Circular Behavioral Reasoning

Grigore Roşu Dorel Lucanu

UIUC UAIC

07/11/2011, Core Workshop, CWI, Amsterdam

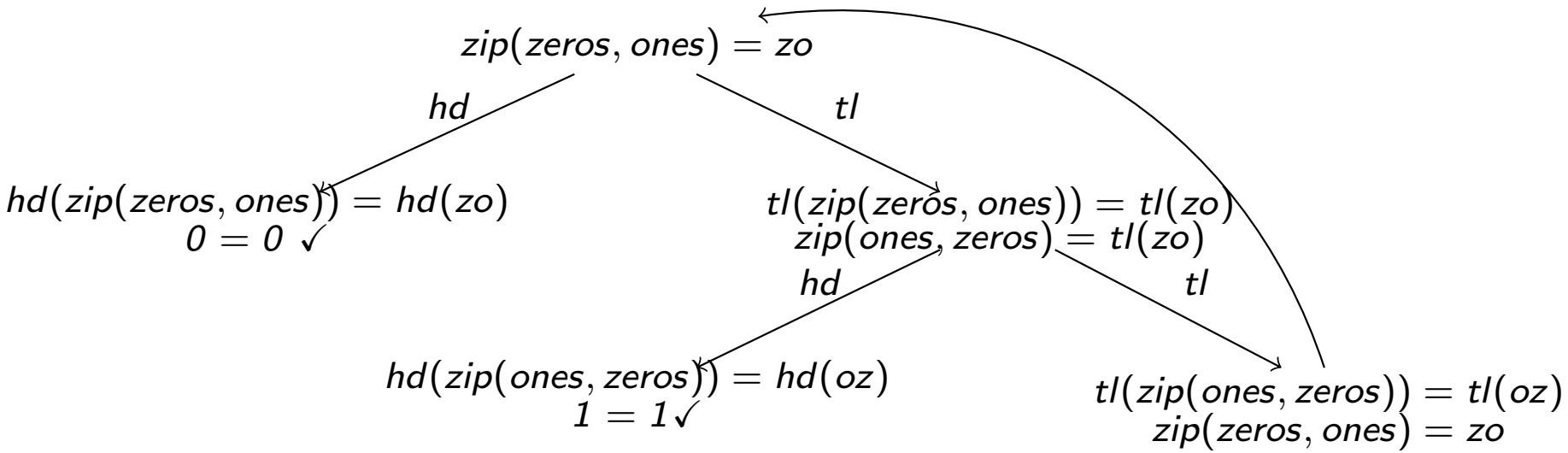
Plan

Intuitive proof by circular coinduction

$$zeros = 0 : zeros \qquad ones = 1 : ones \qquad zip(a : S_1, S_2) = a : zip(S_2, S_1) \qquad zo = 0 : 1 : zo$$

$$hd(zeros) = 0 \qquad hd(ones) = 1 \qquad hd(zip(S, S')) = hd(S) \qquad hd(zo) = 0$$
$$tl(zeros) = zeros \qquad tl(ones) = ones \qquad tl(zip(S, S')) = zip(S', tl(S)) \qquad hd(tl(zo)) = 1$$
$$tl(tl(zo)) = zo$$

$$\vdash zip(zeros, ones) = zo$$



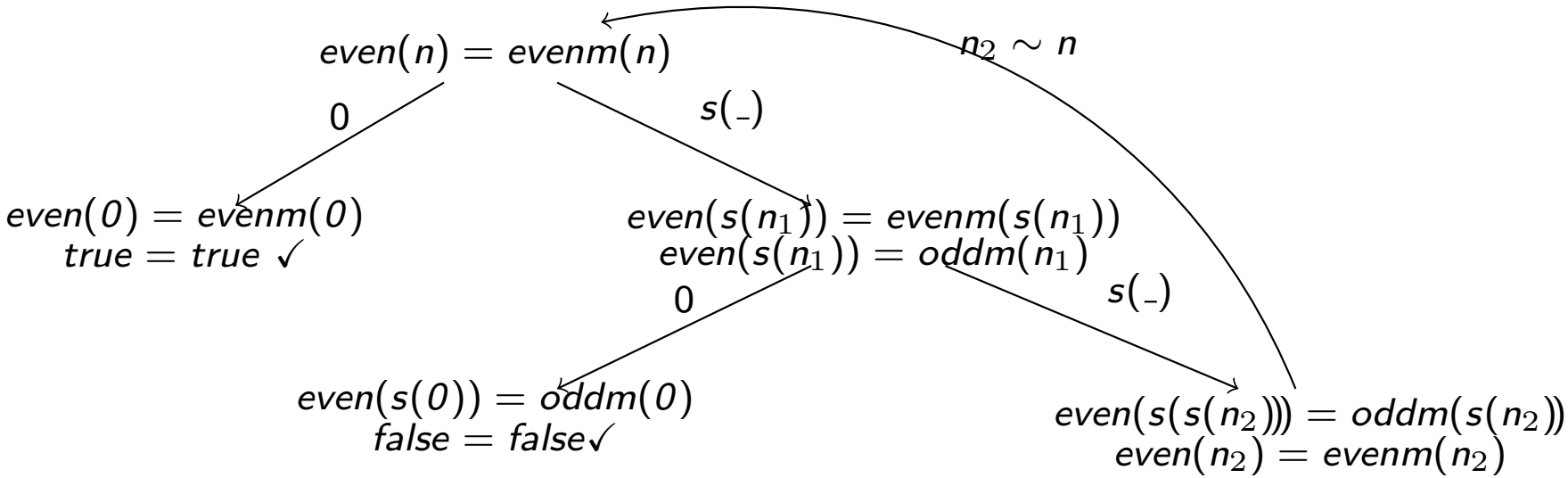
Intuitive proof by circular induction

$even(0) = true$ $\quad$ $evenm(0) = true$ $\quad$ $oddm(0) = true$

$even(s(0)) = false$ $\quad$ $evenm(s(N)) = oddm(N)$ $\quad$ $oddm(s(N)) = evenm(N)$

$even(s(s(N))) = even(N)$

$$\vdash (\forall N)\, even(N) = evenm(N)$$



Challenge

Is there any abstract formalism such that the two proving methods are instances of it?

