Certification of Nonclausal Connection Tableaux Proofs

Michael Färber Cezary Kaliszyk

September 3rd 2019

Introduction

leanCoP/nanoCoP

leanCoP/nanoCoP

- Provers for classical first-order logic
- Written by Jens Otten in Prolog
- Clausal (leanCoP) / nonclausal (nanoCoP) connection calculi
- Variants for intuitionistic and modal logic exist

My Work

- Functional implementation in OCaml
- Monte Carlo Proof Search (only for leanCoP)
- Proof reconstruction in HOL Light

History

- 2015: Jens Otten mentions nonclausal proof search at PiWo'15
- 2016:
  - First competitive nonclausal ATP nanoCoP implemented
  - Presented at IJCAR'16
- 2017:
  - Reimplementation of nanoCoP in OCaml
  - Reconstruction of nonclausal proofs in HOL Light

Connection Proof Search

Connection Method/Mating

Persons

- Peter B. Andrews (*mating*)
- Wolfgang Bibel (*connection method*)

Advantages

- Formula-orientedness: keep formula structure instead of destroying it
- Goal-orientedness: naturally treat conjecture differently from axioms
- Uniformity: cover different logics
- Global view: small prover state in contrast to “sea of clauses”

Connection Calculus

Formula representation

- Matrix (horizontal): conjunction of its clauses[a]
- Clause (vertical): disjunction of its elements

[a] Contrary to Otten, we use a refutational point of view.

How to make connections?

- Connecting between literals L_1, L_2: if L_1 can be unified with complement of L_2
- Connecting with a clause: by connecting with *all* clause elements
- Connecting with a matrix: by connecting with *some* matrix element
- Connecting with some element also connects with its parent elements

Goal

Have at least one connection in main matrix

Clausal Example

$$F' = \forall x.(Q \wedge P(a) \wedge (\neg P(x) \vee \neg P(s^2 x)) \wedge (\neg P(x) \vee P(sx) \vee \neg Q))$$

Clausal Example

$$F' = \forall x.(Q \wedge P(a) \wedge (\neg P(x) \vee \neg P(s^2 x)) \wedge (\neg P(x) \vee P(sx) \vee \neg Q))$$

$$\left[[Q] \begin{bmatrix} \neg P(x') \\ P(sx') \\ \neg Q \end{bmatrix} [P(a)] \begin{bmatrix} \neg P(\bar{x}) \\ \neg P(s^2 \bar{x}) \end{bmatrix} \right]$$

$$\sigma = \{\}$$

Clausal Example

$$F' = \forall x.(Q \wedge P(a) \wedge (\neg P(x) \vee \neg P(s^2 x)) \wedge (\neg P(x) \vee P(sx) \vee \neg Q))$$

$$\left[[Q] \begin{bmatrix} \neg P(x') \\ P(sx') \\ \neg Q \end{bmatrix} [P(a)] \begin{bmatrix} \neg P(\bar{x}) \\ \neg P(s^2 \bar{x}) \end{bmatrix} \right]$$

1

$$\sigma = \{\}$$

Clausal Example

$$F' = \forall x.(Q \wedge P(a) \wedge (\neg P(x) \vee \neg P(s^2 x)) \wedge (\neg P(x) \vee P(sx) \vee \neg Q))$$

(2)

$$\left[[Q] \begin{bmatrix} \neg P(x') \\ P(sx') \\ \neg Q \end{bmatrix} [P(a)] \begin{bmatrix} \neg P(\bar{x}) \\ \neg P(s^2 \bar{x}) \end{bmatrix} \right]$$

