Circular Coinduction
–A Proof Theoretical Foundation–

Grigore Roșu\textsuperscript{1} Dorel Lucanu\textsuperscript{2}

\textsuperscript{1}Department of Computer Science
University of Illinois at Urbana-Champaign, USA
grosu@illinois.edu

\textsuperscript{2}Faculty of Computer Science
Alexandru Ioan Cuza University, Iași, Romania
dlucanu@info.uaic.ro

08/09/2009, CALCO 2009, Udine
Introduction
- CC History
- Behavioral Equivalence, intuitively
- Behavioral Specifications, intuitively
- Circular Coinduction, intuitively

Circular Coinduction Proof System
- Formal Framework
- Coinductive Circularity Principle
- The Proof System

Conclusion
Plan

1. Introduction
   - CC History
   - Behavioral Equivalence, intuitively
   - Behavioral Specifications, intuitively
   - Circular Coinduction, intuitively

2. Circular Coinduction Proof System
   - Formal Framework
   - Coinductive Circularity Principle
   - The Proof System

3. Conclusion
Circular Coinduction: History

1998  first implementation of CC in BOBJ system [J. Goguen & K. Lin & G. Roşu, ASE 2000]

2000  CC formalized as a inference rule enriching hidden logic [G. Roşu & J. Goguen, written in 1999]

2002  CC described as a more complex algorithm [J. Goguen & K. Lin & G. Roşu, WADT 2002]
       (a first version for special contexts, case analysis)


2006  CC implemented in Maude (first version of CIRC) [D. Lucanu & A. Popescu & G. Roşu]

2007  first major refactoring of CIRC [CALCO Tools, 2007]
       (Maude meta-language application, regular strategies as proof tactics, simplification rules)

2009  CC formalized as a proof system [CALCO 2009, this paper]
       – second major refactoring of CIRC [CALCO Tools, 2009]
Behavioral Equivalence: Intuition 1/2

Behavioral equivalence is the non-distinguishability under experiments

Example of streams:

- A stream (of bits) $S$ is an infinite sequence $b_1 : b_2 : b_3 : \ldots$
  - The head of $S$: $hd(S) = b_1$
  - The tail of $S$: $tl(S) = b_2 : b_3 : \ldots$

- Experiments:
  - $hd(\ast \cdot Stream)$, $hd(tl(\ast \cdot Stream))$, $hd(tl(tl(\ast \cdot Stream)))$, \ldots

- The basic elements upon on the experiments are built (here $hd(\ast)$ and $tl(\ast)$) are called derivatives

- Application of an experiment over a stream: $C[S] = C[S/\ast]$

- Two streams $S$ and $S'$ are behavioral equivalent ($S \equiv S'$) iff $C[S] = C[S']$ for each exp. $C$

- For this particular case, beh. equiv. is the same with the equality of streams

- Showing beh. equiv. is $\Pi^0_2$-hard (S. Buss, G. Roşu, 2000, 2006)
Behavioral Equivalence: Intuition 2/2

(not in this paper)

Example of infinite binary trees (over bits):

- a infinite binary tree over $D$ is a function $T: \{L, R\}^* \rightarrow D$
- the root of $T$: $T(\varepsilon)$
- the left subtree $T_\ell$: $T_\ell(w) = T(Lw)$ for all $w$
- the right subtree $T_r$: $T_r(w) = T(Rw)$ for all $w$
- knowing the root $d$, $T_\ell$ and $T_r$, then $T$ can be written as $d/T_\ell, T_r\backslash$.
- the derivatives: $root(\ast:\text{Tree})$, $left(\ast:\text{Tree})$, and $right(\ast:\text{Tree})$
- the experiments: $root(\ast:\text{Tree})$, $root(left(\ast:\text{Tree}))$, $root(right(\ast:\text{Tree}))$ and so on
- two trees $T$ and $T'$ are beh. equiv. ($T \equiv T'$) iff $C[T] = C[T']$ for each exp. $C$
**Behavioral Specifications: Intuition 1/2**