What is common for and what differentiate the two proving methods?

What really means inductive and coinductive in this context?

behavioral reasoning more abstract than coinductive reasoning? (it seems so ...)

Our Approach 1/2

- define abstract behavioral specifications, which includes
  - sentences (formulas),
  - experiments defined as compounds of derivatives, and
  - an entailment relation
- define the behavioral equivalence as indistinguishabilty under experiments

Our Approach 2/2

- define abstract circular reasoning proof systems, which
  - uses frozen versions of the goals to be proved as hypothesis (circular principle)
- extends the circular reasoning proof systems with special hypotheses and equational interpolants
- show that the coinductive properties and the inductive properties are behavioral
- consequently, we get circular reasoning systems for coinduction and induction in the same formalism

Plan

Behavioral Specification

a behavioral specification $\mathcal{B}$ consists of:

1. a specification (Σ, E)
   Σ = signature (operations), E = sentences (axioms)

2. quasi-experiments Qxp and experiments Exp: Exp $\subseteq$ Qxp
   1. $C \in$ Qxp defines a partial sentence transformer $e \mapsto C[e]$
      ($C[e]$ could be a set of sentences)
   2. quasi-experiments splits the sentences in derivable and non-derivable:
      e is derivable if there is C s. t. $C[e]$ is defined
   3. C an experiment implies $C[e]$ not derivable

3. an entailment relation $E \vdash e$ that is reflexive, transitive, monotonic, and compatible with the quasi-experiments: $E \vdash e$ implies $E \vdash C[e]$ for each quasi-experiment C.

Behavioral Equivalence

$\mathcal{B}$ behaviorally entails e, write $\mathcal{B} \Vdash e$, iff

- $\mathcal{B} \vdash e$, if e is non-derivable, and
- $\mathcal{B} \vdash C[e]$ for each experiment $C \in \text{Exp}$, if e is derivable.

Behavioral equivalence of $\mathcal{B}$ is the set

$$\equiv \quad = \quad \{e \mid \mathcal{B} \Vdash e\}$$

No derivatives up to now!

The behavioral equivalence is the largest behaviorally closed set

Assumptions:

1. there is a well-founded partial order $\prec$ over quasi-experiments
2. derivative $=$ a minimal (non-identity) quasi-experiment
3. C non-derivative $\Rightarrow$ $C = C_1[C_2]$ and $C_1, C_2 \prec C_1[C_2]$

Notation: $\Delta =$ the set of derivatives (generators of quasi-experiments)

A set of equations $\mathcal{G}$ is behaviorally closed iff

$\mathcal{B} \vdash$ nonderivable($\mathcal{G}$) and

$\Delta(\mathcal{G} - \mathcal{B}^\bullet) \subseteq \mathcal{G}$, where $\mathcal{B}^\bullet = \{e \mid \mathcal{B} \vdash e\}$

Theorem

For any behavioral specification $\mathcal{B}$, the behavioral equivalence $\equiv$ is the largest behaviorally closed set of sentences.

First Instance: Coinductive Behavioral Reasoning

- sentences: $(\forall X)\, t = t' \text{ if } \wedge_{i \in I}\, u_i = v_i$
- derivatives (destructors): Σ-contexts $\delta[*{:}h]$
  e.g., $hd(*{:}Stream)$, $tl(*{:}Stream)$
- quasi-experiments: Δ-contexts
  e.g., $hd(tl(*{:}Stream))$, $tl(tl(*{:}Stream))$
- quasi-experiments splits the sorts in hidden and visible:
  the sort h is hidden, e.g., $Stream$
  remaining sorts are visible, or data sorts, e.g., Bit, Nat, Int
- experiment $=$ visible Δ-context
- behavioral equivalence: $\mathcal{B} \Vdash e$ iff
  $\mathcal{B} \vdash e$ if e is visible, and
  $(\forall C \in \mathrm{Exp})\, \mathcal{B} \vdash C[e]$ if e is hidden
  e.g., $STREAM \Vdash S = S'$ iff
  $STREAM \vdash hd(tl^i(S)) = hd(tl^i(S')), i = 0, 1, 2, \ldots$

Streams as a CIRC theory

```
theory BIT is
  sort Bit .
  ops 0 1 : -> Bit .
  ...
endtheory

theory BITSTREAM is including BIT .
  sort Stream .
  vars S S' : Stream .

  --- destructors
  op hd : Stream -> Bit .
  op tl : Stream -> Stream .

  --- zip of streams
  op zip : Stream Stream -> Stream .
  eq hd(zip(S,S')) = hd(S) .
  eq tl(zip(S,S')) = zip(S',tl(S)) .
  ...
  --- derivatives
  derivative hd(*:Stream) .
  derivative tl(*:Stream) .
endtheory
```