(1)

$$\sigma = \{x' \mapsto a\}$$

Clausal Example

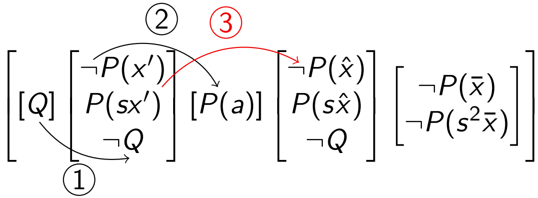
$$F' = \forall x.(Q \land P(a) \land (\neg P(x) \lor \neg P(s^2 x)) \land (\neg P(x) \lor P(sx) \lor \neg Q))$$

$$\left[[Q] \begin{bmatrix} \neg P(x') \\ P(sx') \\ \neg Q \end{bmatrix} [P(a)] \begin{bmatrix} \neg P(\hat{x}) \\ P(s\hat{x}) \\ \neg Q \end{bmatrix} \begin{bmatrix} \neg P(\bar{x}) \\ \neg P(s^2\bar{x}) \end{bmatrix} \right]$$

$\sigma = \{x' \mapsto a\}$

Clausal Example

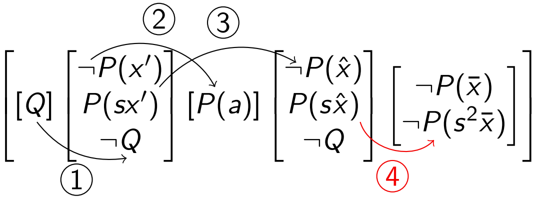
$$F' = \forall x.(Q \land P(a) \land (\neg P(x) \lor \neg P(s^2 x)) \land (\neg P(x) \lor P(sx) \lor \neg Q))$$



$$\sigma = \{x' \mapsto a, \hat{x} \mapsto sx'\}$$

Clausal Example

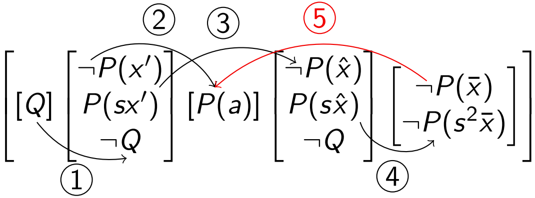
$$F' = \forall x.(Q \wedge P(a) \wedge (\neg P(x) \vee \neg P(s^2 x)) \wedge (\neg P(x) \vee P(sx) \vee \neg Q))$$



$$\sigma = \{x' \mapsto a, \hat{x} \mapsto sx', \bar{x} \mapsto x'\}$$

Clausal Example

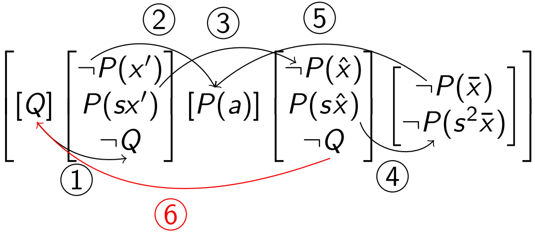
$$F' = \forall x. (Q \land P(a) \land (\neg P(x) \lor \neg P(s^2 x)) \land (\neg P(x) \lor P(sx) \lor \neg Q))$$



$$\sigma = \{x' \mapsto a, \hat{x} \mapsto sx', \bar{x} \mapsto x'\}$$

Clausal Example

$$F' = \forall x.(Q \land P(a) \land (\neg P(x) \lor \neg P(s^2 x)) \land (\neg P(x) \lor P(sx) \lor \neg Q))$$



$$\sigma = \{x' \mapsto a, \hat{x} \mapsto sx', \bar{x} \mapsto x'\}$$

Nonclausal Connection Calculus

Nonclausal Connection Calculus

Clauses can now not only contain literals, but also matrices

Complexity

- Nonclausal calculus can linearly simulate clausal calculus
- Clausal calculus can *not* polynomially simulate nonclausal calculus

Nonclausal Example

$$F = Q \land P(a) \land \forall x.(\neg P(x) \lor (\neg P(s^2 x) \land (P(sx) \lor \neg Q)))$$