Streams:

- derivatives: \( \text{hd}(\ast : \text{Stream}) \) and \( \text{tl}(\ast : \text{Stream}) \)
- beh specs are derivative-based specs

**STREAM:**

<table>
<thead>
<tr>
<th>Corecursive spec</th>
<th>Behavioral spec</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{zeroes} = 0 : \text{zeroes} )</td>
<td>( \text{hd}(\text{zeroes}) = 0 )</td>
</tr>
<tr>
<td></td>
<td>( \text{tl}(\text{zeroes}) = \text{zeroes} )</td>
</tr>
<tr>
<td>( \text{ones} = 1 : \text{ones} )</td>
<td>( \text{hd}(\text{ones}) = 1 )</td>
</tr>
<tr>
<td></td>
<td>( \text{tl}(\text{ones}) = \text{ones} )</td>
</tr>
<tr>
<td>( \text{blink} = 0 : 1 : \text{blink} )</td>
<td>( \text{hd}(\text{blink}) = 0 )</td>
</tr>
<tr>
<td></td>
<td>( \text{tl}(\text{blink}) = 1 : \text{blink} )</td>
</tr>
<tr>
<td>( \text{zip}(B : S, S') = B : \text{zip}(S', S) )</td>
<td>( \text{hd}(\text{zip}(S, S')) = \text{hd}(S) )</td>
</tr>
<tr>
<td></td>
<td>( \text{tl}(S, S') = \text{zip}(S', S) )</td>
</tr>
</tbody>
</table>

- for streams, this can be done with STR tool (see H. Zantema’s tool paper)
Behavioral Specifications: Intuition 2/2

Infinite binary trees (TREE):
- derivatives: \( \text{root}(\_:\text{Tree}) \), \( \text{left}(\_:\text{Tree}) \), and \( \text{right}(\_:\text{Tree}) \)
- beh specs are derivative-based specs

<table>
<thead>
<tr>
<th>Corecursive spec</th>
<th>Behavioral spec</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{ones} = 1/\text{ones}, \text{ones} \backslash )</td>
<td>( \text{root}(\text{ones}) = 1 )</td>
</tr>
<tr>
<td>( b/\text{T}<em>\ell, \text{T}<em>r \backslash + b'/\text{T}'</em>\ell, \text{T}'<em>r \backslash = b \lor b'/\text{T}</em>\ell + \text{T}'</em>\ell, \text{T}_r + \text{T}'_r \backslash )</td>
<td>( \text{root}(\text{T} + \text{T}') = \text{root}(\text{T}) \lor \text{root}(\text{T}') )</td>
</tr>
<tr>
<td>( \text{thue} = 0/\text{thue}, \text{thue} + \text{one} \backslash )</td>
<td>( \text{root}(\text{thue}) = 0 )</td>
</tr>
<tr>
<td>( \text{left}(\text{thue}) = \text{thue} )</td>
<td>( \text{right}(\text{thue}) = \text{thue} + \text{one} )</td>
</tr>
</tbody>
</table>
Circular Coinduction, intuitively

- the goal is to prove that $\text{zip}(\text{zeroes}, \text{ones}) \equiv \text{blink}$ holds in STREAM

$\text{zip}(\text{zeroes}, \text{ones}) = \text{blink} \leftarrow$

$0 = 0 \checkmark \quad \text{zip}(\text{ones}, \text{zeroes}) = 1: \text{blink}$

$1 = 1 \checkmark \quad \text{zip}(\text{zeroes}, \text{ones}) = \text{blink}$
Circular Coinduction: Intuition 2/2

– the goal is to prove that $\text{ones} + T \equiv \text{ones}$ holds in TREE

\[
\text{ones} + T = \text{ones}
\]

\[
1 \lor \text{root}(T) = 1 \checkmark
\]

\[
\text{ones} + \text{left}(T) = \text{ones} \quad \text{ones} + \text{right}(T) = \text{ones}
\]