Second Instance: Inductive Behavioral Reasoning

- sentences: $(\forall Y)(\forall Z)\, t = t' \texttt{ if } \wedge_{i\in I}\, u_i{=}v_i$ (Y = inductive variables)
- experiments: constructor ground substitutions $\theta : Y \to \mathcal{T}_{\Sigma^{ctor}}$
- quasi-experiment: $\theta : Y_0 \to \mathcal{T}_{\Sigma^{ctor}}(Y' \cup Z')$, $Y_0 \subseteq Y$
- θ defines an equation transformer $e \mapsto \theta[e]$, where $\theta[e]$ is $(\forall \ldots)\theta(t){=}\theta(t') \texttt{ if } \wedge_{i\in I}\theta(u_i){=}\theta(v_i)$ whenever $Y_0 \cap Y \neq \emptyset$
- Each constructor $c \in \Sigma^{ctor}$ defines a derivative $\delta_c = \{\delta_{c,y}\}$:
  - $\delta_{c,y}$ is $y \mapsto c$ if c is a constant constructor
  - $\delta_{c,y}$ is the quasi-experiment $y \mapsto c(y_1, \ldots, y_n)$ if c is a non-constant constructor
  - if e is $(\forall Y)e'$, then $\delta_c[e] = \{\delta_{c,y}[e] \mid y \in Y\}$
- behavioral equivalence: $\mathcal{B} \Vdash e$ iff

  $\mathcal{B} \vdash e$ if e is ground, and

  $(\forall \theta \in \mathrm{Exp})\, \mathcal{B} \vdash \theta[e]$ if e is not ground

Inductive Behavioral Entailment - Example

- naturals with the constructors 0 and $s(\_)$
- e: $(\forall P)(\forall M, N)sum(sum(M, N), P)) = sum(M, sum(N, P))$
- experiments: $P \mapsto 0$, $P \mapsto s(0)$,$P \mapsto s(s(0))$,...
- quasi-experiments: $P \mapsto s(P')$, $P \mapsto s(s(P'))$, ...
- derivatives:
  δ_0 is $P \mapsto 0$ and $\delta_0[e]$ is
  $(\forall M, N)sum(sum(M, N), 0)) = sum(M, sum(N, 0))$

  δ_s is $P \mapsto s(P')$ and $\delta_s[e]$ is
  $(\forall P')(\forall M, N)sum(sum(M, N), s(P'))) = sum(M, sum(N, s(P')))$
- $NAT \Vdash sum(sum(M, N), P)) = sum(M, sum(N, P))$ iff
  $NAT \vdash sum(sum(M, N), s^i(0)) = sum(M, sum(N, s^i(0)))$,
  $i = 0, 1, 2, \ldots$

Naturals as a CIRC theory

. . . just a Maude specification. . .

```
theory NAT is
  including BOOL .

  sort Nat .

  op 0 :      -> Nat [ctor] .
  op s : Nat -> Nat [ctor] .

  var M N : Nat .

  op sum : Nat Nat -> Nat .
  eq sum(M, 0) = M .
  eq sum(M, s(N)) = s(sum(M, N)) .
  ...
endtheory
```

Plan

Behavioral Specification with Freezing

a behavioral specification with freezing is

a behavioral specification $\mathcal{B}$ + a freezing operator $e \mapsto \boxed{e}$

that splits the sentences in frozen and unfrozen s.t.:

(A0) $E \cup \{\boxed{f}\} \vdash \boxed{C[f]}$ iff $E \vdash \boxed{C[f]}$ for any quasi-experiment C (f derivable unfrozen);

(A1) $E \cup \boxed{F} \vdash \boxed{e}$ iff $E \vdash e$ (e non-derivable unfrozen);

(A2) $E \cup \boxed{F} \vdash \boxed{G}$ implies $E \cup C[\boxed{F}] \vdash C[\boxed{G}]$ for any quasi-experiment C.

Theorem (**circularity principle**)

If $\mathcal{B}$ is a behavioral specification and F is a set of derivable sentences with $\mathcal{B} \cup \boxed{F} \vdash \boxed{\Delta[F]}$ then $\mathcal{B} \Vdash F$.

(Abstract) Circular Reasoning Proof System

$$\frac{\cdot}{\mathcal{B} \cup \boxed{F} \Vdash^{\circlearrowleft} \emptyset} \quad \text{[Done]}$$

$$\frac{\mathcal{B} \cup \boxed{F} \Vdash^{\circlearrowleft} \boxed{G}, \quad \mathcal{B} \cup \boxed{F} \vdash \boxed{e}}{\mathcal{B} \cup \boxed{F} \Vdash^{\circlearrowleft} \boxed{G} \cup \{\boxed{e}\}} \quad \text{[Reduce]}$$

$$\frac{\mathcal{B} \cup \boxed{F} \cup \{\boxed{e}\} \Vdash^{\circlearrowleft} \boxed{G} \cup \boxed{\Delta[e]}}{\mathcal{B} \cup \boxed{F} \Vdash^{\circlearrowleft} \boxed{G} \cup \{\boxed{e}\}}, \quad \text{if } e \text{ derivable} \quad \text{[Derive]}$$

Theorem (soundness of abstract circular reasoning)

If $\mathcal{B} \Vdash^{\circlearrowleft} \boxed{G}$ *is inferable using the above proof system, then* $\mathcal{B} \Vdash G$.