Nonclausal Example

$$F = Q \wedge P(a) \wedge \forall x.(\neg P(x) \vee (\neg P(s^2 x) \wedge (P(sx) \vee \neg Q)))$$

$$\left[[Q][P(a)] \left[\begin{matrix} \neg P(x') \\ \left[[\neg P(s^2 x')] \begin{bmatrix} P(sx') \\ \neg Q \end{bmatrix} \right] \end{matrix} \right] \right]$$

$$\sigma = \{\}$$

Nonclausal Example

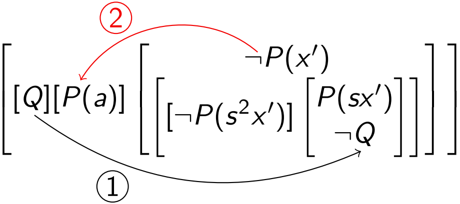
$$F = Q \wedge P(a) \wedge \forall x.(\neg P(x) \vee (\neg P(s^2 x) \wedge (P(sx) \vee \neg Q)))$$

$$\left[[Q][P(a)] \left[\begin{matrix} \neg P(x') \\ \left[[\neg P(s^2 x')] \begin{bmatrix} P(sx') \\ \neg Q \end{bmatrix} \right] \end{matrix} \right] \right]$$

$$\sigma = \{\}$$

Nonclausal Example

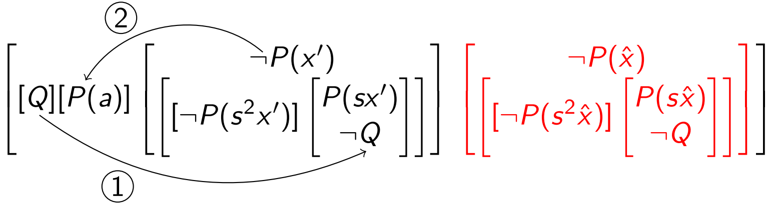
$$F = Q \wedge P(a) \wedge \forall x.(\neg P(x) \vee (\neg P(s^2 x) \wedge (P(sx) \vee \neg Q)))$$



$$\sigma = \{x' \mapsto a\}$$

Nonclausal Example

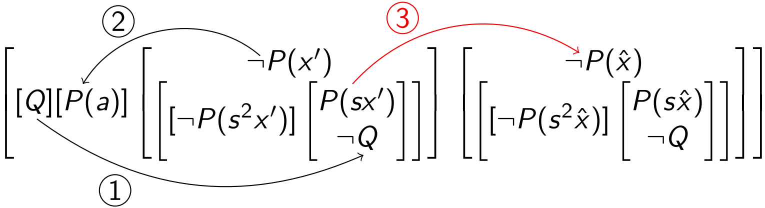
$$F = Q \land P(a) \land \forall x.(\neg P(x) \lor (\neg P(s^2x) \land (P(sx) \lor \neg Q)))$$



$$\sigma = \{x' \mapsto a\}$$

Nonclausal Example

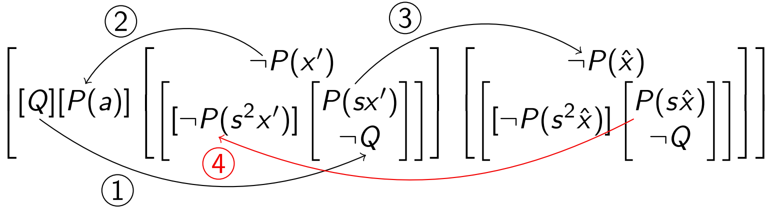
$$F = Q \wedge P(a) \wedge \forall x.(\neg P(x) \vee (\neg P(s^2 x) \wedge (P(sx) \vee \neg Q)))$$



$$\sigma = \{x' \mapsto a, \hat{x} \mapsto sx'\}$$

Nonclausal Example

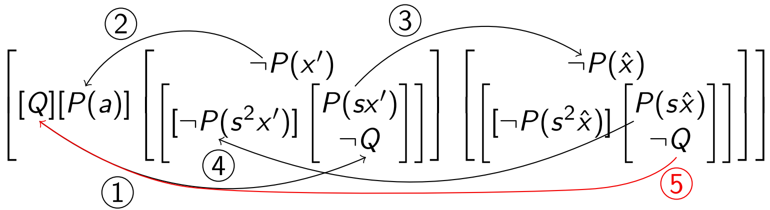
$$F = Q \land P(a) \land \forall x.(\neg P(x) \lor (\neg P(s^2 x) \land (P(sx) \lor \neg Q)))$$



