Applications of positive and intuitionistic bounded arithmetic to proof complexity

Anupam Das

École Normale Supérieure de Lyon

Abstract. We introduce uniform versions of monotone and deep inference proof systems in the setting of *bounded arithmetic*, relating the size of propositional proofs to forms of proof-theoretic strength in weak fragments of arithmetic.
This continues the recent program of studying the complexity of propositional deep inference. In particular this work is inspired by previous work where proofs of the propositional pigeonhole principle in the minimal deep inference system, KS, were constructed using rather abstract uniform templates, exemplifying the pertinence of such an approach.
Along the way we are required to draw on a blend of tools from bounded arithmetic, deep inference, and intuitionistic proof theory in order to achieve the required correspondences.

1 Introduction

Bounded arithmetic has been a fruitful way to relate complexity classes with logical theories and propositional proof systems. For example Paris and Wilkie showed that proofs of Π_1-sentences[1] in the theory $I\Delta_0$[2] translate to classes of polynomial-size Hilbert-Frege proofs of bounded depth [21]; conversely, the soundness of bounded-depth Hilbert-Frege systems can be proved in such theories.

In this work we introduce theories of bounded arithmetic for various *monotone* and *deep inference* proof systems.

Deep inference proof complexity has received much attention in recent years, and the complexity of the minimal system, KS, is considered as yet unresolved [6] [18] [13]. While an extension of it, KS^+, is known to quasipolynomially simulate Hilbert-Frege systems[3] there is neither such a simulation known for KS nor some nontrivial lower bound separating the two systems.

So-called 'analytic' deep inference systems ([5]) for propositional logic can be viewed as subclasses of (tree-like) monotone proofs (exemplified in e.g. [3] and [18]), first introduced as sequent calculus proofs free of negation steps (MLK) and studied in e.g. [22], [2] and [1], and we exploit this correspondence here.[4]

[1] These are closed formulae of the form $\forall x_1, \ldots, x_n.A(\boldsymbol{x})$, with $A(\boldsymbol{x})$ a Δ_0-formula, i.e. one in which all quantifiers are of the form $\exists x \leq t$ or $\forall x \leq t$ for some term t.

[2] This is the fragment of Peano Arithmetic with induction is restricted to Δ_0-formulae.

[3] In fact, it is widely believed that a polynomial simulation holds.

[4] Indeed, it was in this setting that the aforementioned quasipolynomial simulation was first proved [1].

The starting point of this work is a natural extension of the Paris-Wilkie translation to theories with a positive least fixed point operator, which are mapped to uniform classes of positive propositional formulae of increasing depth, the objects of reasoning in *MLK*. More specifically, inductive arguments conducted over fixed points naturally correspond to deep inference reasoning in the system KS^+ (or tree-like *MLK*) on the translated formulae.

We show that a theory corresponding to KS[5] can thence be obtained by certain logical restrictions on induction, namely requiring inductive steps to have a divide-and-conquer format, known as *PIND* [7]. The proof of this makes use of well-studied normalisation procedures from deep inference proof theory, and in particular certain estimates of their complexity that appeared in [13].

The ideas we present were in fact at the heart of the design of quasipolynomial-size proofs of the propositional pigeonhole principle in KS in [14]. In that work the arguments were presented internally to propositional logic for accessibility, but for this reason we believe that our theories might be useful in future research.

In order to obtain converse statements to these simulations, usually in the form of 'reflection' principles in bounded arithmetic, we turn to *intuitionistic* variants of our bounded arithmetic theories. Here we recover the ability to conduct some metamathematical reasoning (which seemed impossible before due to the absence of negation), while at the same time preserving monotonicity by viewing intuitionistic logic as a "logic of proofs". Complexity is, again, controlled by restrictions to the induction rule.

We present only brief arguments in the main part of this article, due to space restrictions; full proofs and discussion of concepts can be found in the appendices.

2 Preliminaries

We formulate propositional logic (PL) with connectives $\bot, \top, \vee, \wedge, \supset$ and countably many propositional variables, e.g. p, q.[6] First-order logic (FOL) extends PL by eigenvariables, e.g. a, b, c, first-order variables, e.g. x, y, z, countably many predicate symbols R_i^k, for each arity $k \in \mathbb{N}$, and a symbol $=$ for equality. We also have first-order quantifiers $\exists$ and $\forall$.

In all settings, we denote formulae by variables A, B, C etc., possibly indicating free variables within parentheses, e.g. $A(x, y)$. A *positive* or *monotone* formula is one not containing $\supset$.

2.1 Bounded arithmetic and the Paris-Wilkie translation

We work in the language of Buss' theories S_2^i, T_2^i [7], which extends FOL by symbols S (successor), $+$, $\times$, $\leq$, $|\cdot|$ (length of binary representation), $\lfloor \frac{\cdot}{2} \rfloor$ (halve the argument and round down) and $\#$ (with $x \# y = 2^{|x| \cdot |y|}$).

[5] In fact, we work rather with systems called 'Mon' and 'Nor', which are polynomially equivalent to KS and KS^+ respectively, as shown in [14].

[6] We may use other common connectives symbols as abbreviations for their usual definitions in this basis. Namely, we have $\neg A := A \supset \bot$ and $\top := \neg\bot$.

We typically use symbols s, t to denote terms of the language. A term is *closed* if it is variable-free. $v(t)$ denotes the numerical value of a closed term t.

Due to the presence of #, we adopt quasipolynomial[7] time as our model of feasible computation.

We assume that the aforementioned predicate symbols R_i^k are present in all our theories,[8] and refer to them as *nonarithmetic* symbols.

