Complete Sequent Calculi for Induction and Infinite Descent

James Brotherston

Programming Principles, Logic and Verification Group
Dept. of Computer Science
University College London, UK
J.Brotherston@ucl.ac.uk

Leeds Logic Seminar, 17 February 2016

Introduction

- We investigate and compare two related styles of inductive reasoning:

Introduction

- We investigate and compare two related styles of inductive reasoning:
    1. explicit rule induction over definitions;

Introduction

- We investigate and compare two related styles of inductive reasoning:
  1. explicit rule induction over definitions;
  2. infinite descent *à la* Fermat.

Introduction

- We investigate and compare two related styles of inductive reasoning:
    1. explicit rule induction over definitions;
    2. infinite descent *à la* Fermat.
- We work in first-order logic with inductive definitions.

Introduction

- We investigate and compare two related styles of inductive reasoning:
  1. explicit rule induction over definitions;
  2. infinite descent *à la* Fermat.
- We work in first-order logic with inductive definitions.
- We formulate and compare proof-theoretic foundations of thes two styles of reasoning above, using Gentzen-style sequent calculus proof systems.

Part I

Inductive definitions in first-order logic

First-order logic with inductive definitions (FOL_{ID})

- We extend standard first-order logic with a schema for inductive definitions.

First-order logic with inductive definitions (FOL_{ID})

- We extend standard first-order logic with a schema for inductive definitions.
- Our inductive rules are each of the form:

$$P_1(\mathbf{t_1}(\mathbf{x})) \ldots P_m(\mathbf{t_m}(\mathbf{x})) \Rightarrow P(\mathbf{t}(\mathbf{x}))$$

where $P, P_1, \ldots, P_m$ are predicate symbols.

First-order logic with inductive definitions (FOL_{ID})

- We extend standard first-order logic with a schema for inductive definitions.

- Our inductive rules are each of the form:

$$P_1(\mathbf{t_1}(\mathbf{x})) \ldots P_m(\mathbf{t_m}(\mathbf{x})) \Rightarrow P(\mathbf{t}(\mathbf{x}))$$

  where $P, P_1, \ldots, P_m$ are predicate symbols.

- E.g., define N, E, O, R^+ (natural nos; even/odd nos; transitive closure of R) by rules

$$\begin{array}{rcl} & \Rightarrow & N0 \\ Nx & \Rightarrow & Nsx \end{array} \qquad \begin{array}{rcl} & \Rightarrow & E0 \\ Ox & \Rightarrow & Esx \\ Ex & \Rightarrow & Osx \end{array} \qquad \begin{array}{rcl} Rxy & \Rightarrow & R^+xy \\ R^+xy, R^+yz & \Rightarrow & R^+xz \end{array}$$

Standard models of FOL_{ID}

- The inductive rules determine a monotone operator φ_Φ on any first-order structure M.

Standard models of FOL_{ID}

- The inductive rules determine a monotone operator φ_Φ on any first-order structure M. E.g., for N:

$$\varphi_{\Phi_N}(X) = \{0^M\} \cup \{s^M x \mid x \in X\}$$

- In standard models, P^M is the least prefixed point of the corresponding operator.

Standard models of FOL_{ID}

- The inductive rules determine a monotone operator φ_Φ on any first-order structure M. E.g., for N:

$$\varphi_{\Phi_N}(X) = \{0^M\} \cup \{s^M x \mid x \in X\}$$

- In standard models, P^M is the least prefixed point of the corresponding operator.
- This least prefixed point can be approached via a sequence (φ^α_Φ) of approximants.

Standard models of FOL_{ID}

- The inductive rules determine a monotone operator φ_Φ on any first-order structure M. E.g., for N:

$$\varphi_{\Phi_N}(X) = \{0^M\} \cup \{s^M x \mid x \in X\}$$

- In standard models, P^M is the least prefixed point of the corresponding operator.

- This least prefixed point can be approached via a sequence (φ^α_Φ) of approximants. E.g. for N we have:

$$\varphi^0_{\Phi_N} = \emptyset,\ \varphi^1_{\Phi_N} = \{0^M\},\ \varphi^2_{\Phi_N} = \{0^M, s^M 0^M\}, \ldots$$

Henkin models of FOL_{ID}

- We can also give non-standard interpretations to the inductive predicates of the language, in so-called Henkin models.