$$\sigma = \{x' \mapsto a, \hat{x} \mapsto sx'\}$$

Nonclausal Example

$$F = Q \land P(a) \land \forall x.(\neg P(x) \lor (\neg P(s^2 x) \land (P(sx) \lor \neg Q)))$$



$$\sigma = \{x' \mapsto a, \hat{x} \mapsto sx'\}$$

Proof Certification

Preliminaries

Calculus

The words of the connection tableaux calculi have the shape $\langle L, M, \mathit{Path} \rangle$, where L is a literal, M is the matrix, and Path is a set of literals.

Clausal Certification

Given a connection proof of $\langle L, M, \mathit{Path} \rangle$, find an LK proof of $L, M, \mathit{Path} \vdash \bot$ in linear time.

How?

By recursion on the proof tree.

Clausal Certification Example

$$\begin{bmatrix} [Q] \begin{bmatrix} \neg P(x') \\ P(sx') \\ \neg Q \end{bmatrix} [P(a)] \begin{bmatrix} \neg P(\bar{x}) \\ \neg P(s^2 \bar{x}) \end{bmatrix} \end{bmatrix}$$

Clausal Certification Example

$$\left[[Q] \begin{bmatrix} \neg P(x') \\ P(sx') \\ \neg Q \end{bmatrix} [P(a)] \begin{bmatrix} \neg P(\bar{x}) \\ \neg P(s^2\bar{x}) \end{bmatrix} \right]$$

①

$$① \equiv \langle Q, M, \{\} \rangle$$

$$\frac{}{①}$$

Clausal Certification Example

$$\textcircled{2}$$

$$\left[[Q] \begin{bmatrix} \neg P(x') \\ P(sx') \\ \neg Q \end{bmatrix} [P(a)] \begin{bmatrix} \neg P(\bar{x}) \\ \neg P(s^2\bar{x}) \end{bmatrix} \right]$$

$$\textcircled{1}$$

$$\textcircled{1} \equiv \langle Q, M, \{\} \rangle$$

$$\textcircled{2} \equiv \langle \neg P(x'), M, \{Q\} \rangle$$

$$\cfrac{\cfrac{}{\textcircled{2}}}{\textcircled{1}}$$

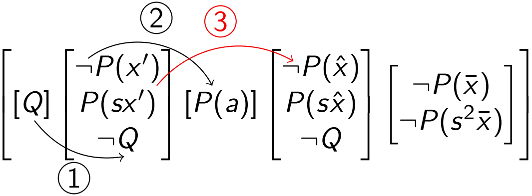
Clausal Certification Example

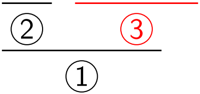
$$\left[[Q] \begin{bmatrix} \neg P(x') \\ P(sx') \\ \neg Q \end{bmatrix} [P(a)] \begin{bmatrix} \neg P(\hat{x}) \\ P(s\hat{x}) \\ \neg Q \end{bmatrix} \begin{bmatrix} \neg P(\bar{x}) \\ \neg P(s^2\bar{x}) \end{bmatrix} \right]$$