We work in a standard two-sided sequent calculus for FOL, denoted LK, extended by an appropriate set $BASIC$ of quantifier-free and negation-free initial rules (i.e. axioms) which can essentially be found in [7].[9] Unless otherwise mentioned, we always assume $BASIC$ to be contained in any defined theories.

We augment our sequent calculus with rules for bounded quantifiers,

$$\exists\text{-l}\ \frac{\Gamma, a \leq t, A(a) \longrightarrow \Delta}{\Gamma, \exists x \leq t.A(x) \longrightarrow \Delta} \qquad \exists\text{-r}\ \frac{\Gamma \longrightarrow \Delta, A(t)}{\Gamma, t \leq s \longrightarrow \Delta, \exists x \leq s.A(x)}$$

$$\forall\text{-r}\ \frac{\Gamma, a \leq t \longrightarrow \Delta, A(a)}{\Gamma \longrightarrow \Delta, \forall x \leq t.A(x)} \qquad \forall\text{-l}\ \frac{\Gamma, A(t) \longrightarrow \Delta}{\Gamma, t \leq s, \forall x \leq s.A(x) \longrightarrow \Delta}$$

where the eigenvariable a occurs only as indicated.

For a class of formulae X the *induction* and *polynomial induction* rules for X-formulae A are defined as follows:

$$X\text{-}IND\ \frac{\Gamma, A(a) \longrightarrow A(a+1), \Delta}{\Gamma, A(0) \longrightarrow A(t), \Delta} \qquad X\text{-}PIND\ \frac{\Gamma, A(\lfloor \frac{a}{2} \rfloor) \longrightarrow A(a), \Delta}{\Gamma, A(0) \longrightarrow A(t), \Delta}$$

The following result is, in fact, a corollary of what is usually called 'free-cut elimination' introduced by Takeuti in [23], which is discussed in detail in [10]. The power of this result lies at the heart of the connections between bounded arithmetic and proof complexity.

Theorem 1 (Free-cut elimination). *Let X be a set of formulae closed under term substitution and under subformulas, containing the set of axioms BASIC. Then every X-IND- (or X-PIND-)proof can be transformed into an IND- (resp. PIND-)proof where all formulae occurring are in X.*

An important point is that the above statement also holds for *intuitionistic* theories (e.g. as stated in [10]), including those we introduce later on.

Definition 2 (Buss' theories). *The theories S_2^i and T_2^i are defined as Σ_i^b-PIND and Σ_i^b-IND respectively. S_2 and T_2 are defined as $\bigcup_i S_2^i$ and $\bigcup_i T_2^i$ respectively.*

---

[7] A quasipolynomial in n is a function $2^{\text{polylog}\, n}$.

[8] In fact it will suffice to assume the existence of a single nonarithmetic symbol R due to the access to primitive recursive codings in our theories.

[9] While the formulation referred to does include some occurrences of negation, these can be easily eliminated by simple arithmetic identities, e.g. as done in [8] and [12].

It is known that $S_2^i \subseteq T_2^i \subseteq S_2^{i+1}$, and so $S_2 = T_2$ [7]. Whether these inclusions are strict is an open problem.

Paris and Wilkie in [21] defined a translation $\langle \cdot \rangle$ of closed Δ_0-formulae to (quasi)polynomial-size propositional formulae, where nonarithmetic symbols are associated with propositional variables.

Definition 3 (PW for closed formulae). *We define a translation of closed formulae as follows,*

$$
\begin{array}{rcl}
\langle \ddagger \rangle & := & \ddagger \\
\langle R(t) \rangle & := & p_{v(t)} \\
\langle s \bowtie t \rangle & := & \begin{cases} \top & v(s) \bowtie v(t) \\ \bot & \text{otherwise} \end{cases}
\end{array}
\qquad
\begin{array}{rcl}
\langle A \star B \rangle & := & \langle A \rangle \star \langle B \rangle \\
\langle \exists x \leq t . A(x) \rangle & := & \bigvee_{k=0}^{v(t)} \langle A(k) \rangle \\
\langle \forall x \leq t . A(x) \rangle & := & \bigwedge_{k=0}^{v(t)} \langle A(k) \rangle
\end{array}
$$

for $\ddagger \in \{\bot, \top\}$, $\bowtie \in \{\leq, =\}$ *and* $\star \in \{\vee, \wedge, \supset\}$

Paris and Wilkie essentially proved the following:[10]

Theorem 4 (PW for proofs). *A* T_2*- (or* S_2*-) proof of a* Π_1*-sentence* $\forall \boldsymbol{x}. A(\boldsymbol{x})$ *can be translated to quasipolynomial-size bounded-depth LK-proofs of* $\langle A(\boldsymbol{n}) \rangle_{\boldsymbol{n} \in \mathbb{N}^{|\boldsymbol{x}|}}$.

2.2 Monotone and normal proofs

The setting we use is essentially due to Jeřábek in [18]: monotone proofs can be represented as term rewriting derivations in the following system,[11]

$$
\begin{array}{ll}
w\uparrow : A \to \top & c\uparrow : A \to A \wedge A \\
w\downarrow : \bot \to A & c\downarrow : A \vee A \to A
\end{array}
\qquad s : A \wedge (B \vee C) \to (A \wedge B) \vee C \tag{1}
$$

modulo associativity and commutativity of $\wedge$ and $\vee$ (denoted AC) and all equations for constants. A *normal* monotone proof is one where all $\uparrow$-steps occur before all $\downarrow$-steps.

We denote by Mon the rewriting system above and by Nor the set of all normal monotone proofs. For the sake of reducing prerequisites, we deal with Mon and Nor in this paper rather than explicitly defining the associated deep inference proof systems, KS^+ and KS respectively.

