Interaction and Depth against Nondeterminism in Proof Search

Ozan Kahramanoğulları
Imperial College London
ozank@doc.ic.ac.uk

University of Birmingham, School of Computer Science
15 December 2006

The Calculus of Structures

- generalizes the sequent calculus with deep inference. [Guglielmi, 99]

The Calculus of Structures

- generalizes the sequent calculus with deep inference. [Guglielmi, 99]
- Inference rules can be applied at any depth inside a formula.

The Calculus of Structures

- generalizes the sequent calculus with deep inference. [Guglielmi, 99]
- Inference rules can be applied at any depth inside a formula.

A proof in the sequent system GS1p

$$\dfrac{\dfrac{}{\vdash a\,,\,\bar{a}}\,\mathrm{Ax} \qquad \dfrac{\dfrac{}{\vdash b\,,\,\bar{b}}\,\mathrm{Ax}}{\vdash b \vee \bar{b}}\,\mathrm{R}\vee}{\dfrac{\vdash a\,,\,\bar{a} \wedge (b \vee \bar{b})}{\vdash a \vee (\bar{a} \wedge (b \vee \bar{b}))}\,\mathrm{R}\vee}\,\mathrm{R}\wedge$$

The Calculus of Structures

- generalizes the sequent calculus with deep inference. [Guglielmi, 99]
- Inference rules can be applied at any depth inside a formula.

A proof in the sequent system GS1p

$$\dfrac{\dfrac{}{\vdash a\,,\,\bar{a}}\,\mathsf{Ax} \qquad \dfrac{\dfrac{}{\vdash b\,,\,\bar{b}}\,\mathsf{Ax}}{\vdash b \vee \bar{b}}\,\mathsf{R}\vee}{\dfrac{\vdash a\,,\,\bar{a} \wedge (b \vee \bar{b})}{\vdash a \vee (\bar{a} \wedge (b \vee \bar{b}))}\,\mathsf{R}\vee}\,\mathsf{R}\wedge$$

$\leadsto$

A proof in system KSg

$$\mathsf{ai}\!\downarrow \dfrac{\mathsf{ai}\!\downarrow \dfrac{\mathsf{tt}\!\downarrow \dfrac{}{\vdash \mathsf{tt}}}{\vdash a \vee \bar{a}}}{\vdash a \vee (\bar{a} \wedge (b \vee \bar{b}))}$$

The Calculus of Structures

- generalizes the sequent calculus with deep inference. [Guglielmi, 99]
- Inference rules can be applied at any depth inside a formula.

A proof in the sequent system GS1p

$$
\dfrac{\dfrac{\dfrac{}{\vdash a , \bar{a}}\,\mathsf{Ax} \qquad \dfrac{\dfrac{}{\vdash b , \bar{b}}\,\mathsf{Ax}}{\vdash b \vee \bar{b}}\,\mathsf{R}\vee}{\vdash a , \bar{a} \wedge (b \vee \bar{b})}\,\mathsf{R}\wedge}{\vdash a \vee (\bar{a} \wedge (b \vee \bar{b}))}\,\mathsf{R}\vee
$$

$\rightsquigarrow$

A proof in system KSg

$$
\mathsf{ai}\!\downarrow \dfrac{\mathsf{ai}\!\downarrow \dfrac{\mathsf{tt}\!\downarrow \dfrac{}{\vdash \mathsf{tt}}}{\vdash a \vee \bar{a}}}{\vdash a \vee (\bar{a} \wedge (b \vee \bar{b}))}
$$

Deep inference brings shorter proofs.

[Polynomial Size Deep-Inference Proofs Instead of Exponential Size Shallow-Inference Proofs, Guglielmi, 2004]

Deep Inference and Resolution

Cut-free sequent calculus does not polynomially simulate popular proof procedures such as resolution, e.g., [Beame, Pitassi,98].

Deep Inference and Resolution

Cut-free sequent calculus does not polynomially simulate popular proof procedures such as resolution, e.g., [Beame, Pitassi,98].

The resolution rule

$$\frac{R \wedge T}{(R \wedge a) \vee (T \wedge \bar{a})}$$

is derivable in the calculus of structures system for classical logic.

Deep Inference and Resolution

Cut-free sequent calculus does not polynomially simulate popular proof procedures such as resolution, e.g., [Beame, Pitassi,98].

The resolution rule

$$\mathsf{s}\,\dfrac{\mathsf{s}\,\dfrac{\mathsf{ai}{\downarrow}\,\dfrac{R \wedge T}{R \wedge T \wedge (a \vee \bar{a})}}{R \wedge (a \vee (T \wedge \bar{a}))}}{(R \wedge a) \vee (T \wedge \bar{a})}$$

is derivable in the calculus of structures system for classical logic.

Deep Inference and Resolution

Cut-free sequent calculus does not polynomially simulate popular proof procedures such as resolution, e.g., [Beame, Pitassi,98].

The resolution rule

$$\mathsf{s}\,\dfrac{\mathsf{s}\,\dfrac{\mathsf{ai}{\downarrow}\,\dfrac{R \wedge T}{R \wedge T \wedge (a \vee \bar{a})}}{R \wedge (a \vee (T \wedge \bar{a}))}}{(R \wedge a) \vee (T \wedge \bar{a})}$$

is derivable in the calculus of structures system for classical logic.

$$\mathsf{s}\,\frac{S((R \vee U) \wedge T)}{S((R \wedge T) \vee U)} \qquad \mathsf{ai}{\downarrow}\,\frac{S\{\mathtt{tt}\}}{S(a \vee \bar{a})}$$

However...

Consider the instance of the sequent calculus inference rule:

$$\frac{\vdash a, \bar{a} \quad \vdash b, \bar{b}}{\vdash a, b, \bar{a} \wedge \bar{b}} \, \mathsf{R}\wedge$$

However...

