Proof-theoretic aspects of hybrid type-logical grammars

Richard Moot and Symon Jory Stevens-Guille

Introduction: Hybrid Type-Logical Grammars

- Hybrid type-logical grammars (HTLG) are a logic introduced by Kubota and Levine (2012)
- HTLG combines the standard Lambek grammar implications with the lambda grammar operations
- It provides a simple account of many phenomena on the syntax-semantics interface, for which neither of its subsystems has equally simple solutions
- Kubota and Levine (2013) 'acknowledge that there remains an important theoretical issue: the formal properties of our hybrid implicational logic are currently unknown'

Natural deduction: Gentzen style

$$\frac{}{p^s : w \vdash M : A}\ \text{Lex} \qquad \frac{}{x^\alpha : A \vdash x^\alpha : A}\ \text{Ax}$$

$$\frac{\Gamma \vdash N^\alpha : A \qquad \Delta \vdash M^{\alpha\to\beta} : A \multimap B}{\Gamma, \Delta \vdash (MN)^\beta : B}\ \multimap E$$

$$\frac{\Gamma, x^\alpha : A \vdash M^\beta : B}{\Gamma \vdash (\lambda x.M)^{\alpha\to\beta} : A \multimap B}\ \multimap I$$

$$\frac{\Gamma \vdash M^s : A/B \qquad \Delta \vdash N^s : B}{\Gamma, \Delta \vdash (M+N)^s : A}\ /E$$

$$\frac{\Gamma, p^s : A \vdash (M+p)^s : B}{\Gamma \vdash M^s : B/A}\ /I$$

$$\frac{\Delta \vdash M^s : B \qquad \Gamma \vdash N^s : B\backslash A}{\Delta, \Gamma \vdash (M+N)^s : A}\ \backslash E$$

$$p^s : A, \Gamma \vdash (p+M)^s : B$$

Natural deduction: Prawitz style

$$\frac{N^\alpha : A \quad M^{\alpha\to\beta} : A \multimap B}{(MN)^\beta : B} \multimap E \qquad \frac{\begin{matrix}[x^\alpha : A]^i \\ \vdots \\ M^\beta : B\end{matrix}}{(\lambda x.M)^{\alpha\to\beta} : A \multimap B} \multimap I_i$$

$$\frac{M^s : A \quad N^s : A\backslash B}{(M+N)^s : B} \backslash E \qquad \frac{\begin{matrix}[x^s : A]^i \\ \vdots \\ (x+M)^s : B\end{matrix}}{M^s : A\backslash B} \backslash I_i$$

$$\frac{M^s : B/A \quad N^s : A}{(M+N)^s : B} /E \qquad \frac{\begin{matrix}[x^s : A]^i \\ \vdots \\ (M+x)^s : B\end{matrix}}{M^s : B/A} /I_i$$

Example

$$\dfrac{\dfrac{\dfrac{\dfrac{\lambda P.(P\,e) : (np \multimap s) \multimap s \qquad \dfrac{\dfrac{[x : np]^1 \quad [y : np\backslash s]^2}{x + y : s}\,\backslash E}{\lambda x.(x + y) : np \multimap s}\,\multimap I_1}{(\lambda P.(P\,e))(\lambda x.(x + y)) : s}\,\multimap E}{(\lambda x.(x + y))e : s}\,\beta}{e + y : s}\,\beta}{e : s/(np\backslash s)}\,/I_2$$

Substitution Lemma

Lemma

Let δ_1 be a proof of $\Gamma \vdash N : A$ and δ_2 a proof of $\Delta, x : A \vdash M[x] : C$ such that N and M share no free variables, then there is a proof of $\Gamma, \Delta \vdash M[N] : C$.

$$\begin{array}{c} \vdots\ \delta_1 \\ N : A \end{array} \qquad\qquad \begin{array}{c} x : A \\ \vdots\ \delta_2 \\ M[x] : C \end{array}$$

$$\begin{array}{c} \vdots\ \delta_1 \\ N : A \\ \vdots\ \delta_2[x := N] \\ M[N] : C \end{array}$$