In what follows we give a 'deep inference' style presentation of rewriting derivations, first appearing in [17], in order to aid the analysis of normalisation complexity later on, e.g. in Sect. 4.2.

---

10 Paris and Wilkie in fact worked with slightly different theories, $I\Delta_0$ and $I\Delta_0 + \Omega_1$, but these only differ by the choice of base language.

11 Strictly speaking, this is not a term rewriting system (TRS) due to the fact that the LHS of $c\uparrow$ is just a variable, but the notions of reduction, derivation etc. otherwise remain the same as for a TRS.

Definition 5 (Derivations). *We define derivations, and premiss and conclusion functions,* pr*,* cn *respectively, inductively as follows:*

1. *Each formula A is a derivation with premiss and conclusion A.*
2. *If Φ, Ψ are derivations and $\star \in \{\wedge, \vee\}$ then $\Phi \star \Psi$ is a derivation with premiss $\mathsf{pr}(\Phi) \star \mathsf{pr}(\Psi)$ and conclusion $\mathsf{cn}(\Phi) \star \mathsf{cn}(\Psi)$.*
3. *If Φ, Ψ are derivations and $\mathsf{cn}(\Phi) \to \mathsf{pr}(\Psi)$ is an instance of a rule ρ then $\rho\,\dfrac{\Phi}{\Psi}$ is a derivation with premiss $\mathsf{pr}(\Phi)$ and conclusion $\mathsf{cn}(\Psi)$.*

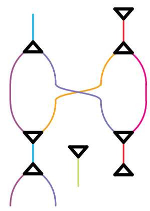
If Φ is a derivation where all inference steps are instances of rules in a system $\mathcal{S}$ with premiss A, conclusion B, we write $\begin{array}{c} A \\ \Phi \| \mathcal{S} \\ B \end{array}$.

Definition 6 (Atomic flows). *The* (atomic) flow *of a proof is the (directed) graph obtained by tracing the paths of all atoms, designating nodes when atoms are created, destroyed or duplicated.*

Atomic flows were first introduced in [16], and from a complexity point of view in [13]. They can be thought of as specialised versions of Buss' flow graphs [9].

We give an example of a MON-derivation and its associated flow, using colours to associate atom occurrences in the proof with edges in the flow:[12]

$$
s\,\dfrac{
  m\,\dfrac{ c{\uparrow}\dfrac{a}{a \wedge a} \vee c{\uparrow}\dfrac{w{\uparrow}\dfrac{\bot}{a}}{a \wedge a} }
  { =\dfrac{c{\downarrow}\dfrac{a \vee a}{a}}{a \vee w{\downarrow}\dfrac{\bot}{b}} \wedge c{\downarrow}\dfrac{a \vee a}{a} }
}{ c{\uparrow}\dfrac{a}{a \wedge a} \vee =\dfrac{b \wedge w{\uparrow}\dfrac{a}{\top}}{b} }
$$

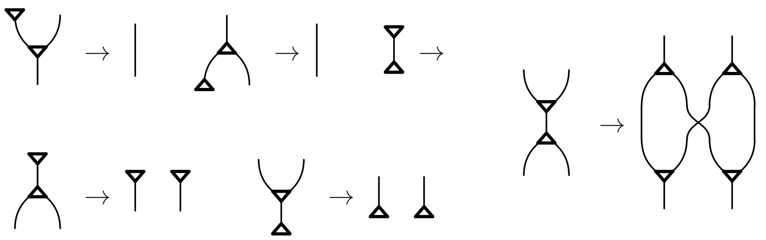


(2)

Using flows, normalisation of monotone proofs can be conducted in an entirely 'syntax-free' way, as shown in [13] and [14].

[12] In this derivation we also make use of the *medial* rule $m : (A \wedge B) \vee (C \wedge D) \to (A \vee C) \wedge (B \vee D)$ to show how $c{\uparrow}$ and $c{\downarrow}$ steps can be reduced to atomic form.

Definition 7. *We denote the following graph-rewriting system* norm*:*



(3)

These steps can be *lifted* to proofs: for a MON-proof π, manipulating its flow via norm results in the flow of a MON-proof with the same premiss and conclusion as π [16]. The proof of this relies somewhat crucially on an extension of MON by the *medial* rule [4], as well as basic manipulations with constants.

norm is *terminating* and *confluent*, and its normal forms are just the flows of NOR-proofs [16] [13]. In particular we have the following results from [13]:

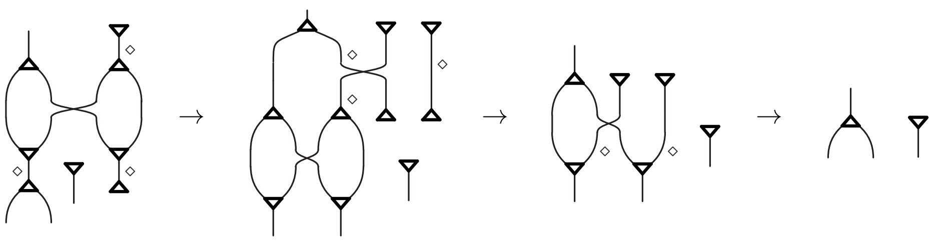
Theorem 8. norm *is weakly normalising in time polynomial in the number of paths of an input flow.*

Corollary 9. norm *induces a normalisation procedure on* MON *of time complexity polynomial in the number of paths in the flow of an input derivation.*

These results can sometimes considerably simplify the calculation of bounds on the complexity of normalisation. In this work it will suffice to estimate the number of paths by the following very simple upper bound:

Fact 10 *The number of paths in a flow of length l is $\leq 2^l$.*

Example 11. We give a norm-derivation that normalises the MON-proof in (2):[13]



Notice that we could have made different choices of redex at each step, but that we would have arrived at the same final flow by confluence. We also have that any flow appearing in the derivation can be lifted to a MON-proof of the same premiss and conclusion as we started with (a and $(a \wedge a) \vee b$ respectively).