– a more challenging property: $\text{thue} + \text{one} = \text{not}(\text{thue})$
Plan

1. Introduction
   - CC History
   - Behavioral Equivalence, intuitively
   - Behavioral Specifications, intuitively
   - Circular Coinduction, intuitively

2. Circular Coinduction Proof System
   - Formal Framework
   - Coinductive Circularity Principle
   - The Proof System

3. Conclusion
A behavioral specification consists of:

- a many-sorted algebraic spec $\mathcal{B} = (S, \Sigma, E)$
  $(S = \text{set of sorts}, \Sigma = \text{set of opns}, E = \text{set of eqns})$
- a set of derivatives $\Delta = \{\delta[\star:h]\}$
  $\delta[\star:h]$ is a context
  the sort $h$ of the special variable $\star$ occurring in a derivative $\delta$ is called hidden; the other sorts are called visible
- each derivative can be seen as an equation transformer:
  if $e$ is $t = t'$ if cond, then $\delta[e]$ is $\delta[t] = \delta[t']$ if cond
  $\Delta[e] = \{\delta[e] \mid \delta \in \Delta\}$
- an entailment relation $\vdash$, which is reflexive, transitive, monotonic, and $\Delta$-congruent ($E \vdash e$ implies $E \vdash \Delta[e]$)
Experiment:
each visible $\delta[\ast:h] \in \Delta$ is an experiment, and
if $C[\ast:h']$ is an experiment and $\delta[\ast:h] \in \Delta$, then so is $C[\delta[\ast:h]]$

Behavioral satisfaction: $B \models e$ iff:
$B \models e$, if $e$ is visible, and $B \models C[e]$ for each experiment $C$, if $e$ is hidden

Behavioral equivalence of $B$: $\equiv_B \overset{\text{def}}{=} \{ e \mid B \models e \}$

A set of equations $G$ is behaviorally closed iff
$B \models \text{visible}(G)$ and $\Delta(G - B^\circ) \subseteq G$,
where $B^\circ = \{ e \mid B \models e \}$

Theorem
(coinduction) The behavioral equivalence $\equiv$ is the largest behaviorally closed set of equations.
The Freezing Operator

– is the most important ingredient of CC

– it inhibits the use of the coinductive hypothesis underneath proper contexts;

– if e is $t = t'$ if $\text{cond}$, then its frozen form is $\boxed{t} = \boxed{t'}$ if $\text{cond}$

($\boxed{-} : s \to \text{Frozen}$)

– $\vdash$ is extended for frozen equations s.t.

(A1) $E \cup F \vdash \boxed{e}$ iff $E \vdash e$, for each visible eqn e;

(A2) $E \cup F \vdash G$ implies $E \cup \delta[F] \vdash \delta[G]$ for each $\delta \in \Delta$, equivalent to saying that for any $\Delta$-context $C$, $E \cup F \vdash G$ implies $E \cup C[F] \vdash C[G]$

Theorem

(coinductive circularity principle) If $\mathcal{B}$ is a behavioral specification and $F$ is a set of hidden equations with $\mathcal{B} \cup \boxed{F} \vdash \boxed{\Delta[F]}$ then $\mathcal{B} \not\vdash F$. 
**Circular Coinduction Proof System**

\[ \begin{array}{c}
\text{B} \cup \text{F} \vdash \emptyset \\
\text{B} \cup \text{F} \vdash \text{G}, \quad \text{B} \cup \text{F} \vdash \{e\} \\
\text{B} \cup \text{F} \cup \{e\} \vdash \text{G} \cup \Delta[e] \\
\end{array} \]

[Done]

[Reduce]

[Derive]

if \( e \) hidden
Soundness

Theorem

(soundness of circular coinduction) If $\mathcal{B}$ is a behavioral specification and $G$ is a set of equations such that $\mathcal{B} \vdash \mathcal{C} G$ is derivable using the Circular Coinduction Proof System, then $\mathcal{B} \vdash G$.

