Transition Invariants for Program Termination

Andreas Podelski

January 9, 2012
Ramsey’s theorem

every infinite complete graph that is colored with finitely many colors contains a monochrome infinite complete subgraph
termination

a program $P$ is *terminating* if

- its transition relation $R_P$ is well-founded
- the relation $R_P$ does not have an infinite chain
- there exists no infinite sequence

$$s_1, s_2, s_3, \ldots$$

where each pair $(s_i, s_{i+1})$ is contained in the relation $R_P$
proving termination

- classical method for proving program termination: construction of a ranking function (one single ranking function for the entire program)
- construction not supported by predicate abstraction
Predicate abstraction

- proof of safety of program
- construction of a (finite) abstract reachability graph
- edges = transitions between (finitely many) abstract states
- abstract reachability graph (with, say, $n$ abstract states) will contain a loop (namely, to accommodate executions with length greater than $n$)
- example: abstraction of \( \text{while}(x>0)\{x--\} \) with set of predicates \{$(x > 0), (x \leq 0)$\}
- finiteness of executions can not be demonstrated by finiteness of paths in abstract reachability graph
new concepts

- transition invariant: combines several ranking functions into a single termination argument
- transition predicate abstraction: automates the computation of transition invariants using automated theorem proving techniques
backward computation for termination

- $\text{terminatingStates} = \text{set of terminating states} = \text{states } s \text{ that do not have an infinite execution}$
- $\text{exitStates} = \text{set of states without successor}$
- state $s$ terminating if $s$ does not have any successor or every successor of $s$ is a terminating state
- $\text{terminatingStates} = \text{least solution of fixpoint equation:}$
  \[ X = \text{weakestPrecondition}(X) \cup \text{exitStates} \]
- program terminates if $\text{initialStates} \subseteq \text{terminatingStates}$
- check of termination requires abstraction of fixpoint (of function based on weakest precondition) from below
- underapproximation - ???
example program: \texttt{ANY-Y}

\begin{verbatim}
l1: y := read_int();
l2: while (y > 0) {
    y := y-1;
}
\end{verbatim}

\[\rho_1 : pc = \ell_1 \land pc' = \ell_2\]
\[\rho_1 : pc = \ell_2 \land pc' = \ell_2 \land y > 0 \land y' = y - 1\]

\begin{itemize}
  \item unbounded non-determinism at line 11 (for \(pc = \ell_1\))
  \item termination of \texttt{ANY-Y} cannot be proved with ranking functions ranging over the set of natural numbers
  \item initial rank must be at least the ordinal \(\omega\)
\end{itemize}
example program Bubble (nested loop)

l1: while (x => 0) {
    y := 1;
    l2: while (y < x) {
        y := y+1;
    }
    x := x-1;
}

ρ₁ : pc = ℓ₁ ∧ pc' = ℓ₂ ∧ x ≥ 0 ∧ x' = x ∧ y' = 1
ρ₂ : pc = ℓ₂ ∧ pc' = ℓ₂ ∧ y < x ∧ x' = x ∧ y' = y + 1
ρ₃ : pc = ℓ₂ ∧ pc' = ℓ₁ ∧ y ≥ x ∧ x' = x − 1 ∧ y' = y

▷ lexicographic ranking function ⟨x, x − y⟩
▷ ordered pair of two ranking functions, x and x − y
program **Choice**

l: while (x > 0 && y > 0) {
  if (read_int()) {
    (x, y) := (x-1, x);
  } else {
    (x, y) := (y-2, x+1);
  }
}

\[ \rho_1 : pc = pc' = \ell \land x > 0 \land y > 0 \land x' = x - 1 \land y' = x \]
\[ \rho_2 : pc = pc' = \ell \land x > 0 \land y > 0 \land x' = y - 2 \land y' = x + 1 \]

- simultaneous-update statements in loop body
- non-deterministic choice
- ranking function?
example program without simple ranking function

```plaintext
1: while (x > 0 && y > 0) {
   if (read_int()) {
      x := x-1;
      y := read_int();
   } else {
      y := y-1;
   }
}
```

\[ \begin{align*}
\rho_1 &: pc = pc' = \ell \land x > 0 \land y > 0 \land x' = x - 1 \\
\rho_2 &: pc = pc' = \ell \land x > 0 \land y > 0 \land x' = x \land y' = y - 1
\end{align*} \]