[13] Here we have marked redexes (with unique applicable reduction step) by $\diamond$.

3 Monotone inductive definitions

Recall that our motivation is to design theories for deep inference (and monotone) systems. A drawback of the theories S_2 and T_2 is that they can fundamentally only reason about bounded-depth formulae, due to the fixed quantifier alternation appearing in an arithmetic proof. This is often dealt with in bounded arithmetic by incorporating forms of the *comprehension* axiom (e.g. in [20] and [11]), however such an approach is not appropriate in our setting due to the apparent introduction of negated variables in the propositional translation.

Rather, we define a positive *least fixed point* operator whose inductive definitions have closure functions[14] bounded by a polylogarithm in its arguments. The precise implementation of such an operator is somewhat orthogonal to our proof theoretic focus, so here we give a simple definition of such fixed points using explicit bounds rather than a more 'implicit' approach, e.g. as is common in descriptive complexity. However we do consider this pursuit an important issue which is thus deferred to future research.

Definition 12 (Polynomial terms). *A term is* polynomial *if it is #-free. We denote polynomial terms by the (possibly decorated) symbol r.*

Definition 13 (LFP). *LFP^{polylog} is the set of (double) sequents of the form,*

$$R_A(\boldsymbol{x}) \longleftrightarrow A(R_A, \boldsymbol{x}) \tag{4}$$

for each $A(R, \boldsymbol{x})$ that has the form,

$$Q_1 y_1 \leq t_1 \dots Q_d y_d \leq t_d . \left(B(R(\boldsymbol{t}^1), \dots, R(\boldsymbol{t}^k), \boldsymbol{x}) \wedge \bigwedge_{i=1}^{k} r(|\boldsymbol{t}^i|) < r(|\boldsymbol{x}|) \right)$$

for some bounded quantifiers $Q_i x_i \leq t_i$, B positive and quantifier-free, r a polynomial and with all occurrences of R indicated.

The $r(|\cdot|)$ term in B above acts as a 'clock', ensuring that evaluation of the inductive definition terminates in polylogarithmic-time. Consequently, under the Paris-Wilkie translation, these unwind to AC-classes of monotone circuits:

Proposition 14. *The translation $\langle \cdot \rangle$ can be extended to each predicate symbol R_A with a defining axiom of format (4), such that $\langle R_A(\boldsymbol{n}) \rangle \longleftrightarrow \langle A(R_A, \boldsymbol{n}) \rangle$ has* NOR*-derivations of size quasipolynomial in $\boldsymbol{n} \in \mathbb{N}^{|x|}$.*[15]

Remark 15. We point out that one could also define LFP^{poly}, with polynomial-time closure functions, which would allow access to monotone circuits of polynomial depth, and so corresponds to monotone proofs with extension.[16]

[14] This is the number of times the inductive definition needs to be iterated before reaching a fixed point.

[15] In fact, the propositional formulae induced have quasipolynomial size and polylogarithmic depth, but the derivations themselves are rather simple.

[16] However, such a pursuit is probably not well-motivated, in light of the polynomial simulation of extended Hilbert-Frege systems by extended MLK given in [19].

4 Monotone theories and propositional translations

Let Δ_0^+ be the class of *positive* Δ_0-formulae.

Definition 16 (Monotone theories). *We define the theories MS_2 and MT_2 as Δ_0^+-PIND and Δ_0^+-IND respectively.*

Notice that the usual argument showing that $T_2^i \subseteq S_2^{i+1}$ cannot be carried out in the monotone setting, due to the introduction of $\supset$ symbols in the argument (e.g. given in [10]), and so it is not clear whether or not $MS_2 = MT_2$.

4.1 The Paris-Wilkie translation for $MT_2 + LFP^{\text{polylog}}$

We survey the Paris-Wilkie translation briefly for our monotone theories, focusing on the induction rule and its flows to obtain complexity bounds.

A minor contribution to the literature here is our presentation of the Paris-Wilkie translation in 'deep-inference' style. That this is possible at all is largely due to insights from previous works, e.g. [3] [18] [14].

Definition 17 (PW for sequents). *We extend $\langle\cdot\rangle$ to the LHS and RHS of sequents. $\langle\Gamma\rangle_l$ is defined as $\bigwedge_{A\in\Gamma}\langle A\rangle$ and $\langle\Gamma\rangle_r$ is defined as $\bigvee_{A\in\Gamma}\langle A\rangle$.*

We translate an $MT_2+LFP^{\text{polylog}}$ proof $\pi(\boldsymbol{x})$ of a Δ_0^+ sequent $A(\boldsymbol{x}) \longrightarrow B(\boldsymbol{x})$ to derivations $\begin{array}{c}\langle A(\boldsymbol{n})\rangle\\ {\scriptstyle\langle\pi(\boldsymbol{n})\rangle}\Vert{\scriptstyle \text{MON}}\\ \langle B(\boldsymbol{n})\rangle\end{array}$ for each $\boldsymbol{n} \in \mathbb{N}^{|\boldsymbol{x}|}$.