$$\textcircled{1} \equiv \langle Q, M, \{\} \rangle$$

$$\textcircled{2} \equiv \langle \neg P(x'), M, \{Q\} \rangle$$

$$\dfrac{\dfrac{}{\textcircled{2}}}{\textcircled{1}}$$

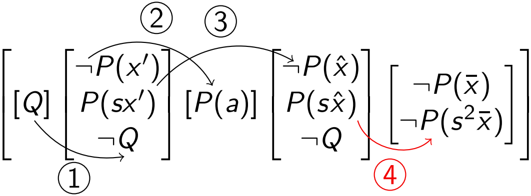


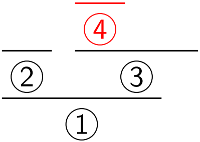


$$\textcircled{1} \equiv \langle Q, M, \{\} \rangle$$

$$\textcircled{2} \equiv \langle \neg P(x'), M, \{Q\} \rangle$$

$$\textcircled{3} \equiv \langle P(sx'), M, \{Q\} \rangle$$





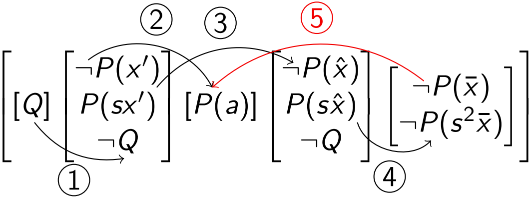
$(1) \equiv \langle Q, M, \{\}\rangle$

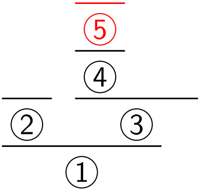
$(2) \equiv \langle \neg P(x'), M, \{Q\}\rangle$

$(3) \equiv \langle P(sx'), M, \{Q\}\rangle$

$(4) \equiv \langle P(s\hat{x}), M, \{Q, P(sx')\}\rangle$

Clausal Certification Example





$(1) \equiv \langle Q, M, \{\}\rangle$

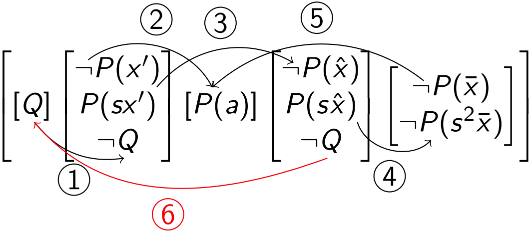
$(2) \equiv \langle \neg P(x'), M, \{Q\}\rangle$

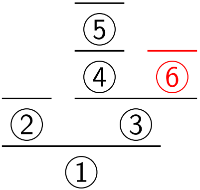
$(3) \equiv \langle P(sx'), M, \{Q\}\rangle$

$(4) \equiv \langle P(s\hat{x}), M, \{Q, P(sx')\}\rangle$

$(5) \equiv \langle \neg P(\bar{x}), M, \{Q, P(sx'), P(\hat{x})\}\rangle$

Clausal Certification Example





$$\text{①} \equiv \langle Q, M, \{\} \rangle$$

$$\text{②} \equiv \langle \neg P(x'), M, \{Q\} \rangle$$

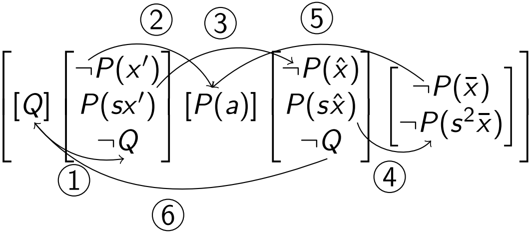
$$\text{③} \equiv \langle P(sx'), M, \{Q\} \rangle$$

$$\text{④} \equiv \langle P(s\hat{x}), M, \{Q, P(sx')\} \rangle$$

$$\text{⑤} \equiv \langle \neg P(\bar{x}), M, \{Q, P(sx'), P(\hat{x})\} \rangle$$

$$\text{⑥} \equiv \langle \neg Q, M, \{Q, P(sx')\} \rangle$$

Clausal Certification of ①



$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\boxed{\neg P(a), M, \{Q\} \vdash}} \quad \dfrac{\vdots}{\boxed{P(sa), M, \{Q\} \vdash}} \quad \dfrac{}{\neg Q, M, \{Q\} \vdash}\,\bot L}{(\neg P(a) \vee P(sa) \vee \neg Q), M, \{Q\} \vdash}\,\vee L}{\forall x.(\neg P(x) \vee P(sx) \vee \neg Q), M, \{Q\} \vdash}\,\forall L}{M, \{Q\} \vdash}\,\wedge L}{\boxed{Q, M, \{\} \vdash}}}{\boxed{M \vdash}}\,\wedge L$$