Reduction steps

$$
\dfrac{\begin{array}{c}\vdots\ \Pi_1\\ N^s : A\end{array} \quad \dfrac{\begin{array}{c}x^s : A\\ \vdots\ \Pi_2\\ (x+M)^s : B\end{array}}{M^s : A\backslash B}\ \backslash I}{(N+M)^s : B}\ \backslash E
\quad\rightsquigarrow\quad
\begin{array}{c}\vdots\ \Pi_1\\ N^s : A\\ \vdots\ \Pi_2[x := N]\\ (N+M)^s : B\end{array}
$$

$$
\dfrac{\begin{array}{c}\vdots\ \Pi_1\\ N^\alpha : A\end{array} \quad \dfrac{\begin{array}{c}x^\alpha : A\\ \vdots\ \Pi_2\\ M^\beta : B\end{array}}{(\lambda x.M)^{\alpha\rightarrow\beta} : A \multimap B}\ \multimap I}{((\lambda x.M)N)^\beta : B}\ \multimap E
\quad\rightsquigarrow\quad
\begin{array}{c}\vdots\ \Pi_1\\ N^\alpha : A\\ \vdots\ \Pi_2[x := N]\\ (M[x := N])^\beta : B\end{array}
$$

Strong normalisation

Theorem

HTLG is strongly normalizing.

Proof.

To show strong normalization, we need to show that there are no infinite reduction sequences. Since each reduction reduces the size of the proof, this is trivial. □

Confluence

Theorem

Normalization for HTLG proofs is confluent.

Proof.

It is easy to show weak confluence: whenever a proof can be reduced by two different reductions R_1 and R_2, then reducing either redex will preserve the other redex , and R_1 followed by R_2 will produce the same proof as R_2 followed by R_1.
Since we have already shown strong normalisation, weak confluence entails strong confluence by Newman's Lemma. □

Unicity of normal forms and subformula property

Corollary

HTLG proofs have a unique normal form.

Proof.

Immediate by strong normalisation and confluence. □

Corollary

HTLG proofs have the subformula property.

Proof.

In a normal form proof, every formula is either a subformula of one of the hypotheses or a subformula of the conclusion. □

Taking stock

- we have shown several basic properties for HTLG natural deduction showing the logic introduced by Kobuta and Levine is well-behaved,
- we generally want a logic to have multiple proof systems,
- it is fairly easy to give a sequent calculus for HTLG,
- instead we will look at *proof nets* for HTLG.

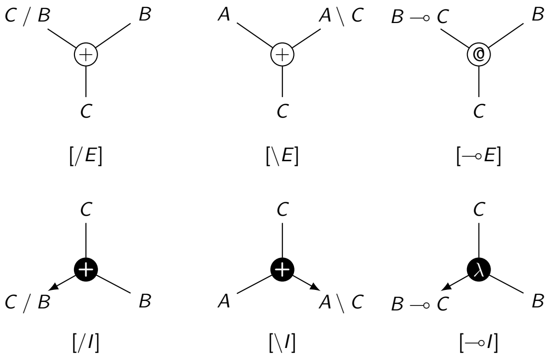
Proof nets

- introduced for linear logic as a way to represent proofs as (hyper)graphs,
- removes 'boring' rule permutations of the sequent calculus,
- easily extends to other connectives ('$\bullet$', '$\Diamond$', '$\Box$') and structural rules,
- natural combinatorial representation of the search space for proofs (facilitating proof search and complexity analysis).

Proof net parsing

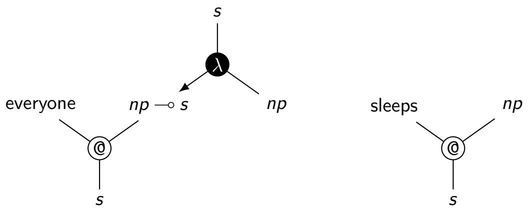
1. deterministically unfold formulas using links ($\approx$ logical rules),
2. connect atomic formulas to form proof structures (proof candidates),
3. verify correctness of the proof candidate using graph-theoretic propertie of the underlying graph (here: contractions in the style of Danos).

