Circular Coinduction in Coq using Bisimulation-Up-To Techniques

Jörg Endrullis   Dimitri Hendriks   Martin Bodin

VU University Amsterdam, Inria Rennes and ENS Lyon

26th of July, 2013

ITP 2013

Motivation

- Tools proving equalities on streams: Circ, Streambox, ...
- They use *rewriting* circular proofs.

```
Variables (A : stream bit) (f : stream bit → stream bit).

Hypothesis hyp_A : A == zero :: one :: A.

Hypothesis hyp_f_0 : ∀s, f (zero :: s) == zero :: one :: f s.
Hypothesis hyp_f_1 : ∀s, f (one :: s) == f s.

Lemma A_bis_f_A : A == f A.
Proof.
cofix CIH. constructor.
rewrite hyp_A; rewrite hyp_f_0; simpl.
(* ... *)
reflexivity.
```

Motivation

- Tools proving equalities on streams: CIRC, STREAMBOX, ...
- They use *rewriting* circular proofs.

```
Variables (A : stream bit) (f : stream bit → stream bit).

Hypothesis hyp_A : A == zero :: one :: A.

Hypothesis hyp_f_0 : ∀s, f (zero :: s) == zero :: one :: f s.
Hypothesis hyp_f_1 : ∀s, f (one :: s) == f s.

Lemma A_bis_f_A : A == f A.
Proof.
cofix CIH. constructor.
rewrite hyp_A; rewrite hyp_f_0; simpl.
(* ... *)
reflexivity.
```

```
No more subgoals.
```

Motivation

- Tools proving equalities on streams: CIRC, STREAMBOX, ...
- They use *rewriting* circular proofs.

But those proofs aren't directly accepted by COQ!

```
Error:
Recursive definition of CIH is ill-formed.
In environment
(* ... *)
CIH : A == f A
Sub-expression "(fun H : tail (tail (zero :: one :: A)) == tail (tail (f A)) =>
                  Morphisms.trans_co_eq_inv_impl_morphism
                  (* ... *)
                  (hyp_f_1 A) H) (reflexivity (f A))))))" not in
guarded form (should be a constructor, an abstraction, a match, a cofix or a
recursive call).
```

```
reflexivity.
Qed.
```

Coinduction and Guardedness

Induction

Enforce termination.

Coinduction

Enforce productivity.

```
Inductive list :=
  | nil : list
  | cons : bool → list → list.
```

```
CoInductive stream :=
  | cons : bool → stream → stream.
```

```
Fixpoint f (l : list) : list :=
  match l with
  | nil ⇒ nil
  | cons b l' ⇒ rev (f l')
  end.
```

```
CoFixpoint f (s : stream) : stream :=
  match s with
  | cons b s' ⇒ cons (neg b) (f (zip s' s))
  end.
```

Coinduction and Guardedness

Induction

Enforce termination.

Coinduction

Enforce productivity.

```
Inductive list :=
  | nil : list
  | cons : bool → list → list.
```

```
CoInductive stream :=
  | cons : bool → stream → stream.
```

```
Fixpoint f (l : list) : list :=
  match l with
  | nil ⇒ nil
  | cons b l' ⇒ rev (f l')
  end.
```

```
CoFixpoint f (s : stream) : stream :=
  match s with
  | cons b s' ⇒ cons (neg b) (f (zip s' s))
  end.
```

Guardedness and Productivity

		specification	guarded	productive
ones	$=$	$1 :: \mathit{ones}$	✓	✓
$\mathit{read}(s)$	$=$	$\mathrm{hd}\, s :: \mathit{read}(\mathrm{tl}\, s)$	✓	✓
$\mathit{plus}(s, t)$	$=$	$(\mathrm{hd}\, s + \mathrm{hd}\, t) :: \mathit{plus}(\mathrm{tl}\, s, \mathrm{tl}\, t)$	✓	✓

Guardedness and Productivity

		specification	guarded	productive
ones	$=$	$1 :: \mathit{ones}$	✓	✓
$\mathit{read}(s)$	$=$	$\mathrm{hd}\, s :: \mathit{read}(\mathrm{tl}\, s)$	✓	✓
$\mathit{plus}(s, t)$	$=$	$(\mathrm{hd}\, s + \mathrm{hd}\, t) :: \mathit{plus}(\mathrm{tl}\, s, \mathrm{tl}\, t)$	✓	✓
J	$=$	$0 :: \mathrm{tl}\, J$	$\lightning$	$\lightning$

		specification	guarded	productive
ones	$=$	$1 :: \mathit{ones}$	✓	✓
$\mathit{read}(s)$	$=$	$\mathrm{hd}\, s :: \mathit{read}(\mathrm{tl}\, s)$	✓	✓
$\mathit{plus}(s, t)$	$=$	$(\mathrm{hd}\, s + \mathrm{hd}\, t) :: \mathit{plus}(\mathrm{tl}\, s, \mathrm{tl}\, t)$	✓	✓
J	$=$	$0 :: \mathrm{tl}\, J$	$\lightning$	$\lightning$

$$J_1 = 0 :: 0 :: 0 :: 0 :: 0 :: 0 :: 0 :: \dots$$
$$J_2 = 0 :: 1 :: 1 :: 1 :: 1 :: 1 :: 1 :: \dots$$
$$J_3 = 0 :: 1 :: 1 :: 0 :: 1 :: 1 :: 0 :: \dots$$