- non-deterministic choice
- decrement \( x \), forget value of \( y \) or don't change \( x \), decrement \( y \)
transition invariant

given a program $P$ with transition relation $R_P$,

a binary relation $T$ is a a *transition invariant* if it contains the transitive closure of the transition relation:

$$R_P^+ \subseteq T$$

- compare with *invariant*
- inductiveness
a relation $T$ is \textit{disjunctively well-founded} if it is a finite union of well-founded relations:

$$T = T_1 \cup \cdots \cup T_n$$

- in general, union of well-founded relations is itself not well-founded
proof rule for termination

a program \( P \) is terminating if and only if there exists a disjunctively well-founded transition invariant \( T \) for \( P \)

\( T \) must satisfy two conditions,

transition invariant:

\[ R_P^+ \subseteq T \]

disjunctively well-founded:

\[ T = T_1 \cup \cdots \cup T_n \]

where \( T_1, \ldots, T_n \) well-founded
completeness of proof rule

- “only if” \((\Rightarrow)\)
- program \(P\) is terminating \textit{implies} there exists a disjunctively well-founded transition invariant for \(P\)
- trivial:
- if \(P\) is terminating, then both \(R_P\) and \(R_P^+\) are well-founded
- choose \(n = 1\) and \(T_1 = R_P^+\)
soundness of proof rule

- “If” ($\iff$):
  - a program $P$ is terminating if there exists a disjunctively well-founded transition invariant for $P$
- contraposition:
  - if $R_P^+ \subseteq T$, $T = T_1 \cup \cdots \cup T_n$, and $P$ is not terminating, then at least one of $T_1, \ldots, T_n$ is not well-founded
assume $R_P^+ \subseteq T$, $T = T_1 \cup \cdots \cup T_n$, $P$ non-terminating

- there exists an infinite computation of $P$:

  \[ s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots \]

- each pair $(s_i, s_j)$ lies in one of $T_1, \ldots, T_n$

- one of $T_1, \ldots, T_n$ (say, $T_k$) contains infinitely many pairs $(s_i, s_j)$

- contradiction if we obtain an infinite chain in $T_k$ (since $T_k$ is a well-founded relation)

- but ... in general, those pairs $(s_i, s_j)$ do not form a chain
Ramsey’s theorem

every infinite complete graph that is colored with finitely many colors contains a monochrome infinite \textit{complete subgraph}
assume $R_P^+ \subseteq T$, $T = T_1 \cup \cdots \cup T_n$, $P$ non-terminating

- there exists an infinite computation of $P$:
  \[ s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots \]

- take infinite complete graph formed by $s_i$’s
- edge = pair $(s_i, s_j)$ in $R_P^+$, i.e., in one of $T_1, \ldots, T_n$
- edges can be colored by $n$ different colors
- exists monochrome infinite complete subgraph
- all edges in subgraph are colored by, say, $T_k$
- infinite complete subgraph has an infinite path
- obtain infinite chain in $T_k$
- contradiction since $T_k$ is a well-founded relation
assume $R^+_P \subseteq T$, $T = T_1 \cup \cdots \cup T_n$, $P$ non-terminating

- there exists an infinite computation of $P$:
  $$s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots$$

- let a choice function $f$ satisfy
  $$f(k, \ell) \in \{ T_i \mid (s_k, s_\ell) \in T_i \}$$
  for $k, \ell \in \mathbb{N}$ with $k < \ell$

- condition $R^+_P \subseteq T_1 \cup \cdots \cup T_n$ implies that $f$ exists
  (but does not define it uniquely)
- define equivalence relation $\simeq$ on $f$’s domain by
  $$(k, \ell) \simeq (k', \ell') \text{ if and only if } f(k, \ell) = f(k', \ell')$$