Links



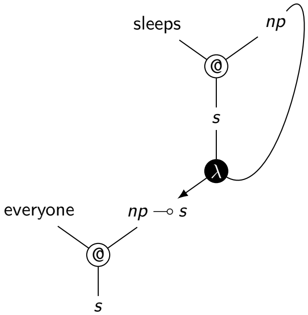
Example proof structure

Given lexical entries '*everyone*' with formula $(np \multimap s) \multimap s$ and '*sleeps*' with for $np \multimap s$, formula unfolding produces the following.

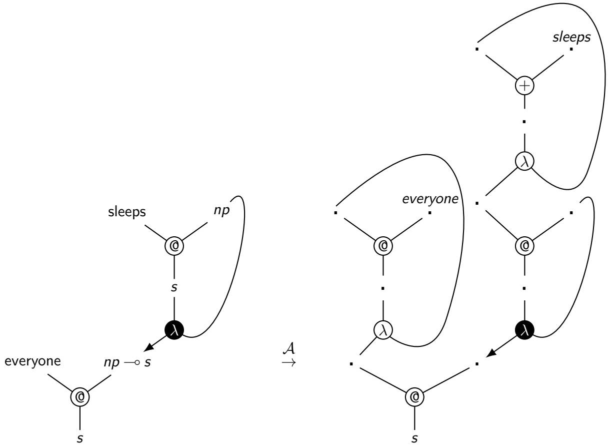


Example proof structure

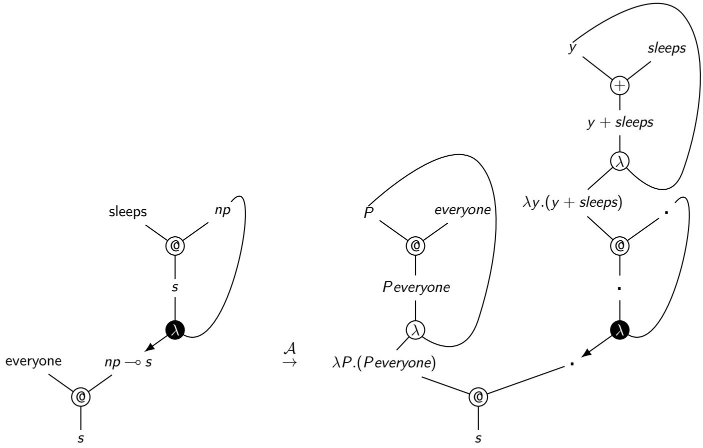
Connecting the *np* and *s* atomic formulas produces the following proof structure from the types of ‘everyone’ and ‘sleeps’ to the type *s*.



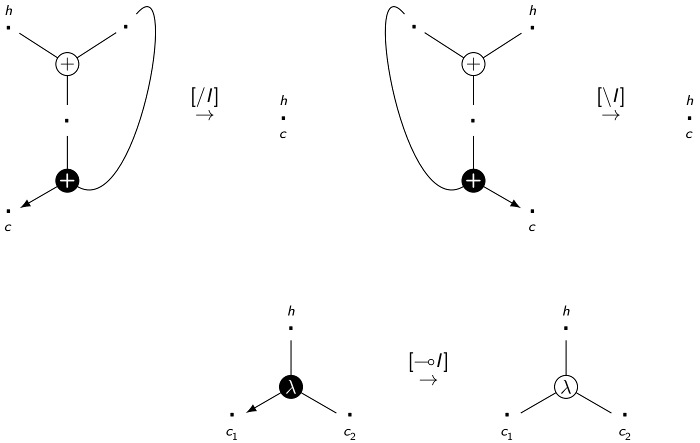
Converting a proof structure to an abstract proof structure