Consider the instance of the sequent calculus inference rule:

$$\frac{\vdash a, \bar{a} \quad \vdash b, \bar{b}}{\vdash a, b, \bar{a} \wedge \bar{b}} \mathsf{R}\wedge \qquad \vdash a, b, \bar{a} \wedge \bar{b} \qquad \rightsquigarrow \qquad a \vee b \vee (\bar{a} \wedge \bar{b})$$

However...

Consider the instance of the sequent calculus inference rule:

$$\frac{\vdash a, \bar{a} \quad \vdash b, \bar{b}}{\vdash a, b, \bar{a} \wedge \bar{b}} \mathsf{R}\wedge \qquad \vdash a, b, \bar{a} \wedge \bar{b} \qquad \rightsquigarrow \qquad a \vee b \vee (\bar{a} \wedge \bar{b})$$

In the calculus of structures this rule is simulated by the switch rule:

$$\mathsf{s}\,\frac{\mathsf{s}\,\dfrac{(a \vee \bar{a}) \wedge (b \vee \bar{b})}{a \vee (\bar{a} \wedge (b \vee \bar{b}))}}{a \vee b \vee (\bar{a} \wedge \bar{b})}$$

However...

Consider the instance of the sequent calculus inference rule:

$$\dfrac{\vdash a, \bar{a} \quad \vdash b, \bar{b}}{\vdash a, b, \bar{a} \wedge \bar{b}}\,\mathsf{R}\wedge \qquad \vdash a, b, \bar{a} \wedge \bar{b} \qquad \leadsto \qquad a \vee b \vee (\bar{a} \wedge \bar{b})$$

In the calculus of structures this rule is simulated by the switch rule:

$$\mathsf{s}\,\dfrac{\mathsf{s}\,\dfrac{(a \vee \bar{a}) \wedge (b \vee \bar{b})}{a \vee (\bar{a} \wedge (b \vee \bar{b}))}}{a \vee b \vee (\bar{a} \wedge \bar{b})}$$

Switch rule can be applied to $a \vee b \vee (\bar{a} \wedge \bar{b})$ in 27 different ways, and

However...

Consider the instance of the sequent calculus inference rule:

$$\dfrac{\vdash a, \bar{a} \quad \vdash b, \bar{b}}{\vdash a, b, \bar{a} \wedge \bar{b}}\,\mathsf{R}\wedge \qquad \vdash a, b, \bar{a} \wedge \bar{b} \qquad \leadsto \qquad a \vee b \vee (\bar{a} \wedge \bar{b})$$

In the calculus of structures this rule is simulated by the switch rule:

$$\mathsf{s}\,\dfrac{\mathsf{s}\,\dfrac{(a \vee \bar{a}) \wedge (b \vee \bar{b})}{a \vee (\bar{a} \wedge (b \vee \bar{b}))}}{a \vee b \vee (\bar{a} \wedge \bar{b})}$$

Switch rule can be applied to $a \vee b \vee (\bar{a} \wedge \bar{b})$ in 27 different ways, and

to $a_1 \vee b_1 \vee (\bar{a}_1 \wedge \bar{b}_1 \wedge (a_2 \vee b_2 \vee (\bar{a}_2 \wedge \bar{b}_2)))$ in 69 different ways.

However...

Consider the instance of the sequent calculus inference rule:

$$\dfrac{\vdash a, \bar{a} \quad \vdash b, \bar{b}}{\vdash a, b, \bar{a} \wedge \bar{b}}\,\mathsf{R}\wedge \qquad \vdash a, b, \bar{a} \wedge \bar{b} \qquad \rightsquigarrow \qquad a \vee b \vee (\bar{a} \wedge \bar{b})$$

In the calculus of structures this rule is simulated by the switch rule:

$$\mathsf{s}\,\dfrac{\mathsf{s}\,\dfrac{(a \vee \bar{a}) \wedge (b \vee \bar{b})}{a \vee (\bar{a} \wedge (b \vee \bar{b}))}}{a \vee b \vee (\bar{a} \wedge \bar{b})}$$

Switch rule can be applied to $a \vee b \vee (\bar{a} \wedge \bar{b})$ in 27 different ways, and

to $a_1 \vee b_1 \vee (\bar{a}_1 \wedge \bar{b}_1 \wedge (a_2 \vee b_2 \vee (\bar{a}_2 \wedge \bar{b}_2)))$ in 69 different ways.

Deep inference causes redundant nondeterminism.

System BV

- System BV: [Guglielmi,99] smallest technically nontrivial system

  MLL $+$ mix $+$ mix0 $+$ a non-commutative self-dual operator

  resembling prefix operator of process algebra: $a.b.P$

System BV

- System BV: [Guglielmi,99] smallest technically nontrivial system

  MLL + mix + mix0 + a non-commutative self-dual operator

  resembling prefix operator of process algebra: $a.b.P$

- BV structures:

  $$S ::= \circ \mid a \mid [\underbrace{S, \ldots, S}_{>0}] \mid (\underbrace{S, \ldots, S}_{>0}) \mid \langle \underbrace{S; \ldots ; S}_{>0} \rangle \mid \bar{S}$$

System BV

- System BV: [Guglielmi,99] smallest technically nontrivial system

  MLL + mix + mix0 + a non-commutative self-dual operator

  resembling prefix operator of process algebra: $a.b.P$

- BV structures:

  $$S ::= \circ \mid a \mid [\underbrace{S, \ldots, S}_{>0}] \mid (\underbrace{S, \ldots, S}_{>0}) \mid \langle \underbrace{S; \ldots; S}_{>0} \rangle \mid \bar{S}$$

  $[(\bar{a}, \bar{b}), a, b]$ corresponds to $((a^{\perp} \otimes b^{\perp}) \parr a \parr b)$

System BV

- System BV: [Guglielmi,99] smallest technically nontrivial system

  MLL + mix + mix0 + a non-commutative self-dual operator

  resembling prefix operator of process algebra: $a.b.P$

- BV structures:

  $$S ::= \circ \mid a \mid [\underbrace{S, \dots, S}_{>0}] \mid (\underbrace{S, \dots, S}_{>0}) \mid \langle \underbrace{S; \dots; S}_{>0} \rangle \mid \bar{S}$$

  $[(\bar{a}, \bar{b}), a, b]$ corresponds to $((a^{\perp} \otimes b^{\perp}) \parr a \parr b)$

- Structures are considered equivalent modulo an equational theory.