Guardedness and Productivity

		specification	guarded	productive
ones	$=$	$1 :: \mathit{ones}$	✓	✓
$\mathit{read}(s)$	$=$	$\mathrm{hd}\, s :: \mathit{read}(\mathrm{tl}\, s)$	✓	✓
$\mathit{plus}(s,t)$	$=$	$(\mathrm{hd}\, s + \mathrm{hd}\, t) :: \mathit{plus}(\mathrm{tl}\, s, \mathrm{tl}\, t)$	✓	✓
J	$=$	$0 :: \mathrm{tl}\, J$	$\lightning$	$\lightning$
twos	$=$	$2 :: \mathit{read}(\mathit{twos})$	$\lightning$	✓
nats	$=$	$0 :: \mathit{plus}(\mathit{nats}, \mathit{ones})$	$\lightning$	✓

$$J_1 = 0 :: 0 :: 0 :: 0 :: 0 :: 0 :: 0 :: \dots$$
$$J_2 = 0 :: 1 :: 1 :: 1 :: 1 :: 1 :: 1 :: \dots$$
$$J_3 = 0 :: 1 :: 1 :: 0 :: 1 :: 1 :: 0 :: \dots$$

Guardedness and Productivity

		specification	guarded	productive
ones	$=$	$1 :: \mathit{ones}$	✓	✓
$\mathit{read}(s)$	$=$	$\mathrm{hd}\, s :: \mathit{read}(\mathrm{tl}\, s)$	✓	✓
$\mathit{plus}(s, t)$	$=$	$(\mathrm{hd}\, s + \mathrm{hd}\, t) :: \mathit{plus}(\mathrm{tl}\, s, \mathrm{tl}\, t)$	✓	✓
J	$=$	$0 :: \mathrm{tl}\, J$	$\lightning$	$\lightning$
twos	$=$	$2 :: \mathit{read}(\mathit{twos})$	$\lightning$	✓
nats	$=$	$0 :: \mathit{plus}(\mathit{nats}, \mathit{ones})$	$\lightning$	✓
$\mathit{from}(n)$	$=$	$n :: \mathit{from}(n + 1)$	✓	✓
nats	$=$	$\mathit{from}(0)$		✓

Idea

Hide the computation of the next stream element in an argument.

Bisimilarity

- In Coq, $\sim$ is defined coinductively by the rule

$$\frac{\pi_0 : \mathrm{hd}\, s = \mathrm{hd}\, t \quad \pi' : \mathrm{tl}\, s \sim \mathrm{tl}\, t}{\sim^+ \pi_0\ \pi' : s \sim t} \sim^+$$

- $\sim$ is the largest relation reversely closed under $\sim^+$, $\sim$ is the largest bisimulation
- A proof of $s \sim t$ can be viewed as an infinite sequence

$$\sim^+ \pi_0\ (\sim^+ \pi_1\ (\sim^+ \pi_2\ (\sim^+ \ \ldots)))$$

Bisimilarity

- In Coq, $\sim$ is defined coinductively by the rule

$$\frac{\pi_0 \;:\; \mathrm{hd}\, s = \mathrm{hd}\, t \qquad \pi' \;:\; \mathrm{tl}\, s \sim \mathrm{tl}\, t}{\sim^+ \;\pi_0\;\pi' \;:\; s \sim t}\;\sim^+$$

- $\sim$ is the largest relation reversely closed under $\sim^+$,
  $\sim$ is the largest bisimulation
- A proof of $s \sim t$ can be viewed as an infinite sequence

$$\sim^+ \;\pi_0\; (\sim^+ \;\pi_1\; (\sim^+ \;\pi_2\; (\sim^+ \;\ldots)))$$

Bisimilarity

- In Coq, $\sim$ is defined coinductively by the rule

$$\frac{\pi_0 \,:\, \mathrm{hd}\, s = \mathrm{hd}\, t \quad \pi' \,:\, \mathrm{tl}\, s \sim \mathrm{tl}\, t}{\sim^+\ \pi_0\ \pi' \,:\, s \sim t}\ \sim^+$$

- $\sim$ is the largest relation reversely closed under $\sim^+$,
  $\sim$ is the largest bisimulation
- A proof of $s \sim t$ can be viewed as an infinite sequence

$$\sim^+\ \pi_0\ (\sim^+\ \pi_1\ (\sim^+\ \pi_2\ (\sim^+\ \ldots)))$$

Some Productive Proofs are not Guarded

Define $Z_1 = 0 :: Z_2$ and $Z_2 = 0 :: Z_1$. Show $Z_1 \sim Z_2$.

$$
\dfrac{
  \dfrac{}{0 = 0}\,\textsc{refl}_=
  \qquad
  \dfrac{
    \dfrac{}{0 = 0}\,\textsc{refl}_=
    \qquad
    \dfrac{}{Z_1 \sim Z_2}\,\pi
  }{0 :: Z_1 \sim 0 :: Z_2}\,{\sim^+}
}{
  \dfrac{0 :: 0 :: Z_1 \sim 0 :: 0 :: Z_2}{Z_1 \sim Z_2}\,\textsc{cofix}\ \pi
}\,{\sim^+}
\qquad\qquad
\dfrac{
  \dfrac{}{0 = 0}\,\textsc{refl}_=
  \qquad
  \dfrac{
    \dfrac{}{Z_1 \sim Z_2}\,\pi
  }{Z_2 \sim Z_1}\,\textsc{sym}_\sim
}{
  \dfrac{0 :: Z_2 \sim 0 :: Z_1}{Z_1 \sim Z_2}\,\textsc{cofix}\ \pi
}\,{\sim^+}
$$

proof term	guarded	productive (correct)
$\pi = \sim^+ \ (\mathit{refl}_= \ 0) \ (\sim^+ \ (\mathit{refl}_= \ 0) \ \pi)$	✓	✓
$\pi = \sim^+ \ (\mathit{refl}_= \ 0) \ (\mathit{sym}_\sim \ \pi)$	↯	✓