Henkin models of FOL_{ID}

- We can also give non-standard interpretations to the inductive predicates of the language, in so-called Henkin models.
- A class of sets $\mathcal{H}$ over a first order structure M is a Henkin class if, roughly speaking, every first-order-definable relation is interpretable inside it.

Henkin models of FOL_{ID}

- We can also give non-standard interpretations to the inductive predicates of the language, in so-called Henkin models.
- A class of sets $\mathcal{H}$ over a first order structure M is a Henkin class if, roughly speaking, every first-order-definable relation is interpretable inside it.
- $(M, \mathcal{H})$ is a Henkin model if the least prefixed point of φ_Φ exists inside $\mathcal{H}$; we define P^M to be this point.

Part II

Sequent calculus for explicit induction

LKID: a sequent calculus for induction in FOL_{ID}

Extend the usual sequent calculus LK_e for classical first-order logic with equality by adding rules for inductive predicates.

LKID: a sequent calculus for induction in FOL_{ID}

Extend the usual sequent calculus LK_e for classical first-order logic with equality by adding rules for inductive predicates.

E.g., right-introduction rules for N are:

$$\frac{}{\Gamma \vdash N0, \Delta}\,(NR_1) \qquad \frac{\Gamma \vdash Nt, \Delta}{\Gamma \vdash Nst, \Delta}\,(NR_2)$$

LKID: a sequent calculus for induction in FOL_{ID}

Extend the usual sequent calculus LK_e for classical first-order logic with equality by adding rules for inductive predicates. E.g., right-introduction rules for N are:

$$\frac{}{\Gamma \vdash N0, \Delta}\,(NR_1) \qquad \frac{\Gamma \vdash Nt, \Delta}{\Gamma \vdash Nst, \Delta}\,(NR_2)$$

The left-introduction rule embodies rule induction:

$$\frac{\Gamma \vdash F0, \Delta \qquad \Gamma, Fx \vdash Fsx, \Delta \qquad \Gamma, Ft \vdash \Delta}{\Gamma, Nt \vdash \Delta}\,(x \text{ fresh})\ (\mathrm{Ind}\ N)$$

LKID: a sequent calculus for induction in FOL_{ID}

Extend the usual sequent calculus LK_e for classical first-order logic with equality by adding rules for inductive predicates. E.g., right-introduction rules for N are:

$$\frac{}{\Gamma \vdash N0, \Delta}\,(NR_1) \qquad \frac{\Gamma \vdash Nt, \Delta}{\Gamma \vdash Nst, \Delta}\,(NR_2)$$

The left-introduction rule embodies rule induction:

$$\frac{\Gamma \vdash F0, \Delta \qquad \Gamma, Fx \vdash Fsx, \Delta \qquad \Gamma, Ft \vdash \Delta}{\Gamma, Nt \vdash \Delta}\,(x \text{ fresh})\ (\text{Ind } N)$$

NB. Mutual definitions give rise to mutual induction rules.

Results about LKID

Proposition (Soundness)

Any LKID-provable sequent is valid in all Henkin models.

Results about LKID

Proposition (Soundness)

Any LKID-provable sequent is valid in all Henkin models.

Theorem (Completeness)

Any sequent valid in all Henkin models is cut-free provable in LKID.

Results about LKID

Proposition (Soundness)

Any LKID-provable sequent is valid in all Henkin models.

Theorem (Completeness)

Any sequent valid in all Henkin models is cut-free provable in LKID.

- Supposing $\Gamma \vdash \Delta$ not provable, we use a uniform infinitary search procedure to build an unprovable limit sequent $\Gamma_\omega \vdash \Delta_\omega$.

Results about LKID

Proposition (Soundness)

Any LKID-provable sequent is valid in all Henkin models.

Theorem (Completeness)

Any sequent valid in all Henkin models is cut-free provable in LKID.

- Supposing $\Gamma \vdash \Delta$ not provable, we use a uniform infinitary search procedure to build an unprovable limit sequent $\Gamma_\omega \vdash \Delta_\omega$.
- We then use this limit sequent to define a syntactic countermodel for $\Gamma \vdash \Delta$.

Results about LKID

Proposition (Soundness)

Any LKID-provable sequent is valid in all Henkin models.

Theorem (Completeness)

Any sequent valid in all Henkin models is cut-free provable in LKID.