Circular Reasoning Explained

initial you want to show $E \Vdash e$

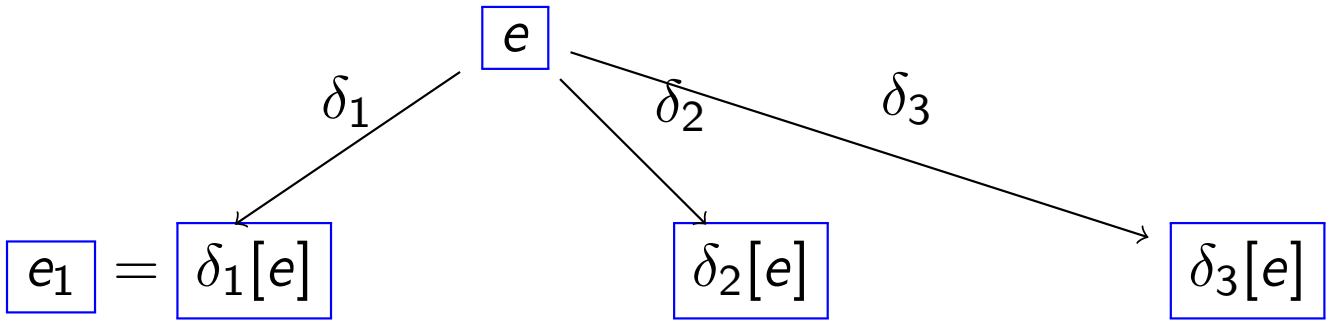
E

$\boxed{e}$

Circular Reasoning Explained

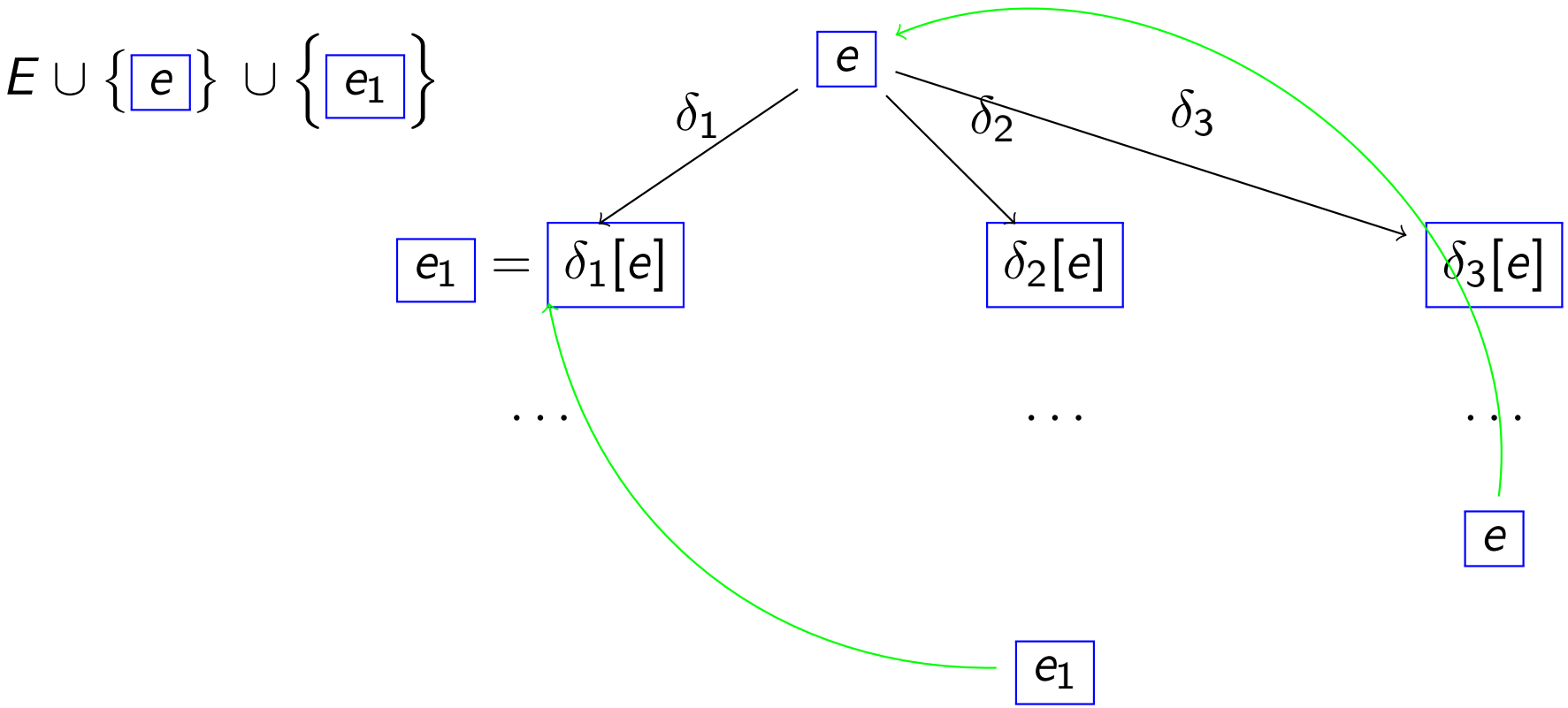
initial you want to show $E \Vdash e$

$E \cup \{\boxed{e}\}$



Circular Reasoning Explained

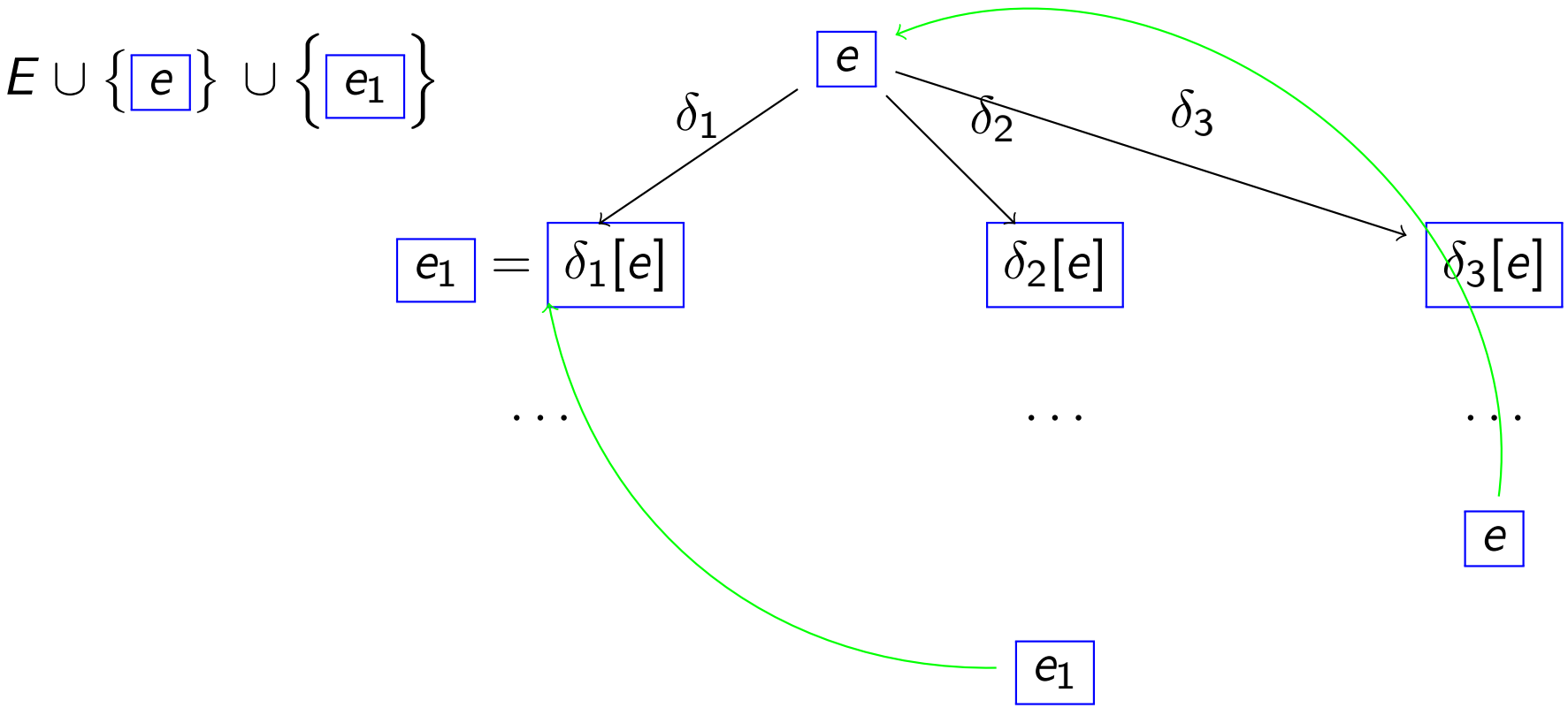
initial you want to show $E \Vdash e$



Circular Reasoning Explained

initial you want to show $E \Vdash e$

. . . and at the end you get $E \Vdash \{e, e_1, \ldots\}$



Plan

SpecialContexts and Special Hypotheses

- some times the constraint $\{\boxed{f}\} \nvdash \boxed{C[f]}$ is too strong
- there are quasi-experiments under which the entailment is safe
- a sentence $\boxed{C[f]}$ for such a C is a particular special hypothesis

e is a special hypothesis for F iff $(\forall C)\, \mathcal{B} \vdash C[e]$ whenever $\mathcal{B} \vdash C^{\leq}[F]$ where $C^{\leq}$ is $\{D \mid |D| \leq |C|\}$.

$F^{\leq}$ = the set of special hypotheses for F

$(F \subseteq F^{\leq})$

$F_1 \subseteq F_2$ implies $F_1^{\leq} \subseteq F_2^{\leq}$

$(F^{\leq})^{\leq} = F^{\leq}$

$\text{derivable}(\equiv)^{\leq} = \text{derivable}(\equiv)$ and $\text{derivable}(\equiv) \subseteq F^{\leq}$

Circular Reasoning Extended with Special Hypotheses

- it is safe to use the special hypotheses during a proof

Theorem (extended circularity principle)