Some Productive Proofs are not Guarded

Define $Z_1 = 0 :: Z_2$ and $Z_2 = 0 :: Z_1$. Show $Z_1 \sim Z_2$.

$$\dfrac{\dfrac{\textsc{refl}_= \quad \dfrac{}{0 = 0} \qquad \dfrac{\textsc{refl}_= \quad \dfrac{}{0 = 0} \qquad \dfrac{}{Z_1 \sim Z_2}\ \pi}{0 :: Z_1 \sim 0 :: Z_2}\ \sim^+}{0 :: 0 :: Z_1 \sim 0 :: 0 :: Z_2}\ \sim^+}{Z_1 \sim Z_2}\ \textsc{cofix}\ \pi \qquad \dfrac{\dfrac{\textsc{refl}_= \quad \dfrac{}{0 = 0} \qquad \dfrac{\dfrac{}{Z_1 \sim Z_2}\ \pi}{Z_2 \sim Z_1}\ \textsc{sym}_\sim}{0 :: Z_2 \sim 0 :: Z_1}\ \sim^+}{Z_1 \sim Z_2}\ \textsc{cofix}\ \pi$$

proof term	guarded	productive (correct)
$\pi = \sim^+ (\mathit{refl}_= \ 0) \ (\sim^+ (\mathit{refl}_= \ 0) \ \pi)$	✓	✓
$\pi = \sim^+ (\mathit{refl}_= \ 0) \ (\mathit{sym}_\sim \ \pi)$	↯	✓

Some Productive Proofs are not Guarded

Define $Z_1 = 0 :: Z_2$ and $Z_2 = 0 :: Z_1$. Show $Z_1 \sim Z_2$.

$$\dfrac{\dfrac{}{0 = 0}\,\textsc{refl}_= \qquad \dfrac{\dfrac{}{0 = 0}\,\textsc{refl}_= \qquad \dfrac{}{Z_1 \sim Z_2}\,\pi}{0 :: Z_1 \sim 0 :: Z_2}\,\sim^+}{\dfrac{0 :: 0 :: Z_1 \sim 0 :: 0 :: Z_2}{Z_1 \sim Z_2}\,\textsc{cofix}\ \pi}\,\sim^+$$

$$\dfrac{\dfrac{}{0 = 0}\,\textsc{refl}_= \qquad \dfrac{\dfrac{}{Z_1 \sim Z_2}\,\pi}{Z_2 \sim Z_1}\,\textsc{sym}_\sim}{\dfrac{0 :: Z_2 \sim 0 :: Z_1}{Z_1 \sim Z_2}\,\textsc{cofix}\ \pi}\,\sim^+$$

proof term	guarded	productive (correct)
$\pi = \sim^+ \; (\mathit{refl}_= \; 0) \; (\sim^+ \; (\mathit{refl}_= \; 0) \; \pi)$	✓	✓
$\pi = \sim^+ \; (\mathit{refl}_= \; 0) \; (\mathit{sym}_\sim \; \pi)$	↯	✓

Guardedness and Rewritings

Prove $A \sim fA$ where Γ consists of

$$A \sim 0 :: 1 :: A \qquad\qquad \begin{aligned} f(0 :: \sigma) &\sim 0 :: 1 :: f\sigma \\ f(1 :: \sigma) &\sim f\sigma \end{aligned}$$

$$\frac{}{A \sim fA}$$

as a proof term:

Guardedness and Rewritings

Prove $A \sim fA$ where Γ consists of

$$A \sim 0 :: 1 :: A \qquad\qquad \begin{aligned} f(0 :: \sigma) &\sim 0 :: 1 :: f\sigma \\ f(1 :: \sigma) &\sim f\sigma \end{aligned}$$

$$\dfrac{\dfrac{}{A \sim fA}}{A \sim fA}\ \textsc{cofix}\ \pi$$

as a proof term:

cofix π. $(\ldots)$

Guardedness and Rewritings

Prove $A \sim fA$ where Γ consists of

$$A \sim 0 :: 1 :: A \qquad\qquad f(0 :: \sigma) \sim 0 :: 1 :: f\sigma$$

$$f(1 :: \sigma) \sim f\sigma$$

$$\dfrac{\dfrac{\mathrm{hd}\,A = \mathrm{hd}\,(fA) \mid \pi_1 \qquad \dfrac{}{\mathrm{tl}\,A \sim \mathrm{tl}\,(fA)}}{A \sim fA}\,\sim^{+}}{A \sim fA}\ \mathrm{COFIX}\ \pi$$

as a proof term:

cofix $\pi.\,(\sim^{+}\ \pi_1\ \ldots)$

Prove $A \sim fA$ where Γ consists of

$$A \sim 0 :: 1 :: A \qquad\qquad f(0 :: \sigma) \sim 0 :: 1 :: f\sigma$$
$$f(1 :: \sigma) \sim f\sigma$$

