Implementing Deep Inference

24.11.04, LORIA, Nancy

Ozan Kahramanoğulları

Institut für Informatik, Universität Leipzig
International Center for Computational Logic, TU Dresden
ozan@informatik.uni-leipzig.de

Outline

- Motivation
- The Calculus of Structures & System BV
- From Derivations to Rewritings
- The Implementation
- Tuning the Performance: System BVn & System BVu
- Performance Comparison
- Conclusion and Future Work

Motivation:

– general recipe for implementing deductive systems with deep inference

– tools for developing the proof theory of different logics

– implementing logic BV: a self-dual, non-commutative operator

- Process Algebra:
  $a\,,b$ processes : $a.b \neq b.a$
  CCS vs. BV   [Bruscoli, 02]

- Planning:
  $\langle\, openDoor\,;\, enterRoom\,\rangle \neq \langle\, enterRoom\,;\, openDoor\,\rangle$
  Conjunctive Planning Problems   [K, AIA'05]

- (Natural Language Processing)

The Calculus of Structures & System BV

- [Guglielmi, 99] proof theoretical formalism,
  generalization of the one-sided sequent calculus

- Deep inference: Rules can be applied deep inside a structure.

$$\rho \, \frac{S\{R\}}{S\{T\}}$$

- System BV:
  MLL $+ \; mix \; + \; mix0 +$ a non-commutative, self-dual operator

- BV structures:

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

- Structures are considered equivalent modulo an equational theory.

Syntactic Equivalence of BV Structures

Associativity

$$[\vec{R}, [\vec{T}]] = [\vec{R}, \vec{T}]$$

$$(\vec{R}, (\vec{T})) = (\vec{R}, \vec{T})$$

$$\langle\langle\vec{R}\rangle; \langle\vec{T}\rangle\rangle = \langle\vec{R}; \vec{T}\rangle$$

Commutativity

$$[\vec{R}, \vec{T}] = [\vec{T}, \vec{R}]$$

$$(\vec{R}, \vec{T}) = (\vec{T}, \vec{R})$$

Unit

$$(\circ, R) = R \qquad \langle \circ\,; R\rangle = R$$

$$[\circ, R] = R \qquad \langle R\,; \circ\rangle = R$$

Context Closure

if $R = T$ then $S\{R\} = S\{T\}$, $\bar{R} = \bar{T}$

Negation

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

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

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

$$\overline{\langle R; T\rangle} = \langle\bar{R}; \bar{T}\rangle$$

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

System BV

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

From Derivations to Rewritings

(1.) Explicit equality steps

Example:

$$s \frac{[(c,[\bar{b},b]),\bar{c}]}{[b,(\bar{b},c),\bar{c}]} \quad \rightsquigarrow \quad s \frac{= \dfrac{[(c,[\bar{b},b]),\bar{c}]}{[([\bar{b},b],c),\bar{c}]}}{= \dfrac{[[(\bar{b},c),b],\bar{c}]}{[b,(\bar{b},c),\bar{c}]}}$$

(2.) From n-ary operators to binary operators

Example: $n22(\,[\bar{a},b,(a,c)]\,) = [\bar{a},[b,(a,c)]]$

(3.) From structures to terms

$\Sigma = \{\circ\,,\,\bar{\_}\,,\ \langle\_;\_\rangle\,,[\_,\_]\,,(\_,\_)\,\} \cup \{a \mid\ a \text{ is a positive atom}\}$

From Derivations to Rewritings

(4.) From inference rules to rewrite rules

Associativity *Commutativity* *Unit*

$$\mathsf{E} = \left\{ \begin{array}{lll} & & \langle \circ ; R \rangle \approx R \; , \\ \langle R ; \langle S ; T \rangle \rangle \approx \langle \langle R ; S \rangle ; T \rangle \; , & [R, T] \approx [T, R] \; , & \langle R ; \circ \rangle \approx R \; , \\ [R, [S, T]] \approx [[R, S], T] \; , & (R, T) \approx (T, R) \; , & [\circ, R] \approx R \; , \\ (R, (S, T)) \approx ((R, S), T) \; , & & (\circ, R) \approx R \end{array} \right\}$$

$$\mathsf{R} = \left\{ \begin{array}{llcll} \mathsf{ai}\!\downarrow & : \; [a, \bar{a}] & \rightarrow & \circ & , \\ \mathsf{s} & : \; [(R, T), U] & \rightarrow & ([R, U], T) & , \\ \mathsf{q}\!\downarrow & : \; [\langle R ; R' \rangle, \langle T ; T' \rangle] & \rightarrow & \langle [R, T] ; [R', T'] \rangle & \end{array} \right\}$$