If $\mathcal{B} \cup \boxed{F^{\leq}} \vdash \boxed{\Delta[F]}$, *then* $\mathcal{B} \Vdash F^{\leq}$ *(in fact,* $F^{\leq} = \text{derivable}(\equiv)$*).*

- so, we may extend the circular resoning proof system

$$\frac{\mathcal{B} \cup \mathcal{F} \cup \{\boxed{e}\} \cup \boxed{K} \Vdash^{\circlearrowleft} \mathcal{G} \cup \boxed{\Delta[e]}}{\mathcal{B} \cup \mathcal{F} \Vdash^{\circlearrowleft} \mathcal{G} \cup \{\boxed{e}\}}, \quad \text{when } e \text{ is derivable and } K \subseteq e^{\leq} \qquad [\text{Derive}^{\text{scx}}]$$

Theorem (soundness of circular reasoning with special hypotheses)

If $\mathcal{B} \Vdash^{\circlearrowleft} \boxed{G}$ *is inferable using the proof system extended with rule* [Derive$^{\text{scx}}$]*, then* $\mathcal{B} \Vdash G$.

Plan

First instance: Circular Coinduction

we only need to define the freezing:

- extend Σ with a new sort *Frozen* and a new operation $\boxed{\text{-}} : s \to \mathit{Frozen}$ for each sort s
- if t is a term, then $\boxed{t}$ is the frozen (form of) t
- a frozen equation $(\forall X)\ \boxed{t} = \boxed{t'}$ `if` c is obtained from a Σ-equation $(\forall X)\ t = t'$ `if` c by freezing its lhs and rhs, whereas the condition c stays unfrozen
  (note that we only assume visible conditions)
- the equations over the original signature Σ are unfrozen equations

circular coinduction is the circular reasoning proof system for coinductive behavioral specs and using the above freezing operator