The proof is monolithic and, intuitively, the correctness can be explained in different ways:

1. since each derived path ends up in a cycle, it means that there is no way to show the two original terms behaviorally different by applications of derivatives;

2. the obtained circular graph structure can be used as a backbone to “consume” any possible experiment applied on the two original terms;

3. the equalities that appear as nodes in the obtained graph can be regarded as lemmas inferred in order to prove the original task;

4. when it stabilizes, it “discovers” a relation which is compatible with the derivatives and is the identity on data, so the stabilized set of equations is included in the behavioral equivalence;

5. it incrementally completes a given equality into a bisimulation relation on terms.
Soundness

Theorem

(soundness of circular coinduction) If $B$ is a behavioral specification and $G$ is a set of equations such that $B \vdash \circ \ G$ is derivable using the Circular Coinduction Proof System, then $B \vdash G$.

The proof is monolithic and, intuitively, the correctness can be explained in different ways:

(1) since each derived path ends up in a cycle, it means that there is no way to show the two original terms behaviorally different by applications of derivatives;

(2) the obtained circular graph structure can be used as a backbone to “consume” any possible experiment applied on the two original terms;

(3) the equalities that appear as nodes in the obtained graph can be regarded as lemmas inferred in order to prove the original task;

(4) when it stabilizes, it “discovers” a relation which is compatible with the derivatives and is the identity on data, so the stabilized set of equations is included in the behavioral equivalence;

(5) it incrementally completes a given equality into a bisimulation relation on terms.
Soundness

Theorem

(soundness of circular coinduction) \textit{If } \mathcal{B} \textit{ is a behavioral specification and } \mathcal{G} \textit{ is a set of equations such that } \mathcal{B} \vdash \sqsubseteq \mathcal{G} \textit{ is derivable using the Circular Coinduction Proof System, then } \mathcal{B} \vdash \mathcal{G} \textit{.}

The proof is monolithic and, intuitively, the correctness can be explained in different ways:

(1) since each derived path ends up in a cycle, it means that there is no way to show the two original terms behaviorally different by applications of derivatives;

(2) the obtained circular graph structure can be used as a backbone to “consume” any possible experiment applied on the two original terms;

(3) the equalities that appear as nodes in the obtained graph can be regarded as lemmas inferred in order to prove the original task;

(4) when it stabilizes, it “discovers” a relation which is compatible with the derivatives and is the identity on data, so the stabilized set of equations is included in the behavioral equivalence;

(5) it incrementally completes a given equality into a bisimulation relation on terms.
Soundness

Theorem

(soundness of circular coinduction) If $B$ is a behavioral specification and $G$ is a set of equations such that $B \vdash_C G$ is derivable using the Circular Coinduction Proof System, then $B \vdash G$.

The proof is monolithic and, intuitively, the correctness can be explained in different ways:

(1) since each derived path ends up in a cycle, it means that there is no way to show the two original terms behaviorally different by applications of derivatives;

(2) the obtained circular graph structure can be used as a backbone to “consume” any possible experiment applied on the two original terms;

(3) the equalities that appear as nodes in the obtained graph can be regarded as lemmas inferred in order to prove the original task;

(4) when it stabilizes, it “discovers” a relation which is compatible with the derivatives and is the identity on data, so the stabilized set of equations is included in the behavioral equivalence;

(5) it incrementally completes a given equality into a bisimulation relation on terms.
Soundness

Theorem

(soundness of circular coinduction) If $B$ is a behavioral specification and $G$ is a set of equations such that $B \vdash^{\circlearrowright} G$ is derivable using the Circular Coinduction Proof System, then $B \vdash G$.