Syntactic Equivalence of BV Structures

Associativity

$$[R, [T, U]] = [[R, T], U]$$
$$(R, (T, U)) = ((R, T), U)$$
$$\langle R; \langle T; U\rangle\rangle = \langle\langle R; T\rangle; U\rangle$$

Syntactic Equivalence of BV Structures

Associativity

$$[R,[T,U]] = [[R,T],U]$$
$$(R,(T,U)) = ((R,T),U)$$
$$\langle R;\langle T;U\rangle\rangle = \langle\langle R;T\rangle;U\rangle$$

Commutativity

$$[R,T] = [T,R]$$
$$(R,T) = (T,R)$$

Syntactic Equivalence of BV Structures

Associativity

$$[R, [T, U]] = [[R, T], U]$$
$$(R, (T, U)) = ((R, T), U)$$
$$\langle R; \langle T; U \rangle\rangle = \langle\langle R; T \rangle; U\rangle$$

Commutativity

$$[R, T] = [T, R]$$
$$(R, T) = (T, R)$$

Unit

$$[\circ, R] = R$$
$$(\circ, R) = R$$
$$\langle R ; \circ \rangle = R$$
$$\langle \circ ; R \rangle = R$$

Syntactic Equivalence of BV Structures

Associativity

$$[R, [T, U]] = [[R, T], U]$$
$$(R, (T, U)) = ((R, T), U)$$
$$\langle R; \langle T; U \rangle \rangle = \langle \langle R; T \rangle; U \rangle$$

Commutativity

$$[R, T] = [T, R]$$
$$(R, T) = (T, R)$$

Unit

$$[\circ, R] = R$$
$$(\circ, R) = R$$
$$\langle R ; \circ \rangle = R$$
$$\langle \circ ; R \rangle = R$$

Negation

$$\overline{[R, T]} = (\bar{R}, \bar{T})$$
$$\overline{(R, T)} = [\bar{R}, \bar{T}]$$
$$\overline{\langle R; T \rangle} = \langle \bar{R}; \bar{T} \rangle$$

$$\bar{\circ} = \circ$$
$$\bar{\bar{R}} = R$$

System BV of the Calculus of Structures

$$\mathsf{ai}\!\downarrow \frac{S\{\circ\}}{S[a,\bar{a}]} \qquad \mathsf{s}\,\frac{S([R,U],T)}{S[(R,T),U]} \qquad \mathsf{q}\!\downarrow \frac{S\langle [R,U];[T,V]\rangle}{S[\langle R;T\rangle,\langle U;V\rangle]}$$

System BV of the Calculus of Structures

$$\mathsf{ai}\!\downarrow \frac{S\{\circ\}}{S[a,\bar{a}]} \qquad \mathsf{s}\, \frac{S([R,U],T)}{S[(R,T),U]} \qquad \mathsf{q}\!\downarrow \frac{S\langle [R,U];[T,V]\rangle}{S[\langle R;T\rangle,\langle U;V\rangle]}$$

$$\downarrow \qquad \downarrow$$

$$\text{MLL} \left\{ \mathsf{ai}\!\downarrow \frac{S\{1\}}{S[a,\bar{a}]} \qquad \mathsf{s}\, \frac{S([R,U],T)}{S[(R,T),U]} \right.$$

System BV of the Calculus of Structures

$$\mathsf{ai}{\downarrow}\,\frac{S\{\circ\}}{S[a,\bar{a}]} \qquad \mathsf{s}\,\frac{S([R,U],T)}{S[(R,T),U]} \qquad \mathsf{q}{\downarrow}\,\frac{S\langle[R,U];[T,V]\rangle}{S[\langle R;T\rangle,\langle U;V\rangle]}$$

$$\downarrow \qquad\qquad \downarrow$$

$$\text{MLL} \left\{ \mathsf{ai}{\downarrow}\,\frac{S\{1\}}{S[a,\bar{a}]} \qquad \mathsf{s}\,\frac{S([R,U],T)}{S[(R,T),U]} \right.$$

All the systems in the calculus of structures follows this scheme.