For brevity, we give only the case of the induction step below. The other cases increase the length of flows by at most a constant factor.[17]

A proof $\pi(a)$ extended by an induction step,

$$\mathit{IND}\ \frac{\Gamma, A(a) \longrightarrow A(Sa), \Delta}{\Gamma, A(0) \longrightarrow A(t), \Delta}$$

is translated to the derivation in Fig. 4.1:

A usual complexity analysis of PW-style translations gives us the following:

Theorem 18. *For an $MT_2+LFP^{\text{polylog}}$ proof $\pi(\boldsymbol{x})$ of a positive sequent $A(\boldsymbol{x}) \longrightarrow B(\boldsymbol{x})$, the derivations $\begin{array}{c}\langle A(\boldsymbol{n})\rangle\\ {\scriptstyle\langle\pi(\boldsymbol{n})\rangle}\Vert{\scriptstyle \text{MON}}\\ \langle B(\boldsymbol{n})\rangle\end{array}$ have size quasipolynomial in $\boldsymbol{n} \in \mathbb{N}^{|\boldsymbol{x}|}$.*

[17] These other cases can be constructed from the usual translations, e.g. in [20] [11], by consulting e.g. [18] or [3], taking care to introduce only necessary structural steps.

$$
\begin{array}{rc}
 & c\uparrow \dfrac{\langle\Gamma\rangle_l}{\langle\Gamma\rangle_l \wedge \langle\Gamma\rangle_l} \wedge \langle A(0)\rangle \\
\cline{2-2}
= & \langle\Gamma\rangle_l \wedge \langle A(0)\rangle \\
 & c\uparrow \dfrac{\langle\Gamma\rangle_l}{\langle\Gamma\rangle_l \wedge \langle\Gamma\rangle_l} \wedge \quad \langle\pi(0)\rangle \Big\Vert \\
 & \langle A(1)\rangle \vee \langle\Delta\rangle_r \\
\cline{2-2}
s & c\uparrow \dfrac{\langle\Gamma\rangle_l}{\langle\Gamma\rangle_l \wedge \langle\Gamma\rangle_l} \wedge \left[\begin{array}{l} \langle\Gamma\rangle_l \wedge \langle A(1)\rangle \\ \langle\pi(1)\rangle \Big\Vert \qquad\qquad \vee \langle\Delta\rangle_r \\ \langle A(2)\rangle \vee \langle\Delta\rangle_r \end{array}\right] \\
\cline{2-2}
s & \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
\cline{2-2}
s & \langle\Gamma\rangle_l \wedge \left[\begin{array}{l} \langle\Gamma\rangle_l \wedge \langle A(t-2)\rangle \\ \langle\pi(t-2)\rangle \Big\Vert \qquad\qquad \vee\, c\downarrow \dfrac{\langle\Delta\rangle_r \vee \langle\Delta\rangle_r}{\langle\Delta\rangle_r} \\ \langle A(t-1)\rangle \vee \langle\Delta\rangle_r \end{array}\right] \\
\cline{2-2}
s & \begin{array}{l} \langle\Gamma\rangle_l \wedge \langle A(t-1)\rangle \\ \langle\pi(t-1)\rangle \Big\Vert \qquad\qquad \vee\, c\downarrow \dfrac{\langle\Delta\rangle_r \vee \langle\Delta\rangle_r}{\langle\Delta\rangle_r} \\ \langle A(t)\rangle \vee \langle\Delta\rangle_r \end{array} \\
\cline{2-2}
= & \langle A(t)\rangle \vee c\downarrow \dfrac{\langle\Delta\rangle_r \vee \langle\Delta\rangle_r}{\langle\Delta\rangle_r}
\end{array}
$$

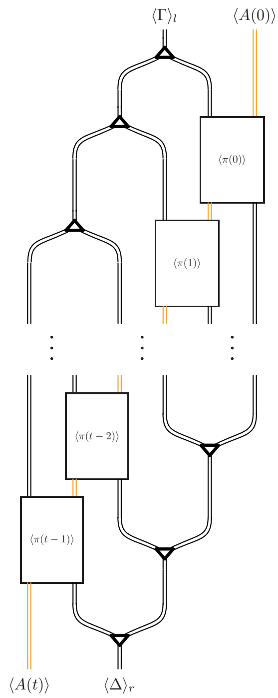


Fig. 1. Translation of a derivation π extended by an induction step. We give the corresponding flow on the right. Thick 'double' edges are used to signify that there are many edges in parallel, one for each distinct atom occurrence, and we use the colour orange in both the derivation and the flow to track instances of the induction formula in the propositional translation.

4.2 $\langle\cdot\rangle$ on $\boldsymbol{MS_2} + \boldsymbol{LFP}^{\mathbf{polylog}}$ normalises in quasipolynomial time

Notice that $\langle\cdot\rangle$ can be adapted for *PIND*-steps similarly to the case for *IND* above. The difference is that there will be only *logarithmically* many boxes $\langle\pi(\cdot)\rangle$ in the flow, due to the divide-and-conquer format of the induction. Consequently the size of flow increases by only a logarithmic factor in the largest term value.

Lemma 19. *$\langle\cdot\rangle$ on MS_2+LFP^{polylog} proofs induces polylogarithmic length flows.*

We can now apply our normalisation result, Thm. 8, to obtain the following:

Theorem 20. *For an MS_2+LFP^{polylog} proof $\pi(\boldsymbol{x})$ of a positive sequent $A(\boldsymbol{x}) \longrightarrow B(\boldsymbol{x})$, there are derivations* $\begin{array}{c}\langle A(\boldsymbol{n})\rangle\\ \|_{\text{NOR}}\\ \langle B(\boldsymbol{n})\rangle\end{array}$ *of size quasipolynomial in $\boldsymbol{n} \in \mathbb{N}^{|\boldsymbol{x}|}$.*

We point out that certain arguments appearing in e.g. [13] and [14] can be seen as special cases of the above result.