- relation $\simeq$ is of finite index since the set of $T_i$’s is finite
- by Ramsey’s Theorem there exists an infinite sequence of natural numbers $k_1 < k_2 < \ldots$ and fixed $m, n \in \mathbb{N}$ such that
  $$(k_i, k_{i+1}) \simeq (m, n) \text{ for all } i \in \mathbb{N}.$$
example program: \texttt{ANY-Y}

\begin{align*}
l1 &: \ y := \text{read_int}(); \\
l2 &: \ \text{while} \ (y > 0) \{} \\
& \quad \ y := y-1; \\
& \ \{} \\
\end{align*}

\begin{align*}
\rho_1 &: \ pc = \ell_1 \land pc' = \ell_2 \\
\rho_1 &: \ pc = \ell_2 \land pc' = \ell_2 \land y > 0 \land y' = y - 1 \\
\end{align*}

\begin{align*}
T_1 &: \ pc = \ell_1 \land pc' = \ell_2 \\
T_2 &: \ y > 0 \land y' < y \\
\end{align*}
example program \textsc{Bubble} \hspace{0.02em} (nested loop)

\hspace{0.02em} l1: while \ (x \Rightarrow \ 0) \{
\hspace{0.02em} \hspace{0.02em} y := 1;
\hspace{0.02em} \hspace{0.02em} l2: \hspace{0.02em} \text{while} \ (y < x) \{
\hspace{0.02em} \hspace{0.02em} \hspace{0.02em} y := y+1;
\hspace{0.02em} \hspace{0.02em} \}
\hspace{0.02em} \hspace{0.02em} x := x-1;
\hspace{0.02em} \}

\rho_1: pc = \ell_1 \land pc' = \ell_2 \land x \geq 0 \land x' = x \land y' = 1
\rho_2: pc = \ell_2 \land pc' = \ell_2 \land y < x \land x' = x \land y' = y + 1
\rho_3: pc = \ell_2 \land pc' = \ell_1 \land y \geq x \land x' = x - 1 \land y' = y

T_1: pc = \ell_1 \land pc' = \ell_2
T_2: pc = \ell_2 \land pc' = \ell_1
T_3: x \geq 0 \land x' < x
T_4: x - y > 0 \land x' - y' < x - y
program \textsc{Choice}

\begin{verbatim}
l: while (x > 0 && y > 0) {
    if (read_int()) {
        (x, y) := (x-1, x);
    } else {
        (x, y) := (y-2, x+1);
    }
}
\end{verbatim}

\textit{\(\rho_1\)}: \(pc = pc' = \ell \land x > 0 \land y > 0 \land x' = x - 1 \land y' = x \)

\textit{\(\rho_2\)}: \(pc = pc' = \ell \land x > 0 \land y > 0 \land x' = y - 2 \land y' = x + 1 \)

\textit{\(T_1\)}: \(x > 0 \land x' < x \)

\textit{\(T_2\)}: \(y > 0 \land y' < y \)

\textit{\(T_3\)}: \(x + y > 0 \land x' + y' < x + y \)
example program without simple ranking function

1: while (x > 0 && y > 0) {
    if (read_int()) {
        x := x-1;
        y := read_int();
    } else {
        y := y-1;
    }
}

\[ \rho_1 : pc = pc' = \ell \land x > 0 \land y > 0 \land x' = x - 1 \]
\[ \rho_2 : pc = pc' = \ell \land x > 0 \land y > 0 \land x' = x \land y' = y - 1 \]

\[ T_1 : x \geq 0 \land x' < x \]
\[ T_2 : y > 0 \land y' < y \]
prove termination of program $P$

- compute a disjunctively well-founded superset of the transitive closure of the transition relation of the program $P$, i.e.,
- construct a finite number of well-founded relations $T_1, \ldots, T_n$ whose union covers $R^+_P$
- show that the inclusion $R^+_P \subseteq T_1 \cup \cdots \cup T_n$ holds
- show that each of the relations $T_1, \ldots, T_n$ is indeed well-founded
prove termination in 3 steps

1. find a finite number of relations $T_1, \ldots, T_n$
2. show that the inclusion $R_P^+ \subseteq T_1 \cup \cdots \cup T_n$ holds
3. show that each relation $T_1, \ldots, T_n$ is well-founded

it is possible to execute the 3 steps in a different order
disjunctively well-founded transition invariants: basis of a new proof rule for program termination

(next) transition predicate abstraction: basis of automation of proof rule

new class of automatic methods for proving program termination
  ▶ combine multiple ranking functions for reasoning about termination of complex program fragments
  ▶ rely on abstraction techniques to make this reasoning efficient