Classical Logic in the Calculus of Structures

$$\mathsf{ai}\!\downarrow \frac{S\{\circ\}}{S[a,\bar{a}]} \qquad \mathsf{s}\,\frac{S([R,U],T)}{S[(R,T),U]} \qquad \mathsf{q}\!\downarrow \frac{S\langle [R,U];[T,V]\rangle}{S[\langle R;T\rangle,\langle U;V\rangle]}$$

Classical Logic in the Calculus of Structures

$$\mathsf{ai}{\downarrow}\,\frac{S\{\circ\}}{S[a,\bar{a}]} \qquad \mathsf{s}\,\frac{S([R,U],T)}{S[(R,T),U]} \qquad \mathsf{q}{\downarrow}\,\frac{S\langle[R,U];[T,V]\rangle}{S[\langle R;T\rangle,\langle U;V\rangle]}$$

$$\downarrow \qquad \downarrow$$

$$\mathsf{KSg}\left\{\begin{array}{cc} \mathsf{ai}{\downarrow}\,\dfrac{S\{\mathrm{tt}\}}{S[a,\bar{a}]} & \mathsf{s}\,\dfrac{S([R,U],T)}{S[(R,T),U]} \\[2ex] \mathsf{w}{\downarrow}\,\dfrac{S\{\mathrm{ff}\}}{S\{R\}} & \mathsf{c}{\downarrow}\,\dfrac{S[R,R]}{S\{R\}} \end{array}\right.$$

[Brünnler,CSL'03]

Reducing Nondeterminism

In BV, the rule s can be applied to $[(a, b), \bar{a}, \bar{b}]$ in 12 different ways:

$$\mathsf{s}\,\frac{([\bar{a}, a, b], \bar{b})}{[(\bar{a}, \bar{b}), a, b]} \qquad \mathsf{s}\,\frac{[([\bar{a}, b], \bar{b}), a]}{[(\bar{a}, \bar{b}), a, b]} \qquad \mathsf{s}\,\frac{[(\bar{a}, \bar{b}, a), b]}{[(\bar{a}, \bar{b}), a, b]}$$

Reducing Nondeterminism

In BV, the rule s can be applied to $[(a, b), \bar{a}, \bar{b}]$ in 12 different ways:

$$\mathsf{s}\,\frac{([\bar{a}, a, b], \bar{b})}{[(\bar{a}, \bar{b}), a, b]} \qquad \mathsf{s}\,\frac{[([\bar{a}, b], \bar{b}), a]}{[(\bar{a}, \bar{b}), a, b]} \qquad \mathsf{s}\,\frac{[(\bar{a}, \bar{b}, a), b]}{[(\bar{a}, \bar{b}), a, b]}$$

Observation: Switch rule breaks the "interaction" between atoms.

$$\mathsf{s}\,\frac{S([R, W], T)}{S[(R, T), W]}$$

Reducing Nondeterminism

In BV, the rule s can be applied to $[(a, b), \bar{a}, \bar{b}]$ in 12 different ways:

$$\mathsf{s}\frac{([\bar{a}, a, b], \bar{b})}{[(\bar{a}, \bar{b}), a, b]} \qquad \mathsf{s}\frac{[([\bar{a}, b], \bar{b}), a]}{[(\bar{a}, \bar{b}), a, b]} \qquad \mathsf{s}\frac{[(\bar{a}, \bar{b}, a), b]}{[(\bar{a}, \bar{b}), a, b]}$$

Observation: Switch rule breaks the "interaction" between atoms.

$$\mathsf{lis}\frac{S([R, W], T)}{S[(R, T), W]} \qquad \text{if} \quad \mathsf{at}\,\bar{R} \cap \mathsf{at}\,W \neq \emptyset$$

Definition: System BVsl is the system $\{\mathsf{ai}{\downarrow}, \mathsf{lis}, \mathsf{q}{\downarrow}\}$.

Lazy Interaction Switch

Consider:

$$[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]$$

$$\mathsf{s}\,\frac{S([R, W], T)}{S[(R, T), W]}$$

Lazy Interaction Switch

Consider:

$$[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]$$

$$\mathsf{s}\,\frac{S([R, W], T)}{S[(R, T), W]}$$

- The rule s can be applied to this structure in 42 different ways.
  (In system KSg, in 111 different ways.)

Lazy Interaction Switch

Consider:

$$[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]$$

$$\text{lis}\,\frac{S([R, W], T)}{S[(R, T), W]} \qquad \text{if} \quad \text{at}\,\bar{R} \cap \text{at}\,W \neq \emptyset$$

- The rule lis can be applied in 14 different ways.

Lazy Interaction Switch

Consider:

$$\frac{[b,([a,\bar{a}],\bar{b},[c,d,(\bar{c},\bar{d},[e,f,(\bar{e},\bar{f})])])]}{[a,b,(\bar{a},\bar{b},[c,d,(\bar{c},\bar{d},[e,f,(\bar{e},\bar{f})])])]}$$

$$\mathsf{lis}\,\frac{S([R,W],T)}{S[(R,T),W]} \qquad \text{if} \quad \mathsf{at}\,\bar{R} \cap \mathsf{at}\,W \neq \emptyset$$

- The rule lis can be applied in 14 different ways.

$$\{a\} \cap \{a\} \neq \emptyset$$

Lazy Interaction Switch

Consider:

$$\frac{[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [f, ([e, \bar{e}], \bar{f})])])]}{[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]}$$

$$\mathsf{lis}\,\frac{S([R, W], T)}{S[(R, T), W]} \qquad \text{if} \quad \mathsf{at}\,\bar{R} \cap \mathsf{at}\,W \neq \emptyset$$

- The rule lis can be applied in 14 different ways.

$$\{e\} \cap \{e\} \neq \emptyset$$

Lazy Interaction Switch

Consider:

$$\frac{[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [f, ([e, \bar{e}], \bar{f})])])]}{[b, a, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]}$$

$$\mathsf{lis}\,\frac{S([R, W], T)}{S[(R, T), W]} \qquad \text{if} \quad \text{at}\,\bar{R} \cap \text{at}\,W \neq \emptyset$$

- The rule s can be applied to this structure in 42 different ways. (In system KSg, in 111 different ways.)
- The rule lis can be applied in 14 different ways.

$$\{a\} \cap \{\bar{b}, c, \bar{c}, d, \bar{d}, e, \bar{e}, f, \bar{f}\} = \emptyset$$