$$\dfrac{\dfrac{\hbox{$\mathrm{hd}\, A = \mathrm{hd}\,(fA) \mid \pi_1$} \qquad \dfrac{\hbox{$\mathrm{hd}\,(\mathrm{tl}\, A) = \mathrm{hd}\,(\mathrm{tl}\,(fA)) \mid \pi_2$} \qquad \dfrac{}{\mathrm{tl}\,(\mathrm{tl}\, A) \sim \mathrm{tl}\,(\mathrm{tl}\,(fA))}}{\mathrm{tl}\, A \sim \mathrm{tl}\,(fA)}\ \sim^{+}}{A \sim fA}\ \sim^{+}}{A \sim fA}\ \textsc{cofix}\ \pi$$

as a proof term:

cofix $\pi. (\sim^{+}\ \pi_1\ (\sim^{+}\ \pi_2\ \ \ldots))$

Guardedness and Rewritings

Prove $A \sim fA$ where Γ consists of

$$A \sim 0 :: 1 :: A \qquad\qquad f(0 :: \sigma) \sim 0 :: 1 :: f\sigma$$
$$f(1 :: \sigma) \sim f\sigma$$

$$
\dfrac{
  \dfrac{
    \operatorname{hd} A = \operatorname{hd}(fA) \mid \pi_1
    \qquad
    \dfrac{
      \operatorname{hd}(\operatorname{tl} A) = \operatorname{hd}(\operatorname{tl}(fA)) \mid \pi_2
      \qquad
      \dfrac{
        \dfrac{\pi_3}{\operatorname{tl}(\operatorname{tl} A) \sim A}
        \qquad
        \dfrac{}{A \sim \operatorname{tl}(\operatorname{tl}(fA))}
      }{\operatorname{tl}(\operatorname{tl} A) \sim \operatorname{tl}(\operatorname{tl}(fA))}\ \textsc{trans}
    }{\operatorname{tl} A \sim \operatorname{tl}(fA)}\ \sim^{+}
  }{A \sim fA}\ \sim^{+}
}{A \sim fA}\ \textsc{cofix}\ \pi
$$

as a proof term:

cofix π. $(\sim^{+}\ \pi_1\ (\sim^{+}\ \pi_2\ (\text{trans}\ \pi_3\ \ \ldots)))$

Guardedness and Rewritings

Prove $A \sim fA$ where Γ consists of

$$A \sim 0 :: 1 :: A \qquad\qquad f(0 :: \sigma) \sim 0 :: 1 :: f\sigma$$
$$f(1 :: \sigma) \sim f\sigma$$

$$\dfrac{\dfrac{\mathrm{hd}\, A = \mathrm{hd}\,(fA) \mid \pi_1 \qquad \dfrac{\mathrm{hd}\,(\mathrm{tl}\, A) = \mathrm{hd}\,(\mathrm{tl}\,(fA)) \mid \pi_2 \qquad \dfrac{\dfrac{\pi_3}{\mathrm{tl}\,(\mathrm{tl}\, A) \sim A} \qquad \dfrac{\dfrac{}{A \sim fA} \qquad \dfrac{\pi_4}{fA \sim \mathrm{tl}\,(\mathrm{tl}\,(fA))}}{A \sim \mathrm{tl}\,(\mathrm{tl}\,(fA))}\ \textsc{trans}}{\mathrm{tl}\,(\mathrm{tl}\, A) \sim \mathrm{tl}\,(\mathrm{tl}\,(fA))}\ \textsc{trans}}{\mathrm{tl}\, A \sim \mathrm{tl}\,(fA)}\ \sim^{+}}{A \sim fA}\ \sim^{+}}{A \sim fA}\ \textsc{cofix}\ \pi$$

as a proof term:

cofix π. $(\sim^{+}\ \pi_1\ (\sim^{+}\ \pi_2\ (\text{trans}\ \pi_3\ (\text{trans}\ \ldots\ \pi_4))))$

Guardedness and Rewritings

Prove $A \sim fA$ where Γ consists of

$$
A \sim 0 :: 1 :: A \qquad\qquad f(0 :: \sigma) \sim 0 :: 1 :: f\sigma
$$
$$
f(1 :: \sigma) \sim f\sigma
$$

$$
\dfrac{\dfrac{\dfrac{\pi_1}{\mathrm{hd}\,A = \mathrm{hd}\,(fA)} \quad \dfrac{\dfrac{\pi_2}{\mathrm{hd}\,(\mathrm{tl}\,A) = \mathrm{hd}\,(\mathrm{tl}\,(fA))} \quad \dfrac{\dfrac{\pi_3}{\mathrm{tl}\,(\mathrm{tl}\,A) \sim A} \quad \dfrac{\dfrac{}{\color{red}{A \sim fA}}\,\pi \quad \dfrac{\pi_4}{fA \sim \mathrm{tl}\,(\mathrm{tl}\,(fA))}}{A \sim \mathrm{tl}\,(\mathrm{tl}\,(fA))}\ \textsc{trans}}{\mathrm{tl}\,(\mathrm{tl}\,A) \sim \mathrm{tl}\,(\mathrm{tl}\,(fA))}\ \textsc{trans}}{\mathrm{tl}\,A \sim \mathrm{tl}\,(fA)}\ \sim^+}{A \sim fA}\ \sim^+}{\color{red}{A \sim fA}}\ \textsc{cofix}\ \pi
$$

as a proof term:

cofix π. $(\sim^+ \ \pi_1 \ (\sim^+ \ \pi_2 \ (\text{trans} \ \pi_3 \ (\text{trans} \ \pi \ \pi_4))))$