- Supposing $\Gamma \vdash \Delta$ not provable, we use a uniform infinitary search procedure to build an unprovable limit sequent $\Gamma_\omega \vdash \Delta_\omega$.
- We then use this limit sequent to define a syntactic countermodel for $\Gamma \vdash \Delta$.
- (We need to define a Henkin class and deal with inductive predicates though.)

Cut-elimination in LKID

Corollary

Any LKID-provable sequent is provable without cut.

Cut-elimination in LKID

Corollary

Any LKID-provable sequent is provable without cut.

This is contrary to the popular myth that cut-elimination is impossible in the presence of induction.

Cut-elimination in LKID

Corollary

Any LKID-provable sequent is provable without cut.

This is contrary to the popular myth that cut-elimination is impossible in the presence of induction. In fact, the real limitation is that the subformula property is not achievable.

Cut-elimination in LKID

Corollary

Any LKID-provable sequent is provable without cut.

This is contrary to the popular myth that cut-elimination is impossible in the presence of induction. In fact, the real limitation is that the subformula property is not achievable.

Proposition

The eliminability of cut in LKID implies the consistency of Peano arithmetic.

Cut-elimination in LKID

Corollary

Any LKID-provable sequent is provable without cut.

This is contrary to the popular myth that cut-elimination is impossible in the presence of induction. In fact, the real limitation is that the subformula property is not achievable.

Proposition

The eliminability of cut in LKID implies the consistency of Peano arithmetic.

Hence there is no elementary proof of cut-eliminability in LKID.

Part III

Sequent calculus for infinite descent

$LKID^{\omega}$: a proof system for infinite descent in FOL_{ID}

- Rules are as for LKID except the induction rules are replaced by weaker case-split rules.

LKID$^\omega$: a proof system for infinite descent in FOL_{ID}

- Rules are as for LKID except the induction rules are replaced by weaker case-split rules. E.g. for N:

$$\frac{\Gamma, t = 0 \vdash \Delta \quad \Gamma, t = sx, Nx \vdash \Delta}{\Gamma, Nt \vdash \Delta} \ (x \text{ fresh}) \ (\text{Case } N)$$

LKID$^\omega$: a proof system for infinite descent in FOL$_{ID}$

- Rules are as for LKID except the induction rules are replaced by weaker case-split rules. E.g. for N:

$$\frac{\Gamma, t = 0 \vdash \Delta \quad \Gamma, t = sx, Nx \vdash \Delta}{\Gamma, Nt \vdash \Delta} \ (x \text{ fresh}) \ (\text{Case } N)$$

- Pre-proofs are infinite (non-well-founded) derivation trees.

LKID$^\omega$: a proof system for infinite descent in FOL$_{ID}$

- Rules are as for LKID except the induction rules are replaced by weaker case-split rules. E.g. for N:

$$\frac{\Gamma, t = 0 \vdash \Delta \quad \Gamma, t = sx, Nx \vdash \Delta}{\Gamma, Nt \vdash \Delta} \ (x \text{ fresh}) \ (\text{Case } N)$$

- Pre-proofs are infinite (non-well-founded) derivation trees.
- For soundness we need to impose an additional condition on pre-proofs.

Traces

- A trace following a path in an LKID^{ω} pre-proof tracks an inductive predicate occurring on the left of the sequents on the path.

Traces

- A trace following a path in an LKID$^\omega$ pre-proof tracks an inductive predicate occurring on the left of the sequents on the path.
- A trace progresses when the inductive predicate is unfolded using its case-split rule.

Traces

- A trace following a path in an LKID$^\omega$ pre-proof tracks an inductive predicate occurring on the left of the sequents on the path.
- A trace progresses when the inductive predicate is unfolded using its case-split rule.
- A pre-proof is a proof if, for every infinite path in it, there is an infinitely progressing trace following some tail of the path.