Orienting the Equalities for Negation

Definition: A Σ-term s is in negation normal form iff the negation is pushed to the leaves (atoms) and no unit $\circ$ appears in it.

$$\mathcal{R}_{Neg} = \left\{ \begin{array}{l} \bar{\circ} \rightarrow \circ\,, \\ \overline{\langle R; T \rangle} \rightarrow \langle \bar{R}; \bar{T} \rangle\,, \\ \overline{[R, T]} \rightarrow (\bar{R}, \bar{T})\,, \\ \overline{(R, T)} \rightarrow [\bar{R}, \bar{T}]\,, \\ \bar{\bar{R}} \rightarrow R \end{array} \right\}$$

Lemma: Term rewriting systems $\mathcal{R}_{Neg}$ is terminating and confluent.

Lemma: For Σ-term s, the normal form of s with respect to $\mathcal{R}_{Neg}$ is in negation normal form.

From Derivations to Rewritings

$$s \longrightarrow_{\mathsf{R}/\mathsf{E}} t \quad \text{iff} \quad (\exists s', t')\; s \approx_{\mathsf{E}} s' \longrightarrow_{\mathsf{R}} t' \approx_{\mathsf{E}} t$$

Example:

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

$$\rightsquigarrow \qquad [\bar{a}, [b, (a, c)]] \approx_{\mathsf{E}}$$

$$[b, [(a, c), \bar{a}]] \longrightarrow_{\mathsf{R}} [b, ([a, \bar{a}], c)]$$

$$\approx_{\mathsf{E}} [b, (c, [a, \bar{a}])]$$

Proposition: Let s and t be two Σ-terms or structures, where t is in negation normal form.
There is a derivation in BV from s to t having length n iff there exists a rewriting $s \xrightarrow{*}_{R_{Neg}} s' \xrightarrow{n}_{\mathsf{R}/\mathsf{E}} t$.

Implementation in Maude

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

- Language Maude allows implementing term rewriting systems modulo associativity, commutativity and unit(s).
  [K, ESSLLI'04]

- Since Maude 2 breadth-first search is available.

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
```

Implementation in Maude

```
Maude> search [- c,[< a ; {c,- b} >,< - a ; b >]] =>+ o .

search in BV : [- c,[< a ; {c,- b} >,< - a ; b >]] =>+ o .
Solution 1 (state 2229)
states: 2230  rewrites: 196866 in 930ms cpu (950ms real) (211683
    rewrites/second)
empty substitution