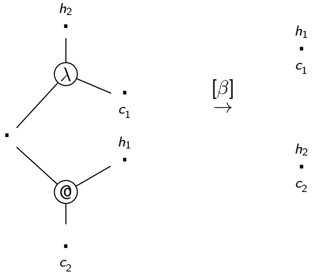
Converting a proof structure to an abstract proof structure



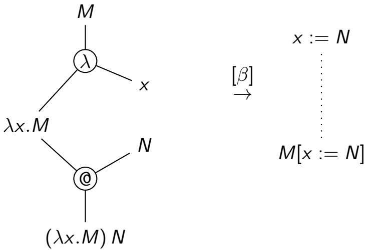
Contractions



Structural rules



Structural rules



Soundness and completeness

Definition

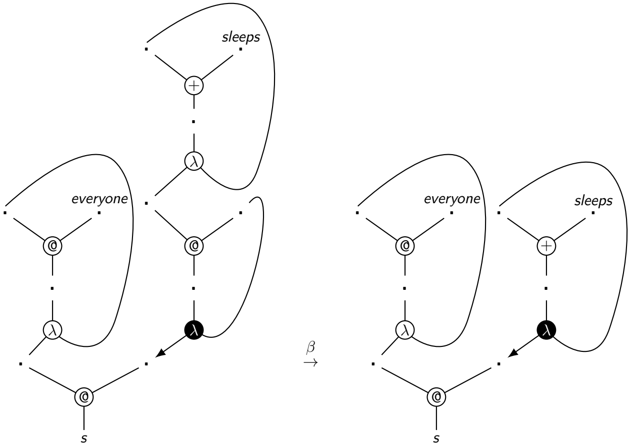
a proof structure is a *proof net* iff its abstract proof structure contracts to a linear lambda term (represented as a graph).

We need to show:

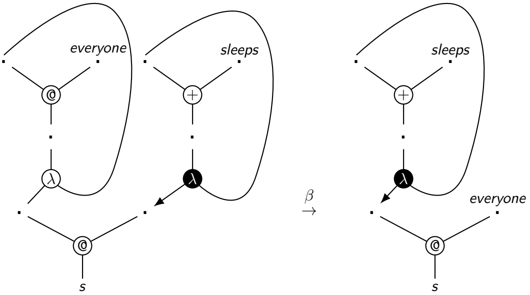
1. for every natural deduction proof, there is a proof net
2. for every proof net there is a natural deduction proof

in both cases the lambda term of the natural deduction proof must correspond to the lambda term of the contracted proof net.

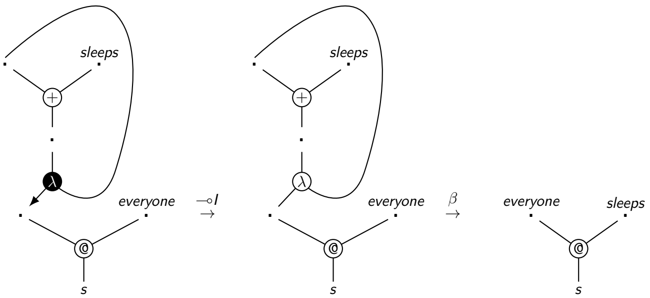
Example



Example



Example



Proof: soundness

Lemma

If δ is a natural deduction proof of $N_1 : A_1, \ldots, N_k : A_k \vdash M : C$, then we can construct a proof net with premisses $A_1, \ldots, A_n$ and conclusion C contracting to M.

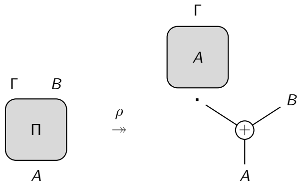
Proof Easy induction on the length of the natural deduction proof and case analysis on the last rule. We show only one of the introduction rules.

Proof: soundness

If the last rules is /I we are in the following case.