Nonclausal Certification Example

$$\left[[Q][P(a)] \left[\begin{matrix} \neg P(x') \\ \left[[\neg P(s^2 x')] \begin{bmatrix} P(sx') \\ \neg Q \end{bmatrix} \right] \end{matrix} \right] \right]$$

Nonclausal Certification Example

$$\left[[Q][P(a)] \left[\begin{matrix} \neg P(x') \\ \left[[\neg P(s^2 x')] \begin{bmatrix} P(sx') \\ \neg Q \end{bmatrix} \right] \end{matrix} \right] \right]$$

①

$$① \equiv \langle Q, M, \{\} \rangle$$

$$\frac{}{①}$$

Nonclausal Certification Example

$$\left[[Q][P(a)] \left[\begin{matrix} \neg P(x') \\ \left[[\neg P(s^2 x')] \begin{bmatrix} P(sx') \\ \neg Q \end{bmatrix} \right] \end{matrix} \right] \right]$$

$$\textcircled{1} \equiv \langle Q, M, \{\} \rangle$$

$$\textcircled{2} \equiv \langle \neg P(x'), M, \{Q\} \rangle$$

$$\dfrac{\dfrac{}{\textcircled{2}}}{\textcircled{1}}$$

Nonclausal Certification Example

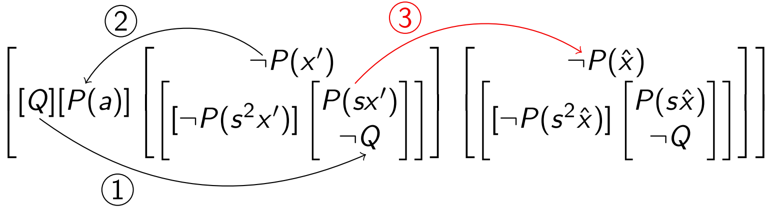
$$\left[\;[Q]\,[P(a)]\;\left[\begin{matrix}\neg P(x') \\ \left[\,[\neg P(s^2x')]\;\begin{bmatrix} P(sx') \\ \neg Q \end{bmatrix}\right]\end{matrix}\right]\;\color{red}{\left[\begin{matrix}\neg P(\hat{x}) \\ \left[\,[\neg P(s^2\hat{x})]\;\begin{bmatrix} P(s\hat{x}) \\ \neg Q \end{bmatrix}\right]\end{matrix}\right]}\;\right]$$

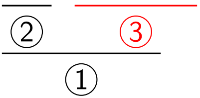
$$\textcircled{1} \equiv \langle Q, M, \{\}\rangle$$