Reducing Nondeterminism

$$\mathsf{lis}\,\dfrac{\mathsf{ai}\!\downarrow\dfrac{\mathsf{ai}\!\downarrow\dfrac{\circ}{[b,\bar{b}]}}{[([a,\bar{a}],\bar{b}),b]}}{[(\bar{a},\bar{b}),a,b]}$$

$$\mathsf{lis}\,\dfrac{\mathsf{lis}\,\dfrac{\mathsf{ai}\!\downarrow\dfrac{\mathsf{ai}\!\downarrow\dfrac{\circ}{[a,\bar{a}]}}{([a,\bar{a}],[b,\bar{b}])}}{[([a,\bar{a}],\bar{b}),b]}}{[(\bar{a},\bar{b}),a,b]}$$

$$\mathsf{lis}\,\dfrac{\mathsf{lis}\,\dfrac{\mathsf{ai}\!\downarrow\dfrac{\mathsf{ai}\!\downarrow\dfrac{\circ}{[b,\bar{b}]}}{([a,\bar{a}],[b,\bar{b}])}}{[([a,\bar{a}],\bar{b}),b]}}{[(\bar{a},\bar{b}),a,b]}$$

$$\mathsf{lis}\,\dfrac{\mathsf{ai}\!\downarrow\dfrac{\mathsf{ai}\!\downarrow\dfrac{\circ}{[a,\bar{a}]}}{[a,(\bar{a},[b,\bar{b}])]}}{[(\bar{a},\bar{b}),a,b]}$$

$$\mathsf{lis}\,\dfrac{\mathsf{lis}\,\dfrac{\mathsf{ai}\!\downarrow\dfrac{\mathsf{ai}\!\downarrow\dfrac{\circ}{[b,\bar{b}]}}{([a,\bar{a}],[b,\bar{b}])}}{[a,(\bar{a},[b,\bar{b}])]}}{[(\bar{a},\bar{b}),a,b]}$$

$$\mathsf{lis}\,\dfrac{\mathsf{lis}\,\dfrac{\mathsf{ai}\!\downarrow\dfrac{\mathsf{ai}\!\downarrow\dfrac{\circ}{[a,\bar{a}]}}{([a,\bar{a}],[b,\bar{b}])}}{[a,(\bar{a},[b,\bar{b}])]}}{[(\bar{a},\bar{b}),a,b]}$$

Reducing Nondeterminism

$$\mathsf{lis}\,\dfrac{\mathsf{ai}{\downarrow}\,\dfrac{\mathsf{ai}{\downarrow}\,\dfrac{\circ}{[b,\bar{b}]}}{[([a,\bar{a}],\bar{b}),b]}}{[(\bar{a},\bar{b}),a,b]}
\qquad
\mathsf{lis}\,\dfrac{\mathsf{lis}\,\dfrac{\mathsf{ai}{\downarrow}\,\dfrac{\mathsf{ai}{\downarrow}\,\dfrac{\circ}{[a,\bar{a}]}}{([a,\bar{a}],[b,\bar{b}])}}{[([a,\bar{a}],\bar{b}),b]}}{[(\bar{a},\bar{b}),a,b]}
\qquad
\mathsf{lis}\,\dfrac{\mathsf{lis}\,\dfrac{\mathsf{ai}{\downarrow}\,\dfrac{\mathsf{ai}{\downarrow}\,\dfrac{\circ}{[b,\bar{b}]}}{([a,\bar{a}],[b,\bar{b}])}}{[([a,\bar{a}],\bar{b}),b]}}{[(\bar{a},\bar{b}),a,b]}$$

$$\mathsf{lis}\,\dfrac{\mathsf{ai}{\downarrow}\,\dfrac{\mathsf{ai}{\downarrow}\,\dfrac{\circ}{[a,\bar{a}]}}{[a,(\bar{a},[b,\bar{b}])]}}{[(\bar{a},\bar{b}),a,b]}
\qquad
\mathsf{lis}\,\dfrac{\mathsf{lis}\,\dfrac{\mathsf{ai}{\downarrow}\,\dfrac{\mathsf{ai}{\downarrow}\,\dfrac{\circ}{[b,\bar{b}]}}{([a,\bar{a}],[b,\bar{b}])}}{[a,(\bar{a},[b,\bar{b}])]}}{[(\bar{a},\bar{b}),a,b]}
\qquad
\mathsf{lis}\,\dfrac{\mathsf{lis}\,\dfrac{\mathsf{ai}{\downarrow}\,\dfrac{\mathsf{ai}{\downarrow}\,\dfrac{\circ}{[a,\bar{a}]}}{([a,\bar{a}],[b,\bar{b}])}}{[a,(\bar{a},[b,\bar{b}])]}}{[(\bar{a},\bar{b}),a,b]}$$

In system $\{\mathsf{s},\mathsf{ai}{\downarrow}\}$ in the proof search space of $[(\bar{a},\bar{b}),a,b]$, there are 358 derivations including these 6 proofs, and no other proofs.

Reducing Nondeterminism

Definition: System BVsl is the system $\{\mathsf{ai}\downarrow, \mathsf{lis}, \mathsf{q}\downarrow\}$.

Theorem: Systems $\{\mathsf{ai}\downarrow, \mathsf{s}, \mathsf{q}\downarrow\}$ (BV) and BVsl are equivalent. [LPAR'06]

Reducing Nondeterminism

Definition: System BVsl is the system $\{\mathsf{ai}\!\downarrow, \mathsf{lis}, \mathsf{q}\!\downarrow\}$.

Theorem: Systems $\{\mathsf{ai}\!\downarrow, \mathsf{s}, \mathsf{q}\!\downarrow\}$ (BV) and BVsl are equivalent. [LPAR'06]

Corollary: Systems $\{\mathsf{ai}\!\downarrow, \mathsf{s}\}$ and $\{\mathsf{ai}\!\downarrow, \mathsf{lis}\}$ are equivalent. [LPAR'06]

Reducing Nondeterminism

Definition: System BVsl is the system $\{\mathsf{ai}\!\downarrow, \mathsf{lis}, \mathsf{q}\!\downarrow\}$.