The proof is monolithic and, intuitively, the correctness can be explained in different ways:

1. since each derived path ends up in a cycle, it means that there is no way to show the two original terms behaviorally different by applications of derivatives;

2. the obtained circular graph structure can be used as a backbone to “consume” any possible experiment applied on the two original terms;

3. the equalities that appear as nodes in the obtained graph can be regarded as lemmas inferred in order to prove the original task;

4. when it stabilizes, it “disCOVERS” a relation which is compatible with the derivatives and is the identity on data, so the stabilized set of equations is included in the behavioral equivalence;

5. it incrementally completes a given equality into a bisimulation relation on terms.
Soundness

Theorem

(soundness of circular coinduction) If \( B \) is a behavioral specification and \( G \) is a set of equations such that \( B \vdash \oslash G \) is derivable using the Circular Coinduction Proof System, then \( B \vdash G \).

The proof is monolithic and, intuitively, the correctness can be explained in different ways:

1. since each derived path ends up in a cycle, it means that there is no way to show the two original terms behaviorally different by applications of derivatives;

2. the obtained circular graph structure can be used as a backbone to “consume” any possible experiment applied on the two original terms;

3. the equalities that appear as nodes in the obtained graph can be regarded as lemmas inferred in order to prove the original task;

4. when it stabilizes, it “discovers” a relation which is compatible with the derivatives and is the identity on data, so the stabilized set of equations is included in the behavioral equivalence;

5. it incrementally completes a given equality into a bisimulation relation on terms.
Example

\[
\begin{align*}
\text{STREAM} & \cup \left\{ \text{zip(odd(S), even(S))} = S \right\} \implies \emptyset & \text{[Done]} \\
\text{STREAM} & \cup \left\{ \text{zip(odd(S), even(S))} = S \right\} \vdash \text{hd(zip(odd(S), even(S)))} = \text{hd(S)} & \text{[Reduce]} \\
\text{STREAM} & \cup \left\{ \text{zip(odd(S), even(S))} = S \right\} \vdash \text{tl(zip(odd(S), even(S)))} = \text{tl(S)} & \text{[Reduce]} \\
\text{STREAM} & \cup \left\{ \text{zip(odd(S), even(S))} = S \right\} \implies \left\{ \begin{array}{l}
\text{hd(zip(odd(S), even(S)))} = \text{hd(S)}, \\
\text{tl(zip(odd(S), even(S)))} = \text{tl(S)}
\end{array} \right\} & \text{[Derive]}
\end{align*}
\]
Plan

1. Introduction
   - CC History
   - Behavioral Equivalence, intuitively
   - Behavioral Specifications, intuitively
   - Circular Coinduction, intuitively

2. Circular Coinduction Proof System
   - Formal Framework
   - Coinductive Circularity Principle
   - The Proof System

3. Conclusion
Related Approaches

**Context induction** [R. Hennicker, 1990]
- exploits the inductive definition of the experiments [used also here in CCP]
- requires human guidance, generalization of the induction assertions

**Observational Logic** [M. Bidoit, R. Hennicker, and Al. Kurz, 2002]
- model based (organized as an institution)
- there is a strong similarity between our beh equiv $\equiv$ and their infinitary proof system

**Coalgebra**[e.g., J. Adamek 2005, B. Jacobs and J. Rutten 1997] – used to study the states and their operations and their properties
- final coalgebras use to give (behavioral) semantics for processes
- when coalgebra specs are expressed as beh. specs, CC Proof System builds a bisimulation

**Observational proofs by rewriting** [A. Bouhoula and M. Rusinowitch, 2002]
- based on *critical contexts*, which allow to prove or disprove conjectures

**A coinductive calculus of streams** [Jan Rutten, 2005]
- almost all properties proved with CIRC
- extended to infinite binary trees [joint work with Al. Silva]
Future Work

Theoretical aspects:
– in some cases the freezing operator is too restrictive ⇒ extend the proof system with new capabilities (special contexts, generalizations, simplifications etc)
– productivity of the behavioral specs vs. well-definedness
– (full) behavioral specification of the non-deterministic processes (behavioral TRS?)
– complexity of the related problems

CIRC Tool:
– automated case analysis
– more case studies (e.g., behavioral semantics of the functors)
– the use of CC as a framework (its use in other applications)
– its use in program verification and analysis
Thanks!