A sample proof

$$
\dfrac{
\dfrac{
\dfrac{}{\vdash E0, O0}\,(ER_1)
}{x_0 = 0 \vdash Ex_0, Ox_0}\,(=\mathrm{L})
\qquad
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\begin{array}{c}\text{(etc.)}\\ \vdots\end{array}
}{{\color{red}Nx_1} \vdash Ex_1, Ox_1}\,(\text{Case } N)
}{{\color{red}Nx_1} \vdash Ox_1, Osx_1}\,(OR_1)
}{{\color{red}Nx_1} \vdash Esx_1, Osx_1}\,(ER_2)
}{x_0 = sx_1, {\color{red}Nx_1} \vdash Ex_0, Ox_0}\,(=\mathrm{L})
}{{\color{red}Nx_0} \vdash Ex_0, Ox_0}\,(\text{Case } N)
$$

A sample proof

$$
\dfrac{\dfrac{\dfrac{}{\vdash E0, O0}\,(ER_1)}{x_0 = 0 \vdash Ex_0, Ox_0}\,(=\text{L}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}(\text{etc.})\\ \vdots\end{array}}{Nx_1 \vdash Ex_1, Ox_1}\,(\text{Case } N)}{Nx_1 \vdash Ox_1, Osx_1}\,(OR_1)}{Nx_1 \vdash Esx_1, Osx_1}\,(ER_2)}{x_0 = sx_1, Nx_1 \vdash Ex_0, Ox_0}\,(=\text{L})}{Nx_0 \vdash Ex_0, Ox_0}\,(\text{Case } N)
$$

Continuing the expansion of the right branch, the formulas in red form an infinitely progressing trace, so the pre-proof thus obtained is indeed an LKID^{ω} proof.

$LKID^{\omega}$: soundness

Proposition

Any $LKID^{\omega}$-provable sequent is valid in all standard models.

$LKID^{\omega}$: soundness

Proposition

Any $LKID^{\omega}$-provable sequent is valid in all standard models.

Roughly:

- Suppose $\Gamma \vdash \Delta$ is not valid. Since rules are locally sound, there must be an infinite path in the pre-proof consisting of invalid sequents.

LKID$^\omega$: soundness

Proposition

Any LKID$^\omega$-provable sequent is valid in all standard models.

Roughly:

- Suppose $\Gamma \vdash \Delta$ is not valid. Since rules are locally sound, there must be an infinite path in the pre-proof consisting of invalid sequents.

- By the soundness condition, there is an infinitely progressing trace of this path following some predicate P say.

$LKID^\omega$: soundness

Proposition

Any $LKID^\omega$-provable sequent is valid in all standard models.

Roughly:

- Suppose $\Gamma \vdash \Delta$ is not valid. Since rules are locally sound, there must be an infinite path in the pre-proof consisting of invalid sequents.
- By the soundness condition, there is an infinitely progressing trace of this path following some predicate P say.
- But then we can construct an infinite descending chain of ordinals based on the approximants of P, contradiction.

Completeness of $LKID^\omega$

Theorem

Any sequent valid in all standard models has a cut-free proof in $LKID^\omega$.

Completeness of $LKID^\omega$

Theorem

Any sequent valid in all standard models has a cut-free proof in $LKID^\omega$.

- Given $\Gamma \vdash \Delta$ (not provable), we construct an infinite derivation tree corresponding to an exhaustive search for a proof of it.

Completeness of $LKID^\omega$

Theorem

Any sequent valid in all standard models has a cut-free proof in $LKID^\omega$.

- Given $\Gamma \vdash \Delta$ (not provable), we construct an infinite derivation tree corresponding to an exhaustive search for a proof of it.
- Either the tree gets stuck at some node which we call $\Gamma_\omega \vdash \Delta_\omega$, or else some branch fails the trace condition, in which case $\Gamma_\omega \vdash \Delta_\omega$ is the "limit union" of the sequents along this branch.

Completeness of LKID$^\omega$

Theorem

Any sequent valid in all standard models has a cut-free proof in LKID$^\omega$.

- Given $\Gamma \vdash \Delta$ (not provable), we construct an infinite derivation tree corresponding to an exhaustive search for a proof of it.
- Either the tree gets stuck at some node which we call $\Gamma_\omega \vdash \Delta_\omega$, or else some branch fails the trace condition, in which case $\Gamma_\omega \vdash \Delta_\omega$ is the "limit union" of the sequents along this branch.
- Either way, we show $\Gamma_\omega \vdash \Delta_\omega$ is not provable (this uses the trace condition).

Completeness of $LKID^\omega$

Theorem

Any sequent valid in all standard models has a cut-free proof in $LKID^\omega$.