$$\textcircled{2} \equiv \langle \neg P(x'), M, \{Q\}\rangle$$

$$\dfrac{\dfrac{}{\textcircled{2}}}{\textcircled{1}}$$

Nonclausal Certification Example



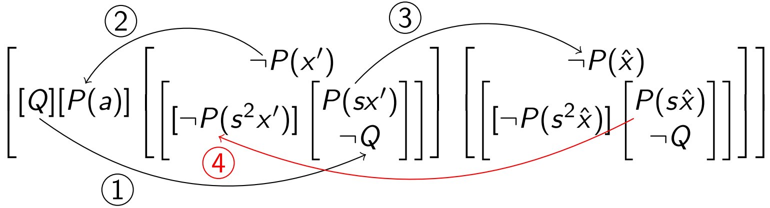


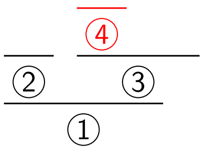
$(1) \equiv \langle Q, M, \{\}\rangle$

$(2) \equiv \langle \neg P(x'), M, \{Q\}\rangle$

$(3) \equiv \langle P(sx'), M, \{Q\}\rangle$

Nonclausal Certification Example





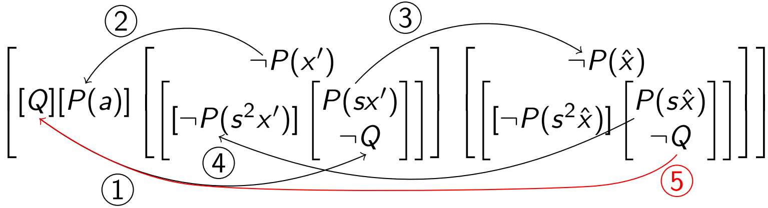
$① \equiv \langle Q, M, \{\} \rangle$

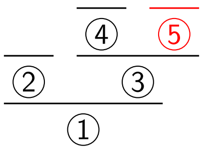
$② \equiv \langle \neg P(x'), M, \{Q\} \rangle$

$③ \equiv \langle P(sx'), M, \{Q\} \rangle$

$④ \equiv \langle P(s\hat{x}), M, \{Q, P(sx')\} \rangle$

Nonclausal Certification Example





$(1) \equiv \langle Q, M, \{\} \rangle$

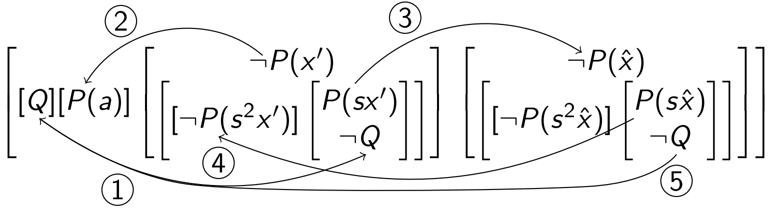
$(2) \equiv \langle \neg P(x'), M, \{Q\} \rangle$

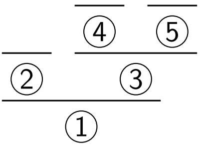
$(3) \equiv \langle P(sx'), M, \{Q\} \rangle$

$(4) \equiv \langle P(s\hat{x}), M, \{Q, P(sx')\} \rangle$

$(5) \equiv \langle \neg Q, M, \{Q, P(sx')\} \rangle$

Nonclausal Certification Example





$$①\equiv\langle Q, M, \{\}\rangle$$

$$②\equiv\langle \neg P(x'), M, \{Q\}\rangle$$

$$③\equiv\langle P(sx'), M, \{Q\}\rangle$$

$$④\equiv\langle P(s\hat{x}), M, \{Q, P(sx')\}\rangle$$

$$⑤\equiv\langle \neg Q, M, \{Q, P(sx')\}\rangle$$

Problem

The proof of ④ does not consider $\neg P(x')$, because it has already been closed by the proof of ②. However, given only the proof of ④, we do not know how to deal with $\neg P(x')$ in the translation.

Nonclausal Certification

Problem

Given a connection proof of $\langle L, M, Path \rangle$, it is not always possible to find an LK proof of $L, M, Path \vdash \bot$ in linear time.

Nonclausal Certification

Given a connection proof of $\langle L, M, Path \rangle$, find an LK proof of $L, M, Path, EC(M, Path) \vdash \bot$ in linear time, where $EC(M, Path)$ are the extension clauses of M with respect to $Path$.

Extension Clauses

- $EC(M, \{Q, P(sx')\} \ni [\neg P(s^2x')]$ (crucial for ④)
- $EC(M, \{\neg Q\}) = ?$

What is a literal, or: $P \neq P$

When are two literals equal?

Two literals in the nonclausal connection calculus are only equal if they are located at the same position in the matrix.

$EC(M, \{\neg Q\}) = ?$

- It depends on the position of $\neg Q$.
- Ignoring the position renders the calculus unsound.