Theorem: Systems $\{\mathsf{ai}\!\downarrow, \mathsf{s}, \mathsf{q}\!\downarrow\}$ (BV) and BVsl are equivalent. [LPAR'06]

Corollary: Systems $\{\mathsf{ai}\!\downarrow, \mathsf{s}\}$ and $\{\mathsf{ai}\!\downarrow, \mathsf{lis}\}$ are equivalent. [LPAR'06]

Theorem: The cut rule is admissible for system BVsl. [Tech.Rep.06]

Reducing Nondeterminism

Definition: System BVsl is the system $\{ai\downarrow, lis, q\downarrow\}$.

Theorem: Systems $\{ai\downarrow, s, q\downarrow\}$ (BV) and BVsl are equivalent. [LPAR'06]

Corollary: Systems $\{ai\downarrow, s\}$ and $\{ai\downarrow, lis\}$ are equivalent. [LPAR'06]

Theorem: The cut rule is admissible for system BVsl. [Tech.Rep.06]

Theorem: System BV is NP-Complete. [WOLLIC'06]

Classical Logic in the Calculus of Structures

$$\text{ai}{\downarrow}\,\frac{S\{\circ\}}{S[a,\bar{a}]} \qquad \text{s}\,\frac{S([R,U],T)}{S[(R,T),U]} \qquad \text{q}{\downarrow}\,\frac{S\langle[R,U];[T,V]\rangle}{S[\langle R;T\rangle,\langle U;V\rangle]}$$

$\downarrow$ $\downarrow$

$$\text{KSg}\left\{\begin{array}{ll} \text{ai}{\downarrow}\,\dfrac{S\{\mathsf{tt}\}}{S[a,\bar{a}]} & \text{s}\,\dfrac{S([R,U],T)}{S[(R,T),U]} \\[2ex] \text{w}{\downarrow}\,\dfrac{S\{\mathsf{ff}\}}{S\{R\}} & \text{c}{\downarrow}\,\dfrac{S[R,R]}{S\{R\}} \end{array}\right.$$

[Brünnler,CSL'03]

Reducing Nondeterminism in Classical Logic System KSg

Theorem: A structure R has a proof in KSg iff

$$\begin{array}{l} \| \{\mathsf{s}, \mathsf{ai}{\downarrow}\} \\ R'' \\ \| \{\mathsf{w}{\downarrow}\} \\ R' \\ \| \{\mathsf{c}{\downarrow}\} \\ R \end{array} \qquad \begin{array}{l} [\mathsf{tt}, \mathsf{tt}] = \mathsf{tt} \\ (\mathsf{ff}, \mathsf{ff}) = \mathsf{ff} \end{array}$$

Reducing Nondeterminism in Classical Logic System KSg

Theorem: A structure R has a proof in KSg iff

$$\begin{array}{l} \parallel \{\mathsf{s},\mathsf{ai}\downarrow\} \\ R'' \\ \parallel \{\mathsf{w}\downarrow\} \\ R' \\ \parallel \{\mathsf{c}\downarrow\} \\ R \end{array}$$

$$[\mathrm{tt}, \mathrm{tt}] = \mathrm{tt}$$
$$(\mathrm{ff}, \mathrm{ff}) = \mathrm{ff}$$

Definition: System KSgi is the system resulting from replacing the switch rule in system KSg with the lazy interaction switch rule.

Reducing Nondeterminism in Classical Logic System KSg

Theorem: A structure R has a proof in KSg iff

$$\begin{array}{l} \| \{\mathsf{s},\mathsf{ai}\!\downarrow\} \\ R'' \\ \| \{\mathsf{w}\!\downarrow\} \\ R' \\ \| \{\mathsf{c}\!\downarrow\} \\ R \end{array} \qquad \begin{array}{l} [\mathsf{tt},\mathsf{tt}] = \mathsf{tt} \\ (\mathsf{ff},\mathsf{ff}) = \mathsf{ff} \end{array}$$

Definition: System KSgi is the system resulting from replacing the switch rule in system KSg with the lazy interaction switch rule.

Theorem: Systems KSg and KSgi are equivalent. [LPAR'06]

Implementation in Maude

- Systems in the calculus of structures can be expressed as term rewriting systems modulo equational theories.
  [K, Hölldobler, TR-04]

Implementation in Maude

- Systems in the calculus of structures can be expressed as term rewriting systems modulo equational theories.
  [K, Hölldobler, TR-04]
- Inference rules can be expressed as (conditional) rewrite rules, modulo equality. For instance, the rule lis becomes

$$[(R, T), W] \rightarrow ([R, W], T) \quad \text{if} \quad \mathsf{at}\, R \cap \mathsf{at}\, W \neq \emptyset$$

Implementation in Maude

- Systems in the calculus of structures can be expressed as term rewriting systems modulo equational theories.
  [K, Hölldobler, TR-04]

- Inference rules can be expressed as (conditional) rewrite rules, modulo equality. For instance, the rule lis becomes

$$[(R, T), W] \to ([R, W], T) \quad \text{if} \quad \mathsf{at}\, R \cap \mathsf{at}\, W \neq \emptyset$$

- Language Maude allows implementing term rewriting systems modulo associativity, commutativity and unit(s).

