Expansion tree proofs with unification

Willem Heijltjes, Michelangelo Mecozzi

University of Bath

***Abstract*—Expansion tree proofs are a representation of proofs in first-order classical logic that factors out both propositional proof content and the bureaucracy of sequent calculus. Their design is therefore aimed at precisely exposing witnessing terms, which by Herbrand's theorem are the essential information of a proof. In this paper, we will descire and use unification based algorithms to reconstruct witnessing term substitutions in expansion tree proofs, yielding a variant that omits such redundancy and further generalises their proof representation.**

I. INTRODUCTION

In 1930 Jacques Herbrand [1] proved that terms witnessing the truth of a statement are the essential information one can extract from a proof of such. This result inspired Dale Miller [5] in the creation of expansion tree proofs, graphical constructions that capture this minimal information. The economical design characterising this constructions raises a natural question as to whether this essential information is itself redundant and could be abstracted to generate a more general proof. The recent works of Hughes [2] in MLL1 and of Heijltjes, Hughes and Strassburger [3] in ALL1 show that in other contexts unification based algorithms can be adapted to reconstruct witnessing terms. In particular, the technique of coalescence described in the latter directly applies in this context: it performs a natural computation of the witnessing term substitution by starting from new axiomatic unifiable leaves, to which all most general unifiers are attached, and removed as the overall proof is reconstructed. What is left is a substitution that reconstructs a most general witnessing term usable in the the regular sequent calculus derivation. As a result, this produces a more general proof that is equivalent to others up to substitution. To achieve the same result in expansion tree proofs, we will first introduce a new version of the sequent calculus that abstracts witnessing terms. This will then form the underlying structure for our new version of expansion tree proofs, and we will explicitly describe the coalesence algorithm used to reconstruct witnessing terms.

II. EXPANSION TREE PROOFS

In this paper, we represent expansion tree proofs as graphs where nodes are inference rules and edges are labeled by formulas, to which we apply an acyclical partial order representing the backtracking; branching from a single node represents contraction. Due to its design, the underlying sequent calculus will be restricted to the following inferences:

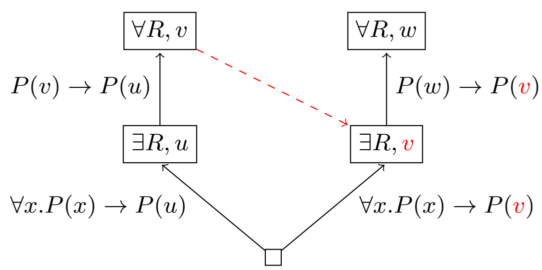
$$\frac{}{\vdash A_1, \ldots, A_n}\ Axiom^{*} \qquad \frac{\vdash A[a/x], \Gamma}{\vdash \forall x.A, \Gamma}\ \forall^{**}$$

$$\frac{\vdash A[t/x], \Gamma}{\vdash \exists x.A, \Gamma}$$

$$\frac{\vdash \Gamma}{\vdash \exists x.A, \Gamma} \qquad \frac{\vdash \exists x.A, \exists x.A, \Gamma}{\vdash \exists x.A, \Gamma}$$

$$^{*} \bigvee_{i=1}^{n} A_i \text{ a tautology} \qquad ^{**} a \notin FV(\Gamma)$$

A typical example of proof representation via ETPs is the Drinker's formula $\exists x.\forall y.P(y) \to P(x)$



Note the root indeed contains the empty inference in accordance to the sequent calculus having no inference coming from the statement that is being proved.

III. UNIFIABLE SEQUENT CALCULUS

As a stepping stone, we discuss how one can construct a sequent calculus without witnessing terms, which we denote $\mathbf{LK^U}$. By inductive arguments and through the usage of most general unifiers, one can instead define an explicit algorithm that, given a formula, abstracts term instantiation in the $\exists R$ inferences and constructs a substitution that *unifies* expressions in the axioms.