- Given $\Gamma \vdash \Delta$ (not provable), we construct an infinite derivation tree corresponding to an exhaustive search for a proof of it.
- Either the tree gets stuck at some node which we call $\Gamma_\omega \vdash \Delta_\omega$, or else some branch fails the trace condition, in which case $\Gamma_\omega \vdash \Delta_\omega$ is the "limit union" of the sequents along this branch.
- Either way, we show $\Gamma_\omega \vdash \Delta_\omega$ is not provable (this uses the trace condition).
- Thus we can use $\Gamma_\omega \vdash \Delta_\omega$ to construct a syntactic counter-model (the inductive predicate case also uses the trace condition).

Eliminability of cut

Corollary

Any $LKID^{\omega}$-provable sequent also has a cut-free $LKID^{\omega}$ proof.

Eliminability of cut

Corollary

Any $LKID^{\omega}$-provable sequent also has a cut-free $LKID^{\omega}$ proof.

Unlike in LKID, cut-free proofs in LKID^{ω} enjoy a property akin to the subformula property, which seems close to the spirit of Girard's "purity of methods".

Part IV

Cyclic proofs by infinite descent

CLKID$^\omega$: a cyclic subsystem of LKID$^\omega$

- The infinitary system LKID$^\omega$ is clearly unsuitable for formal reasoning!

CLKID$^\omega$: a cyclic subsystem of LKID$^\omega$

- The infinitary system LKID$^\omega$ is clearly unsuitable for formal reasoning!
- Indeed, completeness for standard validity implies that there is no complete enumeration of LKID$^\omega$ proofs.

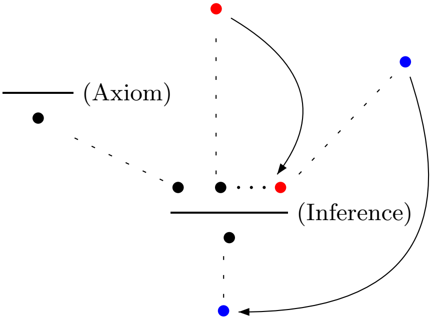
CLKID$^\omega$: a cyclic subsystem of LKID$^\omega$

- The infinitary system LKID$^\omega$ is clearly unsuitable for formal reasoning!
- Indeed, completeness for standard validity implies that there is no complete enumeration of LKID$^\omega$ proofs.
- However, the restriction of LKID$^\omega$ to proofs given by regular trees, which we call CLKID$^\omega$, is a natural one that *is* suitable for formal reasoning.

CLKID$^\omega$: a cyclic subsystem of LKID$^\omega$

- The infinitary system LKID$^\omega$ is clearly unsuitable for formal reasoning!
- Indeed, completeness for standard validity implies that there is no complete enumeration of LKID$^\omega$ proofs.
- However, the restriction of LKID$^\omega$ to proofs given by regular trees, which we call CLKID$^\omega$, is a natural one that *is* suitable for formal reasoning.
- In this restricted system, every proof can be represented as a finite (cyclic) graph.

Cyclic proofs



A cyclic proof

$$\dfrac{\dfrac{}{\vdash E0, O0}\,(ER_1) \qquad \dfrac{\dfrac{\dfrac{\dfrac{Nz \vdash Oz, Ez\ (\dagger)}{Ny \vdash Oy, Ey}\,(\text{Subst})}{Ny \vdash Oy, Osy}\,(OR_1)}{Ny \vdash Esy, Osy}\,(ER_2)}{Nz \vdash Ez, Oz\ (\dagger)}\,(NL)$$

A cyclic proof

$$
\dfrac{\dfrac{}{\vdash E0, O0}\,(ER_1) \qquad \dfrac{\dfrac{\dfrac{\dfrac{Nz \vdash Oz, Ez\ (\dagger)}{Ny \vdash Oy, Ey}\,(\text{Subst})}{Ny \vdash Oy, Osy}\,(OR_1)}{Ny \vdash Esy, Osy}\,(ER_2)}{Nz \vdash Ez, Oz\ (\dagger)}\,(NL)
$$

Any infinite path has a tail consisting of repetitions of the loop indicated by (†), and there is a progressing trace on this loop. By concatenating copies of this trace we obtain an infinitely progressing trace as required.

Results about CLKID$^\omega$

Proposition (Proof-checking decidability)

It is decidable whether a CLKID$^\omega$ pre-proof is a proof.

Results about $CLKID^{\omega}$

Proposition (Proof-checking decidability)

It is decidable whether a $CLKID^{\omega}$ pre-proof is a proof.