Implementation in Maude

- Systems in the calculus of structures can be expressed as term rewriting systems modulo equational theories.
  [K, Hölldobler, TR-04]
- Inference rules can be expressed as (conditional) rewrite rules, modulo equality. For instance, the rule lis becomes

$$[(R, T), W] \rightarrow ([R, W], T) \quad \text{if} \quad \mathsf{at}\,R \cap \mathsf{at}\,W \neq \emptyset$$

- Language Maude allows implementing term rewriting systems modulo associativity, commutativity and unit(s).
- Maude has a built-in breadth-first search function.

Implementation in Maude

- Systems in the calculus of structures can be expressed as term rewriting systems modulo equational theories. [K, Hölldobler, TR-04]
- Inference rules can be expressed as (conditional) rewrite rules, modulo equality. For instance, the rule lis becomes

$$[(R, T), W] \rightarrow ([R, W], T) \quad \text{if} \quad \mathsf{at}\, R \cap \mathsf{at}\, W \neq \emptyset$$

- Language Maude allows implementing term rewriting systems modulo associativity, commutativity and unit(s).
- Maude has a built-in breadth-first search function.
- Systems of the calculus of structures can be easily implemented by resorting to the simple high level language of Maude. [ESSLLI'04,ISCIS'04]

Example: Maude Module for System BV

```
mod BV is
 sorts Atom Unit Structure .
 subsort Atom < Structure .
 subsort Unit < Structure .

  op o  : -> Unit .
  op -_      : Atom -> Atom [ prec 50 ] .
  op [_,_]   : Structure Structure -> Structure [assoc comm id: o] .
  op {_,_}   : Structure Structure -> Structure [assoc comm id: o] .
  op <_;_>   : Structure Structure -> Structure [assoc id: o] .
  ops a b c d e : -> Atom .

  var R T U V : Structure .      var A : Atom .

 rl [ai-down] : [ A , - A ]              => o .
 rl [s]       : [ { R , T } , U ]        => { [ R , U ] , T } .
 rl [q-down]  : [ < R ; T > , < U ; V > ] => < [R,U] ; [T,V] > .
endm
```

Automated Proof Search

1. $[a, b, (\bar{a}, \bar{b}, [a, b, (\bar{a}, \bar{b})])]$
2. $[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]$

Query	System	# states explored	finds a proof in # ms (cpu)
1.	$\{$s, ai$\downarrow\}$	1041	100
	$\{$lis, ai$\downarrow\}$	264	0
2.	$\{$s, ai$\downarrow\}$	–	–
	$\{$lis, ai$\downarrow\}$	6595	1370

Lazy Interaction Switch Revisited

Consider:

$$\frac{[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [f, ([e, \bar{e}], \bar{f})])])]}{[b, a, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]}$$

$$\mathsf{lis}\,\frac{S([R, W], T)}{S[(R, T), W]} \qquad \text{if} \quad \text{at}\,\bar{R} \cap \text{at}\,W \neq \emptyset$$

- The rule s can be applied to this structure in 42 different ways. (In system KSg, in 111 different ways.)
- The rule lis can be applied in 14 different ways.

$$\{a\} \cap \{\bar{b}, c, \bar{c}, d, \bar{d}, e, \bar{e}, f, \bar{f}\} = \emptyset$$

Lazy Interaction Switch Revisited

Consider:

$$\frac{[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [f, ([e, \bar{e}], \bar{f})])])]}{[b, a, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]}$$

$$\text{lis}\,\frac{S([R, W], T)}{S[(R, T), W]} \qquad \text{if} \quad \text{at}\,\bar{R} \cap \text{at}\,W \neq \emptyset$$

- The rule s can be applied to this structure in 42 different ways. (In system KSg, in 111 different ways.)
- The rule lis can be applied in 14 different ways.

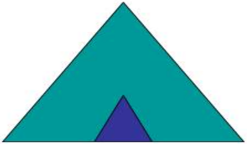
$$\{a\} \cap \{\bar{b}, c, \bar{c}, d, \bar{d}, e, \bar{e}, f, \bar{f}\} = \emptyset$$

The condition of the rule must be performed for 42 such substructures.

This is expensive in proof search.

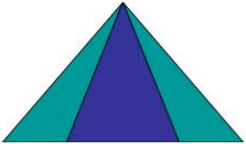
Deep Inference vs. Deepest Inference

Idea: When we restrict the application of the inference rules to the deepest redexes, we are restricted to the smaller substructures.



Deep Inference vs. Deepest Inference

Idea: When we restrict the application of the inference rules to the deepest redexes, we are restricted to the smaller substructures.

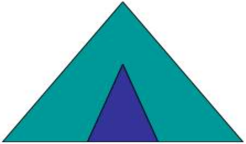


- Sequent calculus (shallow inference) is complete.

$$[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]$$

Deep Inference vs. Deepest Inference

Idea: When we restrict the application of the inference rules to the deepest redexes, we are restricted to the smaller substructures.



- Sequent calculus (shallow inference) is complete.

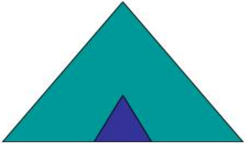
$$[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]$$

- Deep inference is complete.

$$[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]$$

Deep Inference vs. Deepest Inference

Idea: When we restrict the application of the inference rules to the deepest redexes, we are restricted to the smaller substructures.



- Sequent calculus (shallow inference) is complete.

$$[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]$$

- Deep inference is complete.

$$[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]$$

Is there a plausible notion of "deepest inference" that is complete?

$$[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]$$

Deepest Switch

Definition: A instance of the switch rule

$$\mathsf{s}\,\frac{S([R, W], T)}{S[(R, T), W]}$$

is an instance of deep switch (ds) if $R \neq (R_1, R_2)$ and $W \neq [W_1, W_2]$.