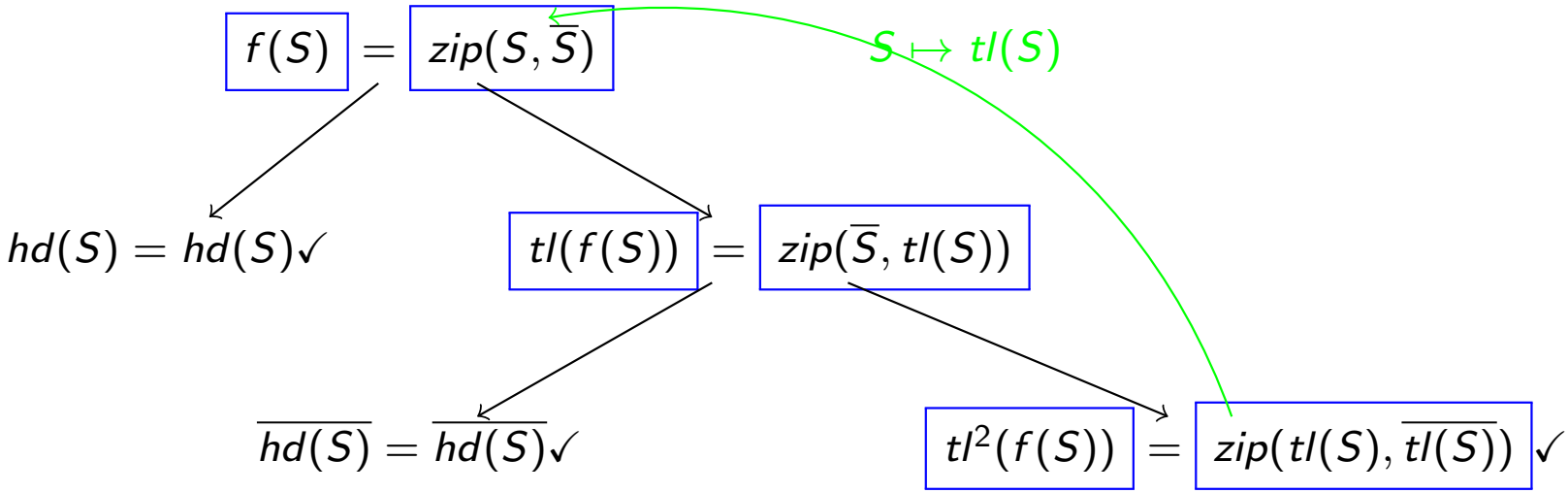
Circular Coinduction:Example

$$E \Vdash f(S) = zip(S, \overline{(S)})$$

E

$\overline{a : S} = \overline{a} : \overline{S}$
$f(a : S) = a : \overline{a} : f(S)$
$zip(a : S_1, S_2) = a : zip(S_2, S_1)$

$$E \cup \{ f(S) = zip(S, \overline{S}) \} \cup \{ \{ tl(f(S)) = zip(\overline{S}, tl(S)) \}$$



Circular Coinduction:Example in Circ

Input:

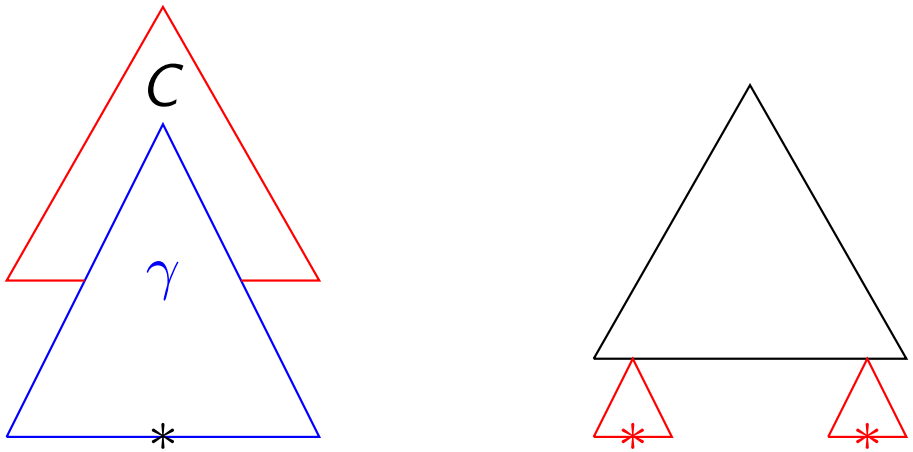
```
(add goal f(S:Stream) = zip(S:Stream, not(S:Stream)) .)
(coinduction .)
```

Output:

```
Goal added: f(S:Stream) = zip(S:Stream,not(S:Stream))
Proof succeeded.
Number of derived goals: 4
Number of proving steps performed: 22
Maximum number of proving steps is set to: 256
Proved properties:
tl(f(S:Stream)) = zip(not(S:Stream),tl(S:Stream))
f(S:Stream) = zip(S:Stream,not(S:Stream))
```

Special Contexts for Coinduction

a ontext $\gamma[*{:}h]$ is special iff for any experiment C for γ there is t s. t, $\mathcal{B} \vdash C[\gamma[*{:}h]] = t$ and each occurrence of $*{:}h$ in t appears only in a subterm in $C^{\leq}$



Theorem

If F is a hidden equation set and γ a special context, $\gamma[F] \subseteq F^{\leq}$.