We start by considering the usual inference of the form

$$\frac{\vdash A[t/x], \Gamma}{\vdash \exists x.A, \Gamma}\ \exists R, t$$

where t is a witnessing term that needs to be chosen so that

$$\frac{}{\vdash A(t), \neg A(t)}$$

becomes a valid axiom in the sequent calculus derivation. Instead, one can substitute by a fresh variable α

$$\frac{\vdash A[\alpha/x], \Gamma}{\vdash \exists x.A, \Gamma}\ \exists R, \alpha$$

which now becomes free in A, and for the relative axiomatic expression

$$\frac{}{\vdash A(\alpha, \alpha_1, \ldots, \alpha_n), \overline{A(\beta_1, \ldots, \beta_m)}}$$

we construct the substitution

$$\sigma \coloneqq MGU(A(\alpha, \alpha_1, \ldots, \alpha_n), A(\beta_1, \ldots, \beta_m))$$

These expressions must be unifiable since a witnessing term must exists since the formula was originally proved in the regular sequent calculus. When performing contractions on existential quantifier expressions, we always ensure that we are injecting fresh variables as they will need to form the domain of a well defined substitution. Conjunction and Cut Rule need a further step as they must *merge* two distinct unifiers that may disagree on common domain elements. Namely, we must merge σ_1 and σ_2 coming from sub-proofs π_1 and π_2, respectively:

$$\pi\ \frac{\overset{\pi_1}{\vdash A, \Gamma_1} \qquad \overset{\pi_2}{\vdash B, \Gamma_2}}{\vdash A \wedge B, \Gamma_1, \Gamma_2}$$

$$\pi\ \frac{\overset{\pi_1}{\vdash A, \Gamma_1} \qquad \overset{\pi_2}{\vdash \overline{B}, \Gamma_2}}{\vdash \Gamma_1, \Gamma_2}$$

To that end, we define the *merger* of two substitutions σ and τ to be the more general substitution

$$\sigma \curlyvee \tau \coloneqq \rho \circ \sigma = \rho \circ \tau$$

where ρ is defined to be the *merging substitution* of σ and τ. A natural way of computing such a substitution starts by considering $\vec{v} = dom(\sigma), \vec{w} = dom(\tau)$ and running the MGU algorithm on the ordered list $<\vec{v} \cap \vec{w}>$ under the two initial substitutions, i.e. for some constant c we consider

$$MGU(\sigma(c{<}\vec{v} \cap \vec{w}{>}), \tau(c{<}\vec{v} \cap \vec{w}{>})) = c\,\vec{t}$$

and construct ρ s.t. $(\vec{v} \cap \vec{w})_i \mapsto \vec{t_i}$ for each index i, in increasing order.

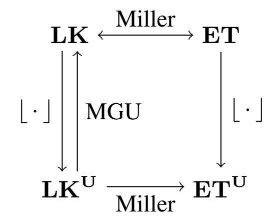
We now can merge unifiers into a single substitution that unifies each axiomatic expression. Indeed, let $\vdash\ A_i, \overline{B_i}$ be an axiomatic expression of subtree π_i, unified by σ_i. Then

$$\sigma_i(A_i) = \sigma_i(B_i) \implies \rho(\sigma_i(A_i)) = \rho(\sigma_i(B_i))$$

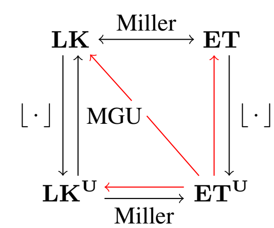
Thus one can *propagate* these unifications throughout the tree derivation and achieve a most general one that unifies all the axioms.