Deepest Switch

Definition: A instance of the switch rule

$$\mathsf{s}\,\frac{S([R, W], T)}{S[(R, T), W]}$$

is an instance of deep switch (ds) if $R \neq (R_1, R_2)$ and $W \neq [W_1, W_2]$.

Example:

$$[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]$$

Deepest Switch

Definition: A instance of the switch rule

$$\mathsf{s}\,\frac{S([R, W], T)}{S[(R, T), W]}$$

is an instance of deep switch (ds) if $R \neq (R_1, R_2)$ and $W \neq [W_1, W_2]$.

Example:

$$[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]$$

Proposition: Switch rule can be replaced with the rule deep switch in system KSg without losing completeness.

Deepest Switch

Definition: A instance of the switch rule

$$\mathsf{s}\,\frac{S([R, W], T)}{S[(R, T), W]}$$

is an instance of deep switch (ds) if $R \neq (R_1, R_2)$ and $W \neq [W_1, W_2]$.

Example:

$$[a, b, (\bar{a}, \bar{b}, [c, d, (\bar{c}, \bar{d}, [e, f, (\bar{e}, \bar{f})])])]$$

Proposition: Switch rule can be replaced with the rule deep switch in system KSg without losing completeness.

Proposition: Every proof in system $\{\mathsf{ai}\!\downarrow, \mathsf{s}\}$ can be transformed to a proof in $\{\mathsf{ai}\!\downarrow, \mathsf{ds}\}$ in linear time.

Deepest Lazy Interaction Switch?

Integrating the ideas from deep switch and lazy interaction switch provides further reduction in nondeterminism.

Deepest Lazy Interaction Switch?

Integrating the ideas from deep switch and lazy interaction switch provides further reduction in nondeterminism.

Theorem: We can replace the switch rule with the deep lazy interaction switch rule without losing completeness.

Deepest Lazy Interaction Switch?

Integrating the ideas from deep switch and lazy interaction switch provides further reduction in nondeterminism.

Theorem: We can replace the switch rule with the deep lazy interaction switch rule without losing completeness.

Further Questions:

- Proof complexity analysis for the developed techniques.

Deepest Lazy Interaction Switch?

Integrating the ideas from deep switch and lazy interaction switch provides further reduction in nondeterminism.

Theorem: We can replace the switch rule with the deep lazy interaction switch rule without losing completeness.

Further Questions:

- Proof complexity analysis for the developed techniques.
- Extending and implementing the techniques to other systems for other logics, e.g., LL, Modal Logics, NEL.

Deepest Lazy Interaction Switch?

Integrating the ideas from deep switch and lazy interaction switch provides further reduction in nondeterminism.

Theorem: We can replace the switch rule with the deep lazy interaction switch rule without losing completeness.

Further Questions:

- Proof complexity analysis for the developed techniques.
- Extending and implementing the techniques to other systems for other logics, e.g., LL, Modal Logics, NEL.
- Providing a confluent deductive system for MLL for structures consisting of pairwise distinct atoms. [Guerrini, 1999]

Summary

- Systems with deep inference bring shorter proofs but also greater nondeterminism in proof search.

Summary

- Systems with deep inference bring shorter proofs but also greater nondeterminism in proof search.
- By means of a new general purely proof theoretical technique, this nondeterminism can be reduced.

Summary

- Systems with deep inference bring shorter proofs but also greater nondeterminism in proof search.
- By means of a new general purely proof theoretical technique, this nondeterminism can be reduced.
- This technique can be used as a proof theoretic tool while proving properties of deductive systems, e.g., system BV is NP-Complete.

Summary

- Systems with deep inference bring shorter proofs but also greater nondeterminism in proof search.
- By means of a new general purely proof theoretical technique, this nondeterminism can be reduced.
- This technique can be used as a proof theoretic tool while proving properties of deductive systems, e.g., system BV is NP-Complete.
- Deep inference systems can implemented as term rewriting systems modulo equality.

Summary

- Systems with deep inference bring shorter proofs but also greater nondeterminism in proof search.
- By means of a new general purely proof theoretical technique, this nondeterminism can be reduced.
- This technique can be used as a proof theoretic tool while proving properties of deductive systems, e.g., system BV is NP-Complete.
- Deep inference systems can implemented as term rewriting systems modulo equality.
  - Maude modules and papers are available for download at `http://www.iccl.tu-dresden.de~/ozan/maude_cos.html` GRAPE: `http://grape.sourceforge.net/`

Summary

- Systems with deep inference bring shorter proofs but also greater nondeterminism in proof search.
- By means of a new general purely proof theoretical technique, this nondeterminism can be reduced.
- This technique can be used as a proof theoretic tool while proving properties of deductive systems, e.g., system BV is NP-Complete.
- Deep inference systems can implemented as term rewriting systems modulo equality.
  - Maude modules and papers are available for download at `http://www.iccl.tu-dresden.de˜/ozan/maude_cos.html` GRAPE: `http://grape.sourceforge.net/`
  - TOM/Java implementation available at the TOM distribution: `http://tom.loria.fr` `http://tom.loria.fr/examples/structures/BV.html`

Summary

- Systems with deep inference bring shorter proofs but also greater nondeterminism in proof search.
- By means of a new general purely proof theoretical technique, this nondeterminism can be reduced.
- This technique can be used as a proof theoretic tool while proving properties of deductive systems, e.g., system BV is NP-Complete.
- Deep inference systems can implemented as term rewriting systems modulo equality.
  - Maude modules and papers are available for download at `http://www.iccl.tu-dresden.de˜/ozan/maude_cos.html` GRAPE: `http://grape.sourceforge.net/`
  - TOM/Java implementation available at the TOM distribution: `http://tom.loria.fr` `http://tom.loria.fr/examples/structures/BV.html`
- The technique for reducing nondeterminism provides a performance improvement in implementations.