Special Contexts in Circ

```
Introduced theory BITSTREAM
rewrites: 38 in 1ms cpu (0ms real) (38000 rewrites/second)
Contexts will be automatically computed.
Initializing ...
The special contexts are:
not(*:Stream)
zip(*:Stream,V#2:Stream)
zip(V#1:Stream,*:Stream)
```

Plan

Second Instance: Circular Induction

- recall inductive sentences e: $(\forall Y)\,(\forall Z)\,t_1 = t_2$ if c
- the frozen form of a variable $y{:}s$ is a distinguished constant $y.s$ of sort s
- the frozen form $\boxed{e}$ of a goal is the equation obtained by freezing all inductive variables, i.e., each $y{:}s \in Y$ is replaced with $y.s$
- notation: if $\delta_{c,y}$ is the quasi-experiment $y \mapsto c(y_1, \ldots, y_n)$, then $y_i \sim y$
- if a frozen inductive variable y is substituted by y' with $y' \sim y$ and $sort(y') = sort(y)$, then the new equation is a frozen special hypothesis in $\boxed{\{e\}^{\leq}}$
- circular induction: take $\mathcal{K}$ in the rule Derive^{scx} equal to the special hypotheses for $\boxed{e}$ computed via the transitive closure of $\sim$.

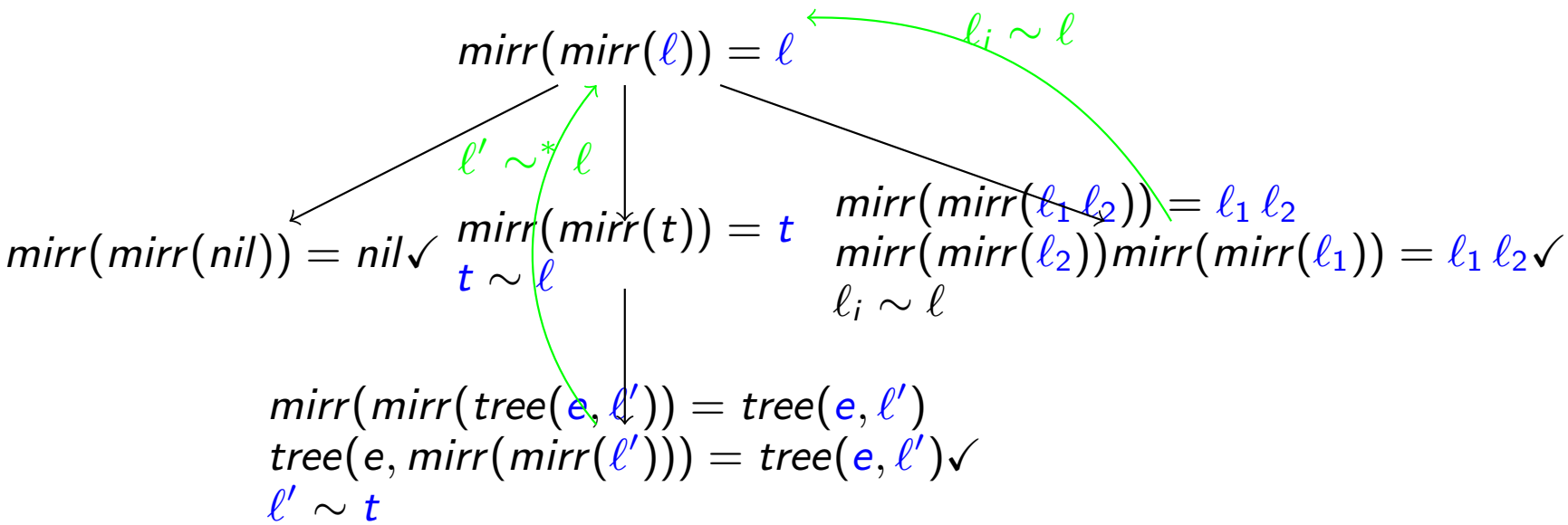
Circular Induction: Example

$E \Vdash mirr(mirr(L)) = L$

E

$Tree < List,\ tree : EltList \to Tree,\ \_\_ : List \times List \to List$
$mirr(nil) = nil \quad mirr(L_1\ L_2) = mirr(L_2)\ mirr(L_1)$
$mirr(tree(E, L)) = tree(E, mirr(L))$

$E \cup \{mirr(mirr(\ell_i)) = \ell_i \mid i = 1, 2\} \cup \{\cup\{mirr(mirr(\ell')) = \ell'\}$



Circular Induction: Example in Circ

Input:

```
(add goal mirror(mirror(L:TList)) = L:TList .)
(induction .)
```

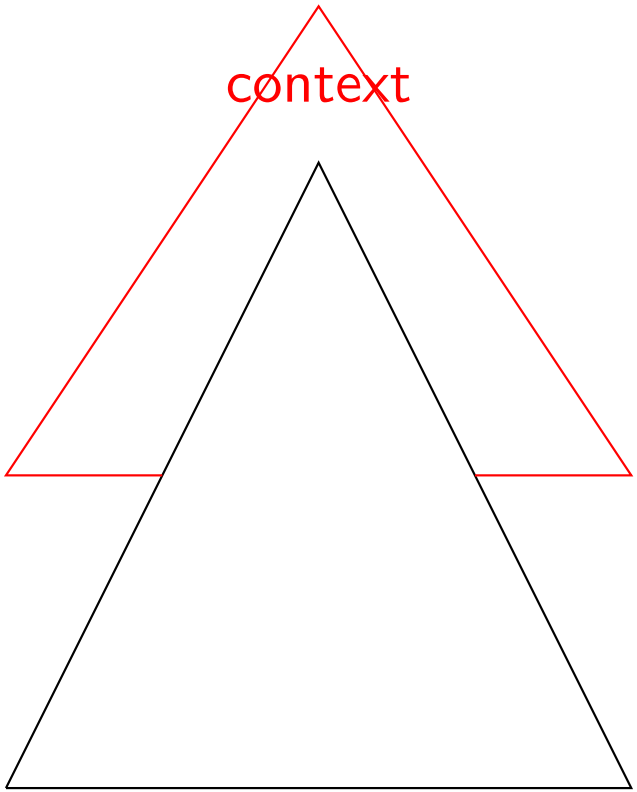
Output

```
Goal added: mirror(mirror(L:TList)) = L:TList
Induction started with the variables:
L:TList
Proof succeeded.
Number of derived goals: 4
Number of proving steps performed: 19
Maximum number of proving steps is set to: 256
Proved properties:
mirror(mirror(L#3:Tree)) = L#3:Tree
mirror(mirror(L:TList)) = L:TList
```