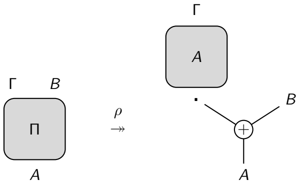
$$\frac{\begin{array}{c}[x : B]^i \\ \vdots\ \delta \\ N + x : A\end{array}}{N : A/B}\ /I_i$$

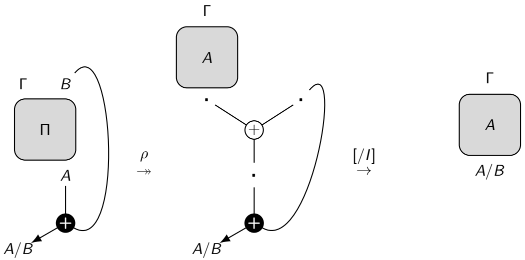
Induction hypothesis gives us a proof net contracting to $N + x$.



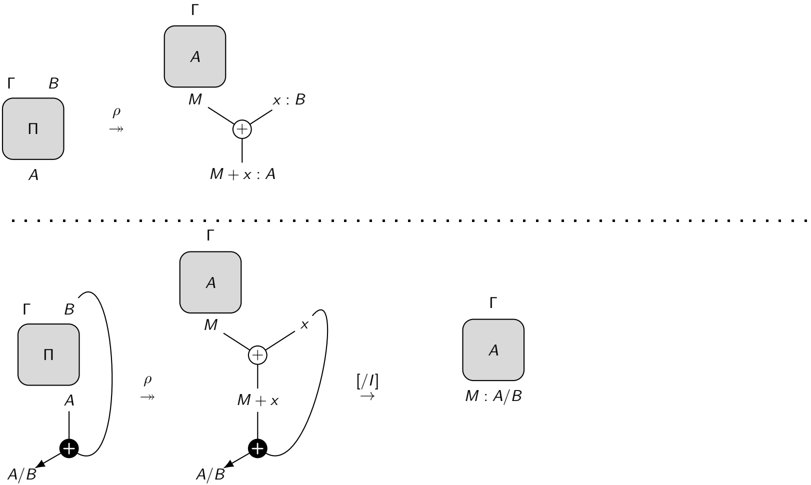
We extend this to a proof net of $N : A/B$ as follows.