No more solutions.
states: 2438  rewrites: 306179 in 1460ms cpu (1490ms real) (209711
    rewrites/second)
```

- Units cause redundant applications of the inference rules which do not lead to a proof.

Removing the Units: Some Definitions

A structure is in *unit normal form* when the only negated structures appearing in it are atoms and no unit $\circ$ appears in it.

Example: $\overline{\langle [b, (\bar{a}, c), \circ, \bar{c}]; a \rangle} = \langle ([a, \bar{c}], \bar{b}, c); \bar{a} \rangle$

Two systems $\mathscr{S}$ and $\mathscr{S}'$ are *equivalent* if for every proof of a structure T in system $\mathscr{S}$, there exists a proof of T in system $\mathscr{S}'$, and vice versa.

System BV $\leadsto$ System BVn

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

$$\leadsto$$

$$\mathsf{u}_1{\downarrow}\,\frac{S\{R\}}{S[R, \circ]} \qquad \mathsf{u}_2{\downarrow}\,\frac{S\{R\}}{S(R, \circ)}$$

$$\mathsf{u}_3{\downarrow}\,\frac{S\{R\}}{S\langle R; \circ \rangle} \qquad \mathsf{u}_4{\downarrow}\,\frac{S\{R\}}{S\langle \circ; R \rangle}$$

Proposition: Every BV structure S can be transformed to one of its unit normal forms S' by applying only the rules $\{\mathsf{u}_1{\downarrow}, \mathsf{u}_2{\downarrow}, \mathsf{u}_3{\downarrow}, \mathsf{u}_4{\downarrow}\}$ bottom-up and the equalities for negation from left to right.

System BV $\rightsquigarrow$ System BVn

$$\mathsf{s}\,\frac{S([R,T],U)}{S[(R,U),T]} \quad \rightsquigarrow \quad \mathsf{s}_1\,\frac{S([R,T],U)}{S[(R,U),T]} \qquad \mathsf{s}_2\,\frac{S(R,T)}{S[R,T]}$$

$$\mathsf{q}{\downarrow}\,\frac{S\langle[R,T];[U,V]\rangle}{S[\langle R;U\rangle,\langle T;V\rangle]} \quad \rightsquigarrow \quad \begin{array}{cc} \mathsf{q}_1{\downarrow}\,\dfrac{S\langle[R,T];[U,V]\rangle}{S[\langle R;U\rangle,\langle T;V\rangle]} & \mathsf{q}_2{\downarrow}\,\dfrac{S\langle R;T\rangle}{S[R,T]} \\ \mathsf{q}_3{\downarrow}\,\dfrac{S\langle[R,T];U\rangle}{S[R,\langle T;U\rangle]} & \mathsf{q}_4{\downarrow}\,\dfrac{S\langle T;[R,U]\rangle}{S[R,\langle T;U\rangle]} \end{array}$$

Proposition: The rules $\mathsf{q}_1{\downarrow}$, $\mathsf{q}_2{\downarrow}$, $\mathsf{q}_3{\downarrow}$, $\mathsf{q}_4{\downarrow}$, s_1 , and s_2 are sound (derivable) for system BV.

System BVn

Theorem: For every derivation $\begin{array}{c} W \\ \Delta \Big\Vert \textsf{BV} \\ Q \end{array}$ there exists a derivation $\begin{array}{c} W' \\ \Delta' \Big\Vert \textsf{BVn} \\ Q \end{array}$ where W' is a unit normal form of the structure W.

Corollary: System BV and system BVn are equivalent.

System BVn $\rightsquigarrow$ System BVu

$$\mathsf{ai}\downarrow \frac{S\{\circ\}}{S[a,\bar{a}]}$$

$$\mathsf{u}_1\downarrow \frac{S\{R\}}{S[R,\circ]} \qquad \mathsf{u}_2\downarrow \frac{S\{R\}}{S(R,\circ)} \qquad \rightsquigarrow \qquad \mathsf{ai}_1\downarrow \frac{S\{R\}}{S[R,[a,\bar{a}]]} \qquad \mathsf{ai}_2\downarrow \frac{S\{R\}}{S(R,[a,\bar{a}])}$$

$$\mathsf{u}_3\downarrow \frac{S\{R\}}{S\langle R;\circ\rangle} \qquad \mathsf{u}_4\downarrow \frac{S\{R\}}{S\langle\circ;R\rangle} \qquad \mathsf{ai}_3\downarrow \frac{S\{R\}}{S\langle R;[a,\bar{a}]\rangle} \qquad \mathsf{ai}_4\downarrow \frac{S\{R\}}{S\langle[a,\bar{a}];R\rangle}$$

$$\circ\downarrow \frac{}{\circ} \qquad \rightsquigarrow \qquad ax\,\frac{}{[a,\bar{a}]}$$

Corollary: System BV and system BVu are equivalent.

Performance Comparison

Example (proof search): `[- c,[< a ; {c,- b} >,< - a ; b >]]`

	finds a proof		search terminates	
	in # millisec.	after # rewrites	in # millisec.	after # rewrites
BV	950	196866	1490	306179
BVn	120	12610	120	12720
BVu	10	1416	60	4691

Proposition: If a structure R in unit normal form with n occurences of positive atoms has a proof in BVn with length k, then R has a proof in BVu with length $k - n$.

Conclusion

- System BV (and other systems in the calculus of structures) can be expressed as term rewriting systems modulo equational theories which can be implemented in Maude.
- BVn, BVu : systems equivalent to BV where equalities for unit become redundant. [K, ISCIS'04]
- These systems provide a better performance in proof search.
- These results can be analogously applied to other systems in the calculus of structures.
- Implementations are available for download at `http://www.informatik.uni-leipzig.de/~ozan/maude_cos.html` .

Future Work

- Using the expressive power of TOM & Java for implementing efficient search strategies.
- Applying local search techniques for more efficient proof search.
- Developing proof theoretical strategies, e.g., splitting theorem, together with other methods attached to rewriting calculus.