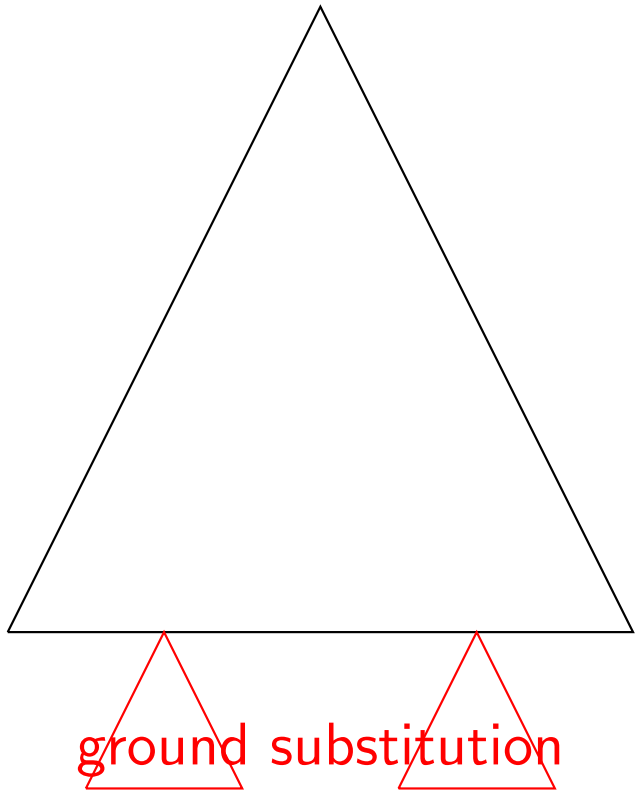
Duality Between Induction and Coinduction 1/2

Experiments

Coinduction



Induction

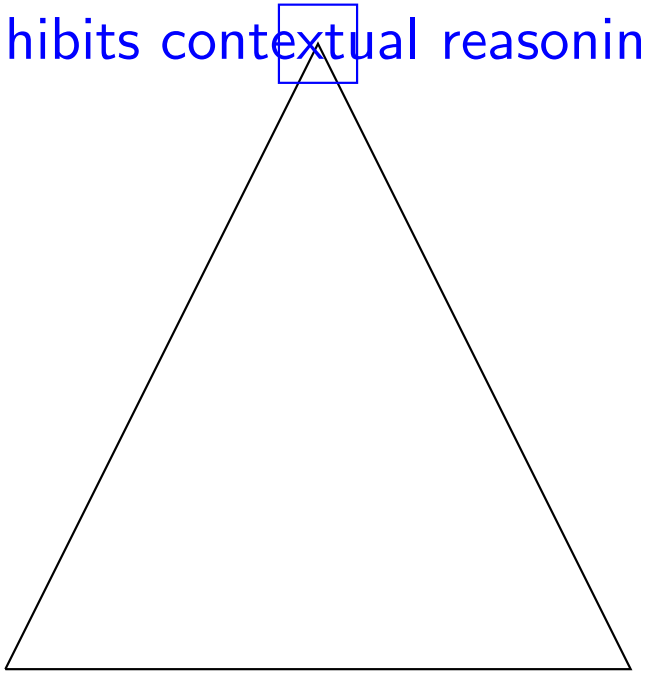


Duality Between Induction and Coinduction 2/2

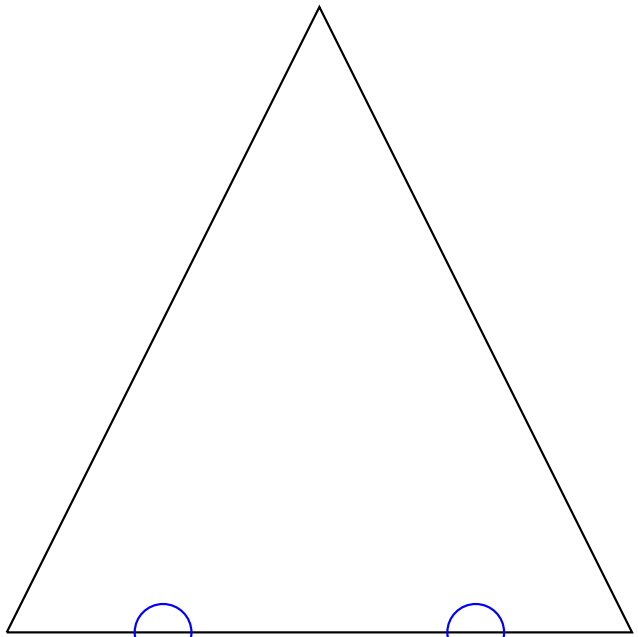
Freezing

Coinduction

inhibits contextual reasoning



Induction



inhibits subst. reasoning

Plan

Conclusion

- we defined abstract behavioral specifications and abstract circular reasoning proof system
- (equational) coinductive behavioral specifications and inductive behavioral specifications are becoming instances
  - inductive properties are related to the initial model
  - coinductive properties are related to the final model (if any)
- we showed that the circular reasoning proof system can be lifted up to the abstract behavioral specifications
- circular coinduction and circular induction proof systems are becoming instances
- both systems are implemented in Circ tool (thanks to Georgiana Caltais and Eugen Goriac)
- the difference between coinductive properties and the inductive ones is given by how the experiments are defined

Thanks!

Questions?