Guardedness and Rewritings

Prove $A \sim fA$ where Γ consists of

$$A \sim 0 :: 1 :: A \qquad\qquad f(0 :: \sigma) \sim 0 :: 1 :: f\sigma$$

$$f(1 :: \sigma) \sim f\sigma$$

$$\dfrac{\dfrac{\mathrm{hd}\,A = \mathrm{hd}\,(fA) \mid \pi_1 \qquad \dfrac{\mathrm{hd}\,(\mathrm{tl}\,A) = \mathrm{hd}\,(\mathrm{tl}\,(fA)) \mid \pi_2 \qquad \dfrac{\dfrac{\pi_3}{\mathrm{tl}\,(\mathrm{tl}\,A) \sim A} \qquad \dfrac{\dfrac{}{A \sim fA}\,\pi \qquad \dfrac{\pi_4}{fA \sim \mathrm{tl}\,(\mathrm{tl}\,(fA))}}{A \sim \mathrm{tl}\,(\mathrm{tl}\,(fA))}\,\text{TRANS}}{\mathrm{tl}\,(\mathrm{tl}\,A) \sim \mathrm{tl}\,(\mathrm{tl}\,(fA))}\,\text{TRANS}}{\mathrm{tl}\,A \sim \mathrm{tl}\,(fA)}\,\sim^+}{A \sim fA}\,\sim^+}{A \sim fA}\,\text{COFIX}\ \pi$$

obstruct guardedness!

as a proof term:

cofix $\pi.\ (\sim^+\ \pi_1\ (\sim^+\ \pi_2\ (\text{trans}\ \pi_3\ (\text{trans}\ \pi\ \pi_4))))$

Guardedness and Rewritings

Remember this?

$$
\begin{array}{rcl cc}
nats & = & 0 :: nats + ones & \lightning & \checkmark \\
from(n) & = & n :: from(n+1) & \checkmark & \checkmark \\
nats & = & from(0) & & \checkmark
\end{array}
$$

- Hide equational reasoning in the arguments!
- Prove $s \sim t$ by showing the equivalent

$$\forall uv.\,(\, u \sim s \;\Rightarrow\; t \sim v \;\Rightarrow\; u \sim v\,)$$

- New rule:

$$\frac{\forall uv.\,(\, u \sim s \;\Rightarrow\; t \sim v \;\Rightarrow\; u \sim v\,)}{s \sim t}\ \textsc{load}$$

Guardedness and Rewritings

Remember this?

$$\begin{array}{rcl}nats & = & 0 :: nats + ones \quad ↯ \quad \checkmark \\ from(n) & = & n :: from(n+1) \quad \checkmark \quad \checkmark \\ nats & = & from(0) \qquad\qquad \checkmark\end{array}$$

- Hide equational reasoning in the arguments!
- Prove $s \sim t$ by showing the equivalent

$$\forall uv.\,(\, u \sim s \;\Rightarrow\; t \sim v \;\Rightarrow\; u \sim v\,)$$

- New rule:

$$\frac{\forall uv.\,(\, u \sim s \;\Rightarrow\; t \sim v \;\Rightarrow\; u \sim v\,)}{s \sim t}\ \textsc{load}$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathrm{hd}\,u = \mathrm{hd}\,v \,|\, \pi_1 \qquad \dfrac{\mathrm{hd}\,(\mathrm{tl}\,u) = \mathrm{hd}\,(\mathrm{tl}\,v) \,|\, \pi_2 \qquad \dfrac{\dfrac{}{\mathrm{tl}\,(\mathrm{tl}\,u) \sim A \;\Rightarrow\; fA \sim \mathrm{tl}\,(\mathrm{tl}\,v) \;\Rightarrow\; \mathrm{tl}\,(\mathrm{tl}\,u) \sim \mathrm{tl}\,(\mathrm{tl}\,v)}\,\pi \qquad \dfrac{\vdots}{\mathrm{tl}\,(\mathrm{tl}\,u) \sim A}\,\textsc{trans} \qquad \dfrac{\vdots}{fA \sim \mathrm{tl}\,(\mathrm{tl}\,v)}\,\textsc{trans}}{\mathrm{tl}\,(\mathrm{tl}\,u) \sim \mathrm{tl}\,(\mathrm{tl}\,v)}\,\forall^- \Rightarrow^-}{\mathrm{tl}\,u \sim \mathrm{tl}\,v}\,\sim^+}{u \sim v}\,\sim^+}{\forall uv.\,(\,u \sim A \;\Rightarrow\; fA \sim v \;\Rightarrow\; u \sim v\,)}\,\forall^+ \Rightarrow^+}{\forall uv.\,(\,u \sim A \;\Rightarrow\; fA \sim v \;\Rightarrow\; u \sim v\,)}\,\textsc{cofix}\ \pi}{A \sim fA}\,\textsc{load}
$$

load $A\ (fA)\ (\text{cofix}\ \pi.\,(\lambda uv\rho_u\rho_v.\,(\sim^+\ \pi_1\ (\sim^+\ \pi_2\ (\pi\ (\mathrm{tl}\,(\mathrm{tl}\,u))\ (\mathrm{tl}\,(\mathrm{tl}\,v))\ \ldots\ \pi_3\ \pi_4)))))$