Theorem

Any LKID proof can be transformed into a $CLKID^{\omega}$ proof.

Results about CLKID$^\omega$

Proposition (Proof-checking decidability)

It is decidable whether a CLKID$^\omega$ pre-proof is a proof.

Theorem

Any LKID proof can be transformed into a CLKID$^\omega$ proof.

(Proof: We show how to derive any induction rule in CLKID$^\omega$.)

Results about CLKID$^\omega$

Proposition (Proof-checking decidability)

It is decidable whether a CLKID$^\omega$ pre-proof is a proof.

Theorem

Any LKID proof can be transformed into a CLKID$^\omega$ proof.

(Proof: We show how to derive any induction rule in CLKID$^\omega$.)

Conjecture

Any CLKID$^\omega$-provable sequent is also LKID-provable.

Results about CLKID$^\omega$

Proposition (Proof-checking decidability)

It is decidable whether a CLKID$^\omega$ pre-proof is a proof.

Theorem

Any LKID proof can be transformed into a CLKID$^\omega$ proof.

(Proof: We show how to derive any induction rule in CLKID$^\omega$.)

Conjecture

Any CLKID$^\omega$-provable sequent is also LKID-provable.

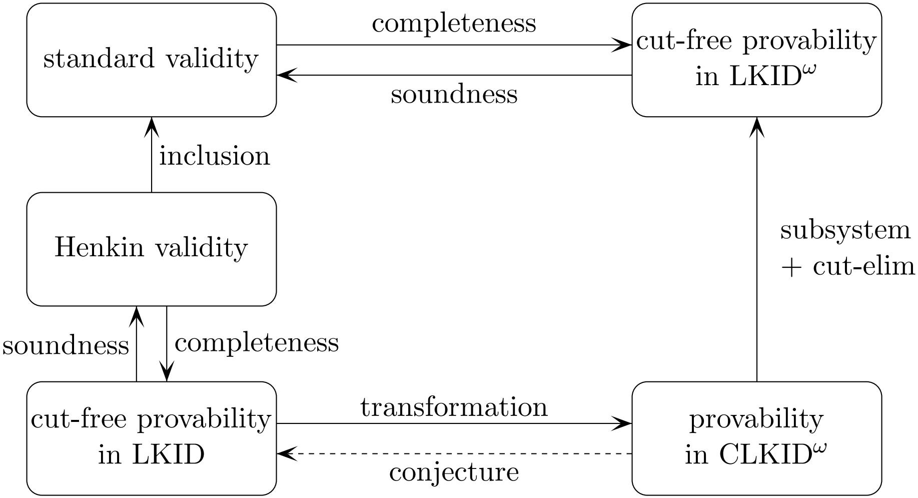
This conjecture can be seen as a formalised version of:

> *Proof by induction is equivalent to regular proof by infinite descent.*

Part V

Summary

Summary



standard validity
completeness
soundness
cut-free provability in LKID^ω
inclusion
Henkin validity
subsystem + cut-elim
soundness
completeness
cut-free provability in LKID
transformation
conjecture
provability in CLKID^ω

Some more recent developments

- Cyclic proof has started to see use in automatic theorem proving and in program verification tools.

Some more recent developments

- Cyclic proof has started to see use in automatic theorem proving and in program verification tools.
- Cyclic systems have been developed for various other logics with inductive definitions or fixed point operators.

Some more recent developments

- Cyclic proof has started to see use in automatic theorem proving and in program verification tools.
- Cyclic systems have been developed for various other logics with inductive definitions or fixed point operators.
- Attempts at solving the conjecture...

Further reading

P. Martin-Löf.
Haupstatz for the intuitionistic theory of iterated inductive definitions.
In *Proc. Second Scandinavian Logic Symposium*, 1971.

J. Brotherston and A. Simpson.
Sequent calculi for induction and infinite descent.
In *Journal of Logic and Computation* 21(6), 2011.

J. Brotherston, R. Bornat and C. Calcagno.
Cyclic proofs of program termination in separation logic.
In *Proc. POPL*, 2008.

J. Brotherston and N. Gorogiannis.
A generic cyclic theorem prover.
In *Proc. APLAS*, 2012.

C. Sprenger and M. Dam.
On the structure of inductive reasoning: circular and tree-shaped proofs in the μ-calculus.
In *Proceedings of FOSSACS*, 2003.