5 An intuitionistic hierarchy and generalised Paris-Wilkie

Unfortunately, the monotone setting in arithmetic does not allow us to readily conduct metamathematical reasoning, and so it seems difficult (perhaps impossible) to prove soundness results (or 'reflection' principles) within MS_2 and MT_2.

Therefore we introduce an *intuitionistic* hierarchy of theories in which to conduct metamathematical reasoning. The idea here is to view intuitionistic logic as a "logic of proofs". In this way we can extend the PW-translation to deal with implication without breaking monotonicity.[18]

Previous work on intuitionistic bounded arithmetic has included only positive induction [8] [12], in order to conduct 'realisability' arguments. In those settings one can, in fact, simulate the full power of non-positive induction, but it is not possible here due to the presence of nonarithmetic predicate symbols.

5.1 The $\boldsymbol{I_j S_2}(+\boldsymbol{LFP}^{\mathbf{polylog}})$ hierarchy and a propositional translation

Sequent calculi for intuitionistic theories coincide with usual ones, except with the proviso that at most one formula occurs on the right of a sequent.[19]

The *left implication depth* of a formula A is the largest number of times a path chooses the left branch of a $\supset$ symbol in the formula tree of A. Let $\Delta_0^{I_j}$ denote the class of Δ_0 formulae whose left implication depth is at most j.

Definition 21. *Define the intuitionistic theories I_jS_2 and I_jT_2 as $\Delta_0^{I_j}$-PIND and $\Delta_0^{I_j}$-IND respectively.*

[18] Such an approach could not be conducted in classical logic due to De Morgan laws, which would induce a collapse to the first level.

[19] In particular this condition is required for the induction rules too, as in [8] and [12].

Remark 22. When restricted to monotone implications, we have that $MT_2 \subseteq I_1S_2$ (with or without LFP^{polylog}): right-contraction and weakening steps on positive formulae can be easily simulated, e.g. as in [2], and $\forall$-r and $\exists$-l rules can be simulated by $\Delta_0^{I_1}$-*PIND*.

Interestingly, I_1S_2+LFP^{polylog} proofs also translate to small Mon-derivations, and this is one of the main results of the present work:

Theorem 23. *For an I_1S_2+LFP^{polylog} proof $\pi(\boldsymbol{x})$ of a positive sequent $A(\boldsymbol{x}) \longrightarrow B(\boldsymbol{x})$, there are derivations* $\begin{array}{c}\langle A(\boldsymbol{n})\rangle \\ \|\textsc{Mon} \\ \langle B(\boldsymbol{n})\rangle\end{array}$ *of size quasipolynomial in $\boldsymbol{n} \in \mathbb{N}^{|\boldsymbol{x}|}$.*

We defer a full proof to the appendices, but present here some of the underlying ideas.

A proof π of I_1S_2 of a $\Delta_0^{I_1}$ sequent,

$$A_1(\boldsymbol{x}) \supset B_1(\boldsymbol{x}), \dots, A_k(\boldsymbol{x}) \supset B_k(\boldsymbol{x}) \longrightarrow A(\boldsymbol{x}) \supset B(\boldsymbol{x})$$

is translated to a quasipolynomial-time *transformation*,

$$\langle \pi \rangle \quad : \quad \begin{pmatrix} A_1(\boldsymbol{n}) & & A_k(\boldsymbol{n}) \\ {\scriptstyle \Phi_1}\| & ,\dots, & {\scriptstyle \Phi_k}\| \\ B_1(\boldsymbol{n}) & & B_k(\boldsymbol{n}) \end{pmatrix} \mapsto \begin{array}{c} A(\boldsymbol{n}) \\ {\scriptstyle \Phi}\| \\ B(\boldsymbol{n}) \end{array}$$

for arbitrary monotone derivations Φ_i of the given format.

Logical combinations of such implication formulae are interpreted in the natural way by our generalised translation .

Complexity is generated by left-contraction steps, which correspond to dag-like behaviour in an *MLK*-proof, e.g. the extension π' of a proof π by a step,

$$\frac{A \supset B, A \supset B \longrightarrow C \supset D}{A \supset B \longrightarrow C \supset D}$$

is translated to $\langle \pi' \rangle$ where $\langle \pi' \rangle (\Phi) := \langle \pi \rangle (\Phi, \Phi)$, for $\Phi : A \xrightarrow[\textsc{Mon}]{*} B$.

However *PIND*, again, ensures that the graphs of these proofs have only polylogarithmic height, and so can be transformed in quasipolynomial-time to tree-like *MLK*, i.e. Mon-derivations.

6 Reflection principles

The *reflection* principle for a propositional proof system (PPS) is a formal statement of its soundness. Proofs in arithmetic theories of such principles serve as converses of propositional translations (e.g. Paris-Wilkie), since they guarantee that the theory is amongst the strongest for which such a translation could exist.

For brevity, we do not present the full formalisation here, but assume we have access to the following Δ_0^+-formulae, defined using LFP^{polylog}:

- $Fla(x)$:= "x codes some positive formula over propositional variables p_i".
- $Der_P(x, y, z) := Fla(y) \wedge Fla(z) \wedge$ "x codes a P-derivation from y to z".
- $Tr_R(x) := Fla(x) \wedge$ "the formula coded by x is true when $p_i \equiv R(i)$"

Standard similar definitions might be found in various textbooks, e.g. [20] [11], however we point out that it is critical here that we exploit the presence of least fixed points in order to reason about formulae of unbounded depth.[20]

Definition 24 (Reflection). *For a PPS P, we define the following:*

$$Rfn_P \quad := \quad \forall x, y, z \;.\; Der_P(x, y, z) \supset (Tr_R(y) \supset Tr_R(z)) \tag{5}$$

Proposition 25 (Quasipolynomial simulation). *We have the following:*

1. *If $MS_2 + LFP^{\mathrm{polylog}}$ proves Rfn_P then* NOR *quasipolynomially simulates P.*
2. *If $MT_2 + LFP^{\mathrm{polylog}}$ proves Rfn_P then* MON *quasipolynomially simulates P.*
3. *If $I_1S_2 + LFP^{\mathrm{polylog}}$ proves Rfn_P then* MON *quasipolynomially simulates P.*

A subtlety that allows the above results to go through is that $Der_P(x, y, z)$ is R-free and so is translated to Boolean combinations of constants, which can be evaluated even in our monontone theories.