Guardedness and Rewritings

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\pi_1}{\mathrm{hd}\,u = \mathrm{hd}\,v} \quad \dfrac{\dfrac{\pi_2}{\mathrm{hd}\,(\mathrm{tl}\,u) = \mathrm{hd}\,(\mathrm{tl}\,v)} \quad \dfrac{\dfrac{}{\mathrm{tl}\,(\mathrm{tl}\,u) \sim A \;\Rightarrow\; fA \sim \mathrm{tl}\,(\mathrm{tl}\,v) \;\Rightarrow\; \mathrm{tl}\,(\mathrm{tl}\,u) \sim \mathrm{tl}\,(\mathrm{tl}\,v)}\,{\color{red}\pi} \quad \dfrac{\vdots}{\mathrm{tl}\,(\mathrm{tl}\,u) \sim A}\,{\color{red}\mathrm{TRANS}} \quad \dfrac{\vdots}{fA \sim \mathrm{tl}\,(\mathrm{tl}\,v)}\,{\color{red}\mathrm{TRANS}}}{\mathrm{tl}\,(\mathrm{tl}\,u) \sim \mathrm{tl}\,(\mathrm{tl}\,v)}\,\forall^- \Rightarrow^-}{\mathrm{tl}\,u \sim \mathrm{tl}\,v}\,\sim^+}{u \sim v}\,\sim^+}{\forall uv.\,(\,u \sim A \;\Rightarrow\; fA \sim v \;\Rightarrow\; u \sim v\,)}\,\forall^+ \Rightarrow^+}{{\color{red}\forall uv.\,(\,u \sim A \;\Rightarrow\; fA \sim v \;\Rightarrow\; u \sim v\,)}}\,\mathrm{COFIX}\ \pi}{A \sim fA}\,\mathrm{LOAD}
$$

load A (fA) $({\color{red}\mathsf{cofix}\ \pi}.\,(\lambda uv\rho_u\rho_v.\,(\sim^+\ \pi_1\ (\sim^+\ \pi_2\ ({\color{red}\pi}\ (\mathrm{tl}\,(\mathrm{tl}\,u))\ (\mathrm{tl}\,(\mathrm{tl}\,v))\ \ldots\ \pi_3\ \pi_4)))))$

Guardedness and Rewritings

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{}{\mathrm{tl}\,(\mathrm{tl}\,u) \sim A \;\Rightarrow\; fA \sim \mathrm{tl}\,(\mathrm{tl}\,v) \;\Rightarrow\; \mathrm{tl}\,(\mathrm{tl}\,u) \sim \mathrm{tl}\,(\mathrm{tl}\,v)}\,\pi
        \quad
        \dfrac{\vdots}{\mathrm{tl}\,(\mathrm{tl}\,u) \sim A}\,\textsc{trans}
        \quad
        \dfrac{\vdots}{fA \sim \mathrm{tl}\,(\mathrm{tl}\,v)}\,\textsc{trans}
      }{
        \mathrm{tl}\,(\mathrm{tl}\,u) \sim \mathrm{tl}\,(\mathrm{tl}\,v)
      }\,\forall^- \Rightarrow^-
      \quad \mathrm{hd}\,(\mathrm{tl}\,u) = \mathrm{hd}\,(\mathrm{tl}\,v)|\pi_2
    }{
      \mathrm{tl}\,u \sim \mathrm{tl}\,v
    }\,\sim^+
    \quad \mathrm{hd}\,u = \mathrm{hd}\,v|\pi_1
  }{
    \dfrac{
      \dfrac{
        u \sim v
      }{
        \forall uv.\,(\,u \sim A \;\Rightarrow\; fA \sim v \;\Rightarrow\; u \sim v\,)
      }\,\forall^+ \Rightarrow^+
    }{
      \forall uv.\,(\,u \sim A \;\Rightarrow\; fA \sim v \;\Rightarrow\; u \sim v\,)
    }\,\textsc{cofix}\;\pi
  }\,\sim^+
}{
  A \sim fA
}\,\textsc{load}
$$

$$\text{load } A\ (fA)\ (\text{cofix } \pi.\,(\lambda uv\rho_u\rho_v.\,(\sim^+\ \pi_1\ (\sim^+\ \pi_2\ (\pi\ (\mathrm{tl}\,(\mathrm{tl}\,u))\ (\mathrm{tl}\,(\mathrm{tl}\,v))\ \ldots\ \pi_3\ \pi_4)))))$$

- We only know about u and v what we needed to know in the previous proof.

Intuition

$$
\begin{prooftree}
\AxiomC{π_1}
\UnaryInfC{$\mathrm{hd}\,A = \mathrm{hd}\,(fA)$}
\AxiomC{π_2}
\UnaryInfC{$\mathrm{hd}\,(\mathrm{tl}\,A) = \mathrm{hd}\,(\mathrm{tl}\,(fA))$}
\AxiomC{π_3}
\UnaryInfC{$\mathrm{tl}\,(\mathrm{tl}\,A) \sim A$}
\AxiomC{}
\RightLabel{π}
\UnaryInfC{$A \sim fA$}
\AxiomC{π_4}
\UnaryInfC{$fA \sim \mathrm{tl}\,(\mathrm{tl}\,(fA))$}
\RightLabel{TRANS}
\BinaryInfC{$A \sim \mathrm{tl}\,(\mathrm{tl}\,(fA))$}
\RightLabel{TRANS}
\BinaryInfC{$\mathrm{tl}\,(\mathrm{tl}\,A) \sim \mathrm{tl}\,(\mathrm{tl}\,(fA))$}
\RightLabel{$\sim^+$}
\BinaryInfC{$\mathrm{tl}\,A \sim \mathrm{tl}\,(fA)$}
\RightLabel{$\sim^+$}
\BinaryInfC{$A \sim fA$}
\RightLabel{COFIX π}
\UnaryInfC{$A \sim fA$}
\end{prooftree}
$$

What we did in the original proof

- Rewrite the left term to A.
- Rewrite the right term to fA.

