3 \leq x \leq 5 \ 2 &amp; x &gt; 5 \end{cases}</td>
<td>\ldots</td>
<td>f(x) = \begin{cases} 4 &amp; x \leq 2 \ 8 &amp; x = 3 \ 6 &amp; 4 \leq x \leq 5 \ 2 &amp; x &gt; 5 \end{cases}</td>
</tr>
<tr>
<td>2 [x \geq 0]</td>
<td>\bot \text{ } f(x) = \begin{cases} \bot &amp; x \leq 5 \ 3 &amp; x &gt; 5 \end{cases}</td>
<td>f(x) = \begin{cases} \bot &amp; x &lt; 0 \ 5 &amp; 0 \leq x \leq 2 \ 1 &amp; 3 \leq x \leq 5 \ 3 &amp; x &gt; 5 \end{cases}</td>
<td>\ldots</td>
</tr>
</tbody>
</table>

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 \((x \in \mathbb{Q})\)

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

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

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

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

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

---

Conclusion

- For well-foundedness/liveness, \textit{Abstract interpretation} with infinitary abstractions and convergence acceleration $\gg$ finitary abstractions

- The well-foundedness/liveness analysis:
  - requires no given satisfaction precondition \cite{PodelskiRybalchenko2004},
  - requires no special form of loops (e.g. linear, no test in \cite{PodelskiRybalchenko2004})
  - is not restricted to linear ranking functions \cite{PodelskiRybalchenko2004},
  - 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) \cite{CookPodelskiRybalchenko2011}


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)