IV. Unified expansion trees

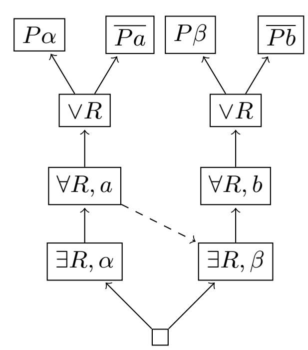
Consider the following diagram:



where $\lfloor \cdot \rfloor$ denotes the Forgetful functor. The goal of this section is to use the natural algorithm called coalescence to define the remaining functors and complete the diagram, i.e. we try to construct

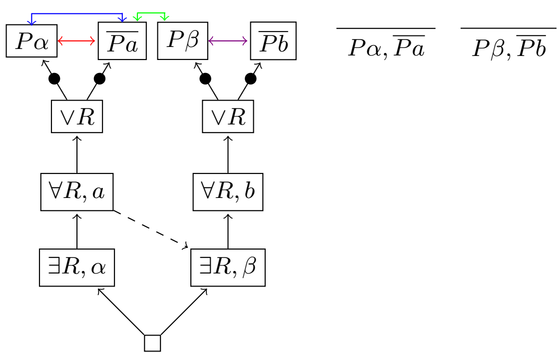


where the red functors are given by coalescence. This technique starts by considering the set of all possible substitutions coming from unifying purely propositional leaves of the original expansion tree, and using *restrictions* comunig from existential and universal quantifier inferences to reduce the size of such a set, up to the point of having exactly substitutions that give witnessing terms. We first depict the coalescence process via an example. Consider again the ET proof for the Drinker's formula $\exists x.\forall y.P(x) \vee \overline{P(y)}$, with expansion applied on disjunctions:



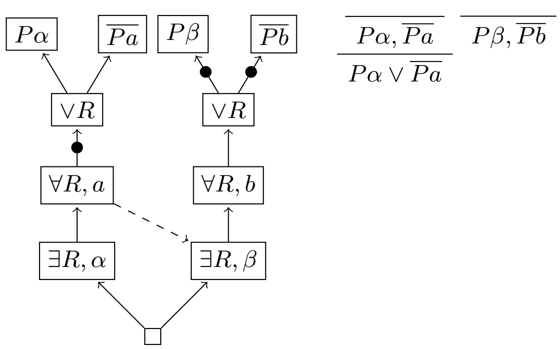
We will first compute all the possible unifiers arising from combining the leaves of the tree, and then reconstruct the correct one by filtering and shrinking such substitution set (we keep track of such substitutions in the figure below). We will proceed by considering the frontier of edges we currently focus on, starting from the top ones, and proceed in the direction of the root. At the same time, we will also look at the current sequent proof corresponding to the current frontier.

$$[\alpha \mapsto a][\alpha \mapsto b][\beta \mapsto a][\beta \mapsto b]$$



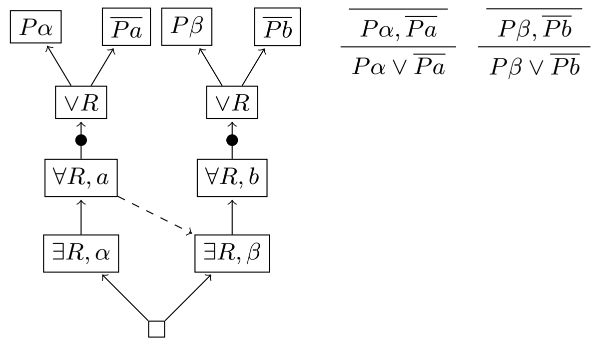
We now make a movement along the tree. In this particular circumstance, we may proceed in any direction we prefer:

$$[\alpha \mapsto a][\alpha \mapsto b][\beta \mapsto a][\beta \mapsto b]$$



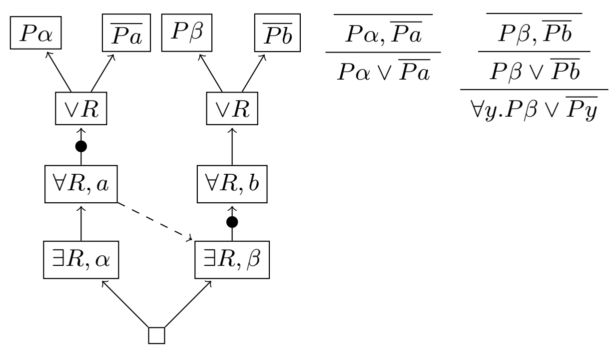
As it does not involve any quantifiers, this particular movement does not affect the set of substitutions currently undert consideration. To keep proceeding, we cannot continue along the $\forall R, a$ node, as the backtracking tells us that such a movement needs to have an effect on the $\exists R, \beta$ node, hence they need to happen in a sequence (which will be illustrated later). We thus proceed in the only possible way:

$$[\alpha \mapsto a][\alpha \mapsto b][\beta \mapsto a][\beta \mapsto b]$$



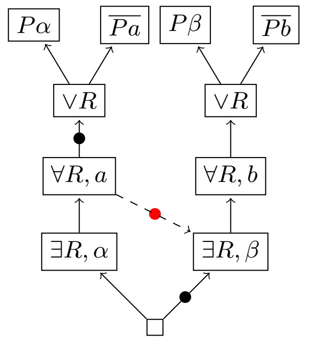
For the same reasons explained before, we must proceed on the right side of the tree. For operations involving universal quantifiers, we must remove all those substitutions containing the corresponding eigenvariable in their image. This because when performing such substitutions we must ensure that the eigenvariable condition is preserved.

$$[\alpha \mapsto a][\beta \mapsto a]$$



The next operation is interesting for two reasons. Firstly, it is an operation involving a backtracking edge, which we deal with by first performing the advancement on the $\exists R, \beta$ node (as this is the one that took advantage of a learning, so must have happened sequentially after) and populate the backtracking edge with a token; this will be necessary for the next operation. Finally, as we are performing an operation on an existential quantifier, we remove from each substitution the injected variable (β in this case) and record all the effects on such a variable:

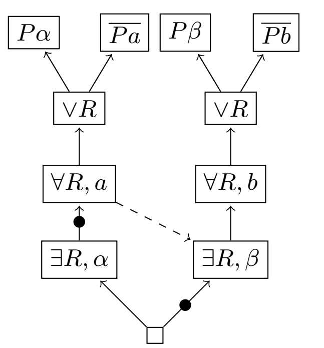
$[\alpha \mapsto a][\emptyset]$ $\beta \mapsto a$



$$\dfrac{\dfrac{}{P\alpha, \overline{Pa}}}{P\alpha \vee \overline{Pa}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{P\beta, \overline{Pb}}}{P\beta \vee \overline{Pb}}}{\forall y.P\beta \vee \overline{Py}}}{\exists x.\forall y.Px \vee \overline{Py}}$$

Now, having an *open gate* on the backtracking edge, represented by the red token, we may proceed on via the $\forall R, \alpha$ node (performing the necessary filtering previously described):

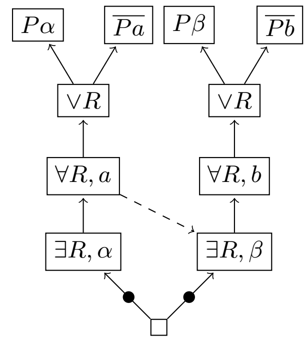
$[\emptyset]$ $\beta \mapsto a$



$$\dfrac{\dfrac{\dfrac{}{P\alpha, \overline{Pa}}}{P\alpha \vee \overline{Pa}}}{\forall y.P\alpha \vee \overline{Py}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{P\beta, \overline{Pb}}}{P\beta \vee \overline{Pb}}}{\forall y.P\beta \vee \overline{Py}}}{\exists x.\forall y.Px \vee \overline{Py}}$$

And finally:

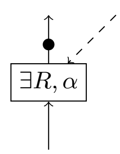
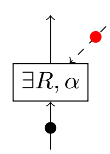
$[\emptyset]$ $\beta \mapsto a$