What we loaded

$$\forall uv.\,(\,u \sim A \;\Rightarrow\; fA \sim v \;\Rightarrow\; u \sim v\,)$$

Bisimulation-up-to

Idea from process algebra (Milner, Sangiorgi):

- Instead of proving $s \sim t$, define a relation R such that $s \; R \; t$ and prove $R \subseteq \sim$.
- Instead of proving $R \subseteq \sim$, let's prove $\mathcal{F}(R) \subseteq \sim$.

Bisimulation-up-to

Idea from process algebra (MILNER, SANGIORGI):

- Instead of proving $s \sim t$, define a relation R such that $s \; R \; t$ and prove $R \subseteq \sim$.
- Instead of proving $R \subseteq \sim$, let's prove $\mathcal{F}(R) \subseteq \sim$.

Definition

$$\mathcal{F}(R) \;::=\; R \mid \sim \mid \underbrace{\mathcal{F}(R)^{-1}}_{\text{Symmetry}} \mid \underbrace{\mathcal{F}(R)\mathcal{F}(R)}_{\text{Transitivity}}$$

Bisimulation-up-to

We want to prove $\mathcal{F}(R) \subseteq \sim$.

Definition

$$R \text{ progresses to } R' \iff s\, R\, t \Rightarrow s(0) = t(0) \wedge s'\; R'\; t'$$

R is a *bisimulation-up-to* $\mathcal{F}$ if R progresses to $\mathcal{F}(R)$.

Theorem (meta)

If R is a bisimulation-up-to $\mathcal{F}$, then $\mathcal{F}(R)$ is a bisimulation.

The loading technique is an instance of this, using $\sim R \sim \subseteq \mathcal{F}(R)$.

Bisimulation-up-to

We want to prove $\mathcal{F}(R) \subseteq\sim$.

Definition

$$R \text{ progresses to } R' \iff s\,R\,t \Rightarrow s(0) = t(0) \wedge s'\;R'\;t'$$

R is a *bisimulation-up-to* $\mathcal{F}$ if R progresses to $\mathcal{F}(R)$.

Theorem (meta)

If R is a bisimulation-up-to $\mathcal{F}$, then $\mathcal{F}(R)$ is a bisimulation.

The loading technique is an instance of this, using $\sim R \sim \subseteq \mathcal{F}(R)$.

What About Rewriting Under a Context?

How does a rule like

$$\frac{\pi : s \sim t}{C[\pi] : C[s] \sim C[t]}\ \textsc{context}$$

combine with coinduction?

- Taking $C = \mathrm{tl}\,\Box$ is *not* productive.
- When is $\pi = \sim^{+}$ $(\mathrm{refl}_{=}\ 0)$ $(\ldots\ C[\pi]\ldots) : s \sim t$ productive?
- When C is causal, this is correct.
  Moreover, then $X = 0 :: C[X]$ is productive.

What About Rewriting Under a Context?

How does a rule like

$$\frac{\pi : s \sim t}{C[\pi] : C[s] \sim C[t]}\ \textsc{context}$$

combine with coinduction?

- Taking $C = \mathrm{tl}\,\square$ is *not* productive.
- When is $\pi = \sim^{+}\ (\mathsf{refl}_{=}\ 0)\ (\ldots\ C[\pi]\ldots) : s \sim t$ productive?
- When C is causal, this is correct.
  Moreover, then $X = 0 :: C[X]$ is productive.

What About Rewriting Under a Context?

How does a rule like

$$\frac{\pi : s \sim t}{C[\pi] : C[s] \sim C[t]} \text{ CONTEXT}$$

combine with coinduction?

- Taking $C = \mathrm{tl}\,\square$ is *not* productive.
- When is $\pi = \sim^{+}$ (refl$_{=}$ 0) $(\ldots\ C[\pi]\ldots) : s \sim t$ productive?
- When C is causal, this is correct.
  Moreover, then $X = 0 :: C[X]$ is productive.

Causal Contexts

s and t are bisimilar up to depth n:

$$s \sim_n t \iff \forall k < n.\, s(k) = t(k)$$

Definition

A stream function $f \colon A^\omega \to B^\omega$ is *causal* if

$$s \sim_n t \implies fs \sim_n ft$$

for all $s, t \in A^\omega$ and $n \in \mathbb{N}$.

Let Γ be a set of equations. A *stream context* C is *causal* if $[\![C, \alpha]\!]_{\mathcal{A}}$ is causal for all models $\mathcal{A}$ of Γ, and assignments $\alpha : \mathcal{X} \to A$.

We Can Extend the Theorem

We can add causal context to $\mathcal{F}(R)$

$$\mathcal{F}(R) ::= R \mid \sim \mid C[\mathcal{F}(R)] \mid \mathcal{F}(R)^{-1} \mid \mathcal{F}(R)\mathcal{F}(R)$$

with C causal context.