Finally we give the following result which, in light of Prop. 25 above, provides a form of converse to Thm. 23.

Theorem 26. *$I_1S_2 + LFP^{\mathrm{polylog}}$ proves Rfn_{MON}.*

Proof (Idea). Notice that, if the $\forall x, y, z$ in (5) is bounded by a parameter w, we have a $\Delta_0^{I_1}$-formula in w, for which we can perform a polynomial induction on w, applying the inductive hypothesis to the first half and second half of a MON rewriting derivation, thence applying cuts to derive the inductive step.

Remark 27. As perhaps expected from the above, also $I_1T_2 + LFP^{\mathrm{polylog}}$ corresponds to dag-like MLK, in the same way as I_1S_2 to MON and tree-like MLK.

7 Further work and conclusions

We gave uniform versions of monotone and analytic deep inference proof systems, in the setting of bounded arithmetic. This constituted an application of least fixed points, deep inference proof normalisation and intuitionistic bounded arithmetic to propositional proof complexity. In the case of monotone proofs we were able to also prove a converse result.

We do not yet have a full correspondence for NOR, and this reflects the difficulty in deep inference proof complexity of conducting any metamathematical reasoning at all in KS. One approach might be to incorporate structural restrictions from *linear logic*, e.g. that induction formulae must be exponential-free.

[20] It is also important at this stage to be careful about the use of negation for nonarithmetic symbols.

While this might not be desirable from the point of view of reasoning, it might allow us to conduct one-off proofs of soundness of various systems, thereby settling certain proof complexity questions.

A more general question is that of the relative strength of our theories. We believe that the (nonuniform) monotone simulation of Hilbert-Frege proofs given in [1] can be made uniform in $I_1S_2 + LFP^{\mathrm{polylog}}$,[21] and so the hierarchy collapses to this level. Hence there are two natural questions arising:

1. Are the inclusions $MS_2+LFP^{\mathrm{polylog}} \subseteq MT_2+LFP^{\mathrm{polylog}} \subseteq I_1S_2+LFP^{\mathrm{polylog}}$ strict?
2. Can a simpler purely logical proof be given of the collapse of the intuitionistic hierarchy, perhaps to some $I_kS_2 + LFP^{\mathrm{polylog}}$ for some large, but finite k?

A positive answer to 2 would give a simple proof of the quasipolynomial-simulation of Hilbert-Frege over monotone sequents by some 'super-dag-like' monotone system, or equivalently propositional intuitionistic Hilbert-Frege systems of bounded implication depth.

As previously mentioned, in the absence of fixed points and negation, it is no longer clear that $MT_2^i \subseteq MS_2^{i+1}$ (contrary to the nonmonotone setting). In general it might be pertinent to study these theories to gain results on bounded-depth monotone proof complexity.[22][23]

This work further brings research in deep inference in line with the standards of mainstream proof complexity. By studying restrictions of monotone systems we also somewhat contribute to 'bridging the gap' between weak systems and Hilbert-Frege systems (for which no nontrivial lower bounds are known), providing finer granularity of various subproblems.

References

1. A. Atserias, N. Galesi, and P. Pudlák. Monotone simulations of non-monotone proofs. *Journal of Computer and System Sciences*, 65(4):626–638, 2002.
2. M. Bılková. Monotone sequent calculus and resolution. *Comment. Math. Univ. Carolin*, 42(3):575–582, 2001.
3. K. Brünnler. Deep inference and its normal form of derivations. In B. L. J. V. T. Arnold Beckmann, Ulrich Berger, editor, *Logical Approaches to Computational Barriers, Second Conference on Computability in Europe*, volume 3988 of *Lecture Notes in Computer Science*, pages 65–74. Springer-Verlag, 2006.
4. K. Brünnler and A. F. Tiu. A local system for classical logic. In R. Nieuwenhuis and A. Voronkov, editors, *LPAR 2001*, volume 2250 of *Lecture Notes in Computer Science*, pages 347–361. Springer-Verlag, 2001.
5. P. Bruscoli and A. Guglielmi. On analyticity in deep inference. `http://cs.bath.ac.uk/ag/p/ADI.pdf`, 2009.

[21] In fact we also have the outline of a proof but it is omitted for space considerations.
[22] Such results have already been presented in e.g. [1].
[23] We point out that our normalisation procedure (norm) does not seem to preserve bounded-depth-ness, but the concept of normality is itself strong enough to retain completeness when depth is bounded.