Resulting problems from positional literals

- Performance bottleneck
- Difficult presentation & formalisation (?)
- Simple correspondence between calculus, implementation and proof certification lost (compared to clausal calculus)

Towards a different calculus

Observation

- The clausal calculus depends on *Path* for reduction steps only.
- The nonclausal calculus uses *Path* for reduction *and* extension steps, however, the encoding of positions in the literals is only crucial for the extension steps.

Suggestion

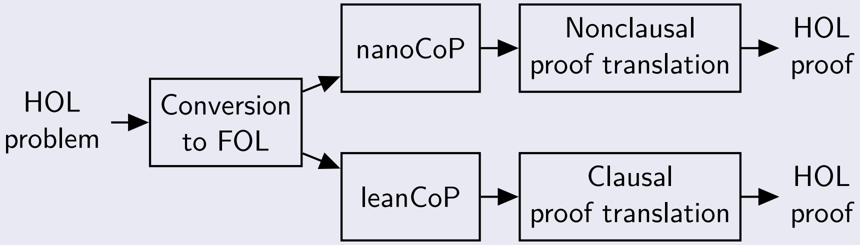
- Create a calculus that uses *Path* only for reduction steps, thus not requiring the encoding of positions in the literals, and use *something else* for the encoding of the extension steps.
- How to make this *something else* efficient?

Evaluation

Implementation

- OCaml implementations of leanCoP/nanoCoP
- Run as proof search tactics inside the interactive theorem prover HOL Light

Structure of the Proof Search Tactic



Evaluation inside HOL Light

Prover	HL-top	HL-msn	FS-top	FS-msn
Problems in dataset	2499	1119	27112	44468
Metis	807	1029	4626	42829
MESON	736	900	4221	39227
leanCoP+cut	724	948	3714	39922
leanCoP−cut	717	844	3800	38528
nanoCoP+cut	538	802	2743	34213
nanoCoP−cut	550	811	2351	34769

Performance problem

Connection provers compiled to native code perform around 30 times faster than inside OCaml's toplevel used by HOL Light due to faster array access

Evaluation outside HOL Light

Prover	TPTP	Bushy	Chainy	Miz40	FS-top	FS-msn
Vampire	4404	1253	656	30341	6358	39760
E	3664	1167	287	26003	7382	39740
Metis	1376	500	75	18519	3537	38625
leanCoP+cut+conj	1859	670	289	12204	3980	35738
leanCoP+cut−conj	1782	598	244	11796	3520	30668
leanCoP−cut+conj	1617	499	192	7826	3849	35204
leanCoP−cut−conj	1534	514	164	11115	3492	36334
nanoCoP+cut	1724	511	192	12332	3178	30327
nanoCoP−cut	1567	542	151	13316	1993	37938

Example Problem

```
!P. (!f s. P s /\ linear f ==> P (IMAGE f s))
    ==> (!f. linear f /\ (!x y. f x = f y ==> x = y)
             ==> (!s. P (IMAGE f s) <=> P s))
==>
!P f s. (!g t. P t /\ linear g ==> P (IMAGE g t)) /\
        linear f /\ (!x y. f x = f y ==> x = y)
        ==> (P (IMAGE f s) <=> P s)
```

- Taken from Flyspeck
- Given 10 seconds, only solved by nanoCoP (in 2.27 seconds)
- Deep structure, involving logical equivalences

Future Work & Conclusion

Future Work

Nonclassical proof reconstruction

- Current approach shows unsatisfiability of negated formula
- Derived from existing HOL Light tactics
- However, this will not work for nonclassical logics
- (Also, the proofs are not that "natural")

Formalisation

- Gain better understanding of connection calculi
- Anomaly in implementation:
  - Certain constellations can lead clause elements to remain unconnected
  - Only discovered by second reimplementation for proof reconstruction
- How to concisely formalise proof search for different logics?

Conclusion

- Clausal connection proofs can be easily certified.
- Nonclausal proofs are more involved, due to subproofs not being independently certifiable and positional literals.
- An improved nonclausal calculus might solve this and perhaps also some other problems (performance, presentation).