Proof: soundness





Proof: soundness



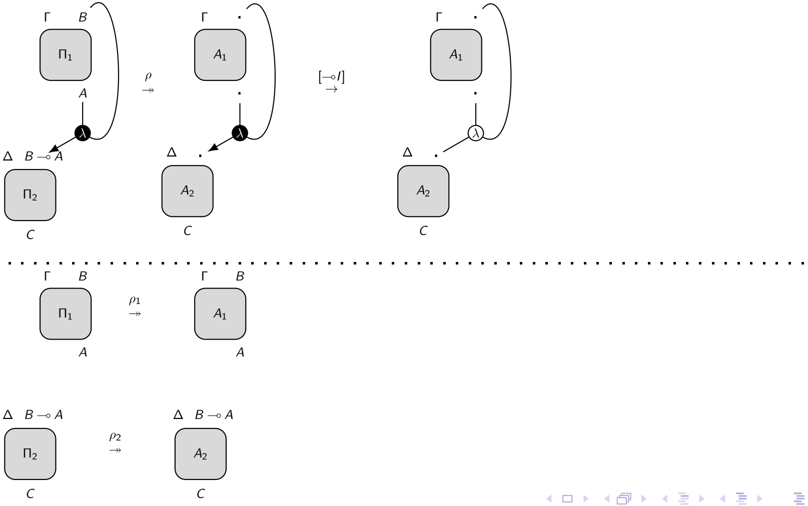
Proof: completeness

Lemma

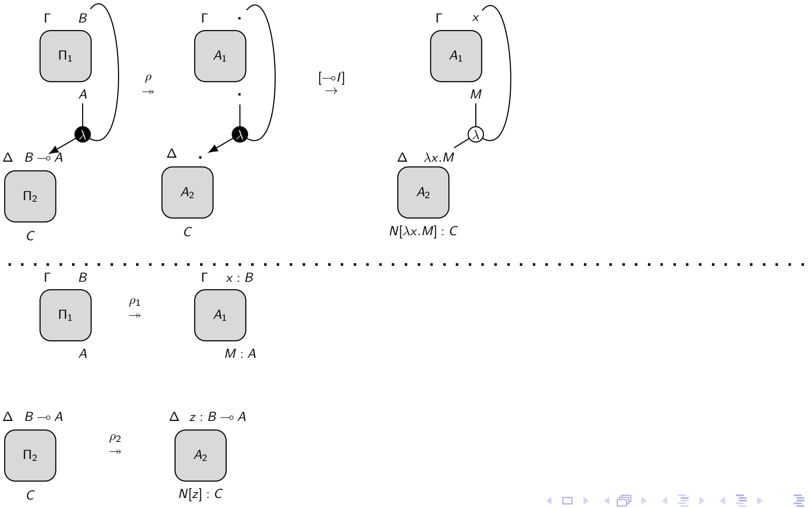
Given a proof net Π with premisses $A_1, \ldots A_n$ and conclusion C converting to a lambda graph M, there is a natural deduction proof $N_1 : A_1, \ldots, N_k : A_n \vdash M : C$.

Proof We only show the case for $\multimap I$. The other cases are simple adaptations of multimodal proof nets.

Proof: completeness



Proof: completeness



Proof: completeness

Induction hypothesis gives us a proof $\delta 1$ of $\Gamma, x : B \vdash M : A$ and a proof δ_2 of $\Delta, z : B \vdash N[z] : C$.
We can combine these as follows to produce the required proof of $\Gamma, \Delta \vdash N[\lambda x.M] : C$.

$$\dfrac{\begin{array}{c} x : B \\ \vdots\ \delta_1 \\ M : A \end{array}}{\lambda x.M : B \multimap A} \multimap I$$

$$\begin{array}{c} \lambda x.M : B \multimap A \\ \vdots\ \delta_2, z := \lambda x.M \\ N[\lambda x.M] : C \end{array}$$

What have we gained?

- new proof calculus for HTLG,
- generic and flexible proof search procedure,
- the proof net calculus can easily be extended to incorporate the '$\bullet$', '$\Diamond$' and '$\Box$' connectives, as well as multimodality and structural rules,
- simplifies complexity analysis of HTLG and its variants.

Confluence

Lemma

The conversions for abstract proof structures are confluent.

Proof.

Weak confluence is easy to show: every contraction preserves all other redexes (using the fact that all lambda-terms are linear). □

We can preserve confluence when adding associativity (compiling away associativity using *n*-premiss links) and the identity element (requires adding a contraction for '/' and '\').

Complexity of contractibility

Lemma

We can verify contractibility of a proof structure in polynomial time.

Proof.

Given confluence and the fact that all contractions reduce the size of the abstract proof structure, there are at most $O(n^2)$ contraction steps (where n is the size of the abstract proof structure). □

The lemma holds even in the multimodal case with associativity for any number of the binary modes.

NP completeness

Theorem

HTLG parsing is NP complete

Proof.

Since HTLG contains (lexicalized) ACG and (implicational) Lambek calculus as subsystems, NP hardness is immediate.
To show HTLG parsing is NP complete, it suffices to show that we can verify that a proof candidate is a proof (i.e. that a proof structure is a proof net) in polynomial time. □

Complexity: the general case

Theorem

multimodal HTLG parsing with an arbitrary set of non-expanding structural rule is NP hard and in PSPACE.

Proof.

It is easy to show that in this general case, HTLG parsing can be done in nondeterministic polynomial space. Given PSPACE completeness of multimodal type-logical grammars (with a variable, non-expanding structural rule component), the theorem follows. □

- We have investigated the proof-theoretic foundations of Hybrid Type-Logical Grammars and proved several basic properties of the natural deduction calculus (strong normalisation, subformula property, uniqueness of normal forms),
- we have introduced a proof net calculus for HTLG,
- we have analysed the complexity of HTLG and some of its variants.