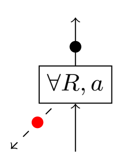
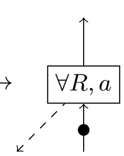
$$\dfrac{\dfrac{\dfrac{\dfrac{}{P\alpha, \overline{Pa}}}{P\alpha \vee \overline{Pa}}}{\forall y.P\alpha \vee \overline{Py}}}{\exists x.\forall y.Px \vee \overline{Py}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{P\beta, \overline{Pb}}}{P\beta \vee \overline{Pb}}}{\forall y.P\beta \vee \overline{Py}}}{\exists x.\forall y.Px \vee \overline{Py}}$$

which ends with a contraction. For such we can deduce that the substitution giving a witnessing term is the result of the remaining empty substitution with their relative recorded effects, namely $\sigma \coloneqq \beta \mapsto a$.

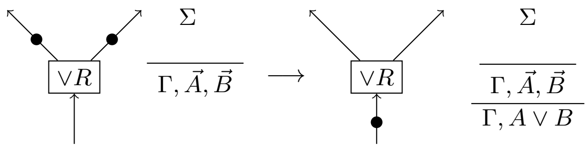
This is an example of the more general algorithm applicable on each branching existential node that requires a witnessing term. For such an algorithm, the formal rules are the following (let $\vec{A} \coloneqq A, \ldots, A$):

 $\dfrac{\Sigma}{\Gamma, A(\vec{\alpha})}$ $\longrightarrow$  $\dfrac{\Sigma \setminus\setminus_{\text{DOM}} \alpha}{\dfrac{\Gamma, A(\vec{\alpha})}{\Gamma, \exists x.A(x)}}$

where $\Sigma \setminus\setminus_{\text{DOM}} \alpha \coloneqq \{\sigma|_{\text{DOM}(\sigma)\setminus\{\alpha\}} \mid \sigma \in \Sigma\}$

 $\dfrac{\Sigma}{\Gamma, A(\vec{a})}$ $\longrightarrow$  $\dfrac{\Sigma \setminus_{\text{IM}} a}{\dfrac{\Gamma, A(\vec{a})}{\Gamma, \forall y.A(y)}}$

where $\Sigma \setminus_{\text{IM}} a \coloneqq \{\sigma \in \Sigma \mid a \notin \text{IM}(\sigma)\}$ and where we also record the effects on α.



V. Conclusions

The coalescence algorithm on an ET^U does not just reconstruct its relative $\mathbf{LK}^U$ proof, but it also provides a substitution map σ (at least one, as the result is assumed to be originally proved in the unificaiton-free case) such that for some ρ:

$$\mathrm{ET} \underset{\rho \circ \sigma}{\overset{\lfloor \cdot \rfloor}{\rightleftarrows}} \mathrm{ET}^U$$

This explicit algorithm therefore enables us to remove the redundant witnessing terms from expansion tree proofs, further contributing to the compactness of such a design. At the current stage, we have strong evidence for the correctness and completeness of this algorithm. We also expect with further work to include conjunction, for which the algorithm will have to be altered or changed.

References

[1] Jacques Herbrand. *Investigations in proof theory: The properties of true propositions*. In: From Frege to Godel: A source book in mathematical logic, 1897–1931, Harvard University Press, pp. 525–581 (1930)

[2] Dominic J. D. Hughes, *Unification nets: canonical proof net quantifiers*, https://arxiv.org/abs/1802.03224, LICS 2018.

[3] Heijltjes, Hughes, and Strassburger, *Proof nets for first-order additive linear logic*, FSCD 2019.

[4] Stefan Hetzl. *A Sequent Calculus with Implicit Term Representation.* Computer Science Logic (CSL) 2010, Aug 2010, Brno, Czech Republic. pp.N/A, 2010. *hal-00498707*.

[5] Miller D.A. (1984) *Expansion Tree Proofs and Their Conversion to Natural Deduction Proofs*. In: Shostak R.E. (eds) 7th International Conference on Automated Deduction. CADE 1984. Lecture Notes in Computer Science, vol 170. Springer, New York, NY.