R is a *bisimulation-up-to* if R progresses to $\mathcal{F}(R)$.

And the theorem holds

If R is a bisimulation-up-to, then $\mathcal{F}(R)$ is a bisimulation.

This Proves the Soundness of This System

- Γ, Δ sets of equations, Δ is the set of coinduction hypotheses.

Equational Reasoning

$$\frac{}{\Gamma, \Delta \vdash C[s^\sigma] \sim C[t^\sigma]} \text{ if } s \sim t \in \Gamma$$

$$\frac{}{\Gamma, \Delta \vdash s \sim s} \qquad \frac{\Gamma, \Delta \vdash t \sim s}{\Gamma, \Delta \vdash s \sim t} \qquad \frac{\Gamma, \Delta \vdash s \sim u \quad \Gamma, \Delta \vdash u \sim t}{\Gamma, \Delta \vdash s \sim t}$$

Coinduction

$$\frac{}{\Gamma, \Delta \vdash C[s^\sigma] \sim C[t^\sigma]} \text{ if } s \sim t \in \Delta \text{ and } C \text{ is causal}$$

$$\frac{\Gamma, \varnothing \vdash \mathrm{hd}\, s = \mathrm{hd}\, t \quad \Gamma, \Delta \cup \{s \sim t\} \vdash \mathrm{tl}\, s \sim \mathrm{tl}\, t}{\Gamma, \Delta \vdash s \sim t} \text{ coin}$$

- NB. without causality $\varnothing, \{s \sim t\} \vdash \mathrm{tl}\, s \sim \mathrm{tl}\, t$ can always be derived!

This Proves the Soundness of This System

- Γ, Δ sets of equations, Δ is the set of coinduction hypotheses.

Equational Reasoning

$$\frac{}{\Gamma, \Delta \vdash C[s^\sigma] \sim C[t^\sigma]} \text{ if } s \sim t \in \Gamma$$

$$\frac{}{\Gamma, \Delta \vdash s \sim s} \qquad \frac{\Gamma, \Delta \vdash t \sim s}{\Gamma, \Delta \vdash s \sim t} \qquad \frac{\Gamma, \Delta \vdash s \sim u \quad \Gamma, \Delta \vdash u \sim t}{\Gamma, \Delta \vdash s \sim t}$$

Coinduction

$$\frac{}{\Gamma, \Delta \vdash C[s^\sigma] \sim C[t^\sigma]} \text{ if } s \sim t \in \Delta \text{ and } C \text{ is causal}$$

$$\frac{\Gamma, \varnothing \vdash \mathrm{hd}\, s = \mathrm{hd}\, t \quad \Gamma, \Delta \cup \{s \sim t\} \vdash \mathrm{tl}\, s \sim \mathrm{tl}\, t}{\Gamma, \Delta \vdash s \sim t} \text{ coin}$$

- NB. without causality $\varnothing, \{s \sim t\} \vdash \mathrm{tl}\, s \sim \mathrm{tl}\, t$ can always be derived!

Conclusion

- We defined a system of axioms, mixing equationnal and corecursive reasoning.
- We proved this system sound.
- There is a systematic way to convert a proof in this system to a proof accepted by COQ.
- We provide a HASKELL implementation: `http://www.cs.vu.nl/~diem/research/up_to.tgz`
- This can easily be generalised to other coinductive structures.

Conclusion

- We defined a system of axioms, mixing equationnal and corecursive reasoning.
- We proved this system sound.
- There is a systematic way to convert a proof in this system to a proof accepted by Coq.
- We provide a Haskell implementation:
  `http://www.cs.vu.nl/~diem/research/up_to.tgz`
- This can easily be generalised to other coinductive structures.

Thank you for listening!

$$\mathit{even}\, s = \mathit{hd} s :: \mathit{even}\, (\mathrm{tl}\, (\mathrm{tl}\, s))$$

$$\forall s t, s \sim t \implies \mathit{even}\, s \sim \mathit{even}\, t$$

```
import Prelude hiding (head, tail, Left, Right, flip, id)
import qualified Data.Map as Map
import Lang

zeros1 = Fun "zeros1" []
zeros2 = Fun "zeros2" []

env :: Environment
env = (
    Map.fromList [
      ("zeros1", ([], Stream, False)),
      ("zeros2", ([], Stream, False))
    ],
    hypFromList [
      ("hyp_zeros1", (zeros1, cons zero zeros1, Stream)),
      ("hyp_zeros2", (zeros2, cons zero zeros2, Stream))
    ]
  )

lemma = ("zeros1_eq_zero2", proof,
    (zeros1, zeros2, Stream))
```

```
proof :: BisProof
proof = Cofix "F" (Eq2Bis e1) (Eq2Bis h1)

-- e1 = ...

h1 = Transitivity step1 h2
step1 = (Step "hyp_zeros1" Right (CFun "tail" [] Hole []) Map.empty)

h2 = Transitivity step2 h3
step2 = (Step "hyp_tail" Right Hole (Map.fromList [("x", zero), ("σ", zeros1)]))

h3 = Transitivity step3 h4
step3 = (Step "F" Right Hole Map.empty)

h4 = Transitivity step4 h5
step4 = (Step "hyp_tail" Left Hole (Map.fromList [("x", zero), ("σ", zeros2)]))

h5 = Transitivity step5 h6
step5 = (Step "hyp_zeros2" Left (CFun "tail" [] Hole []) Map.empty)

h6 = Reflexivity
```