6. P. Bruscoli and A. Guglielmi. On the proof complexity of deep inference. *ACM Transactions on Computational Logic*, 10(2):1–34, 2009.
7. S. Buss. *Bounded arithmetic*. Bibliopolis Naples, 1986.
8. S. R. Buss. The polynomial hierarchy and intuitionistic bounded arithmetic. 1985.
9. S. R. Buss. The undecidability of k-provability. *Annals of Pure and Applied Logic*, 53(1):75–102, 1991.
10. S. R. Buss. *Handbook of proof theory*. Elsevier, 1998.
11. S. Cook and P. Nguyen. *Logical Foundations of Proof Complexity*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.
12. S. Cook and A. Urquhart. Functional interpretations of feasibly constructive arithmetic. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 107–112. ACM, 1989.
13. A. Das. Complexity of deep inference via atomic flows. In S. B. Cooper, A. Dawar, and B. Löwe, editors, *Computability in Europe*, volume 7318 of *Lecture Notes in Computer Science*, pages 139–150. Springer-Verlag, 2012.
14. A. Das. On the pigeonhole and related principles in deep inference and monotone systems. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, CSL-LICS '14, pages 36:1–36:10, New York, NY, USA, 2014. ACM.
15. A. Das. Bounded arithmetic for deep inference and monotone systems, 2015. `http://www.anupamdas.com/bdd-arith-di-full.pdf`.
16. A. Guglielmi and T. Gundersen. Normalisation control in deep inference via atomic flows. *Logical Methods in Computer Science*, 4(1:9):1–36, 2008.
17. A. Guglielmi, T. Gundersen, and M. Parigot. A proof calculus which reduces syntactic bureaucracy. 2010.
18. E. Jeřábek. Proof complexity of the cut-free calculus of structures. *Journal of Logic and Computation*, 19(2):323–339, 2009.
19. E. Jeřábek. Proofs with monotone cuts. Preprint, 2010.
20. J. Krajíček. *Bounded arithmetic, propositional logic, and complexity theory*. Cambridge University Press, New York, NY, USA, 1995.
21. J. Paris and A. Wilkie. δ_0 sets and induction. *Open Days in Model Theory and Set Theory, W. Guzicki, W. Marek, A. Pelc, and C. Rauszer, eds*, pages 237–248, 1981.
22. P. Pudlák. On the complexity of the propositional calculus. *London Mathematical Society Lecture Note Series*, pages 197–218, 1999.
23. G. Takeuti. Proof theory. volume 81 of studies in logic and the foundations of mathematics, 1975.

A Appendix for Sect. 2

A.1 The sequent calculus

We present the sequent calculus with additive formulations of the left-conjunction and right-disjunction rules, and multiplicative versions of all branching rules. The former is so that we can readily adapt the calculus to be sound for intuitionistic logic later on, via the usual restriction of one formula on the right, while the latter serves to tame structural behaviour arising from sequent rules, in order to control the atomic flows extracted from proofs.[24]

We consider cedents as multisets of formulae, and so do not include 'exchange' rules for management of the underlying data structure.

Definition 28 (Sequent calculus). *We define the calculus LK as follows. There are the following initial (i.e. 0-ary) rules, or* axioms*:*

$$\top \frac{}{\longrightarrow \top} \qquad \bot \frac{}{\bot \longrightarrow} \qquad id \frac{}{A \longrightarrow A}$$

There are the following structural *rules:*

$$w\text{-}l \frac{\Gamma \longrightarrow \Delta}{\Gamma, A \longrightarrow \Delta} \qquad w\text{-}r \frac{\Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, A} \qquad c\text{-}l \frac{\Gamma, A, A \longrightarrow \Delta}{\Gamma, A \longrightarrow \Delta} \qquad c\text{-}r \frac{\Gamma \longrightarrow \Delta, A, A}{\Gamma \longrightarrow \Delta, A}$$

There are the following (propositional) logical *rules,*

$$(\vee) \qquad \vee\text{-}l \frac{\Gamma, A \longrightarrow \Delta \quad \Sigma, B \longrightarrow \Delta}{\Gamma, \Sigma, A \vee B \longrightarrow \Delta, \Pi} \qquad \vee\text{-}r \frac{\Gamma \longrightarrow \Delta, A_i}{\Gamma \longrightarrow \Delta, A_1 \vee A_2}$$

$$(\wedge) \qquad \wedge\text{-}l \frac{\Gamma, A_i \longrightarrow \Delta}{\Gamma, A_1 \wedge A_2 \longrightarrow \Delta} \qquad \wedge\text{-}r \frac{\Gamma \longrightarrow \Delta, A \quad \Sigma \longrightarrow \Pi, B}{\Gamma, \Sigma \longrightarrow \Delta, \Pi, A \wedge B}$$

$$(\supset) \qquad \supset\text{-}l \frac{\Gamma \longrightarrow \Delta, A \quad \Sigma, B \longrightarrow \Pi}{\Gamma, \Sigma, A \supset B \longrightarrow \Delta, \Pi} \qquad \supset\text{-}r \frac{\Gamma, A \longrightarrow \Delta, B}{\Gamma \longrightarrow \Delta, A \supset B}$$

for $i \in \{1, 2\}$.

There are the following quantifier *rules,*

$$(\exists) \qquad \exists\text{-}l \frac{\Gamma, A(a) \longrightarrow \Delta}{\Gamma, \exists x.A(x) \longrightarrow \Delta} \qquad \exists\text{-}r \frac{\Gamma \longrightarrow \Delta, A(t)}{\Gamma \longrightarrow \Delta, \exists x.A(x)}$$

$$(\forall) \qquad \forall\text{-}l \frac{\Gamma, A(t) \longrightarrow \Delta}{\Gamma, \forall x.A(x) \longrightarrow \Delta} \qquad \forall\text{-}r \frac{\Gamma \longrightarrow \Delta, A(a)}{\Gamma \longrightarrow \Delta, \forall x.A(x)}$$

[24] In fact, from this point of view, it would be even more pertinent to consider multiplicative versions of all rules, but that would be at the expense of obtaining LJ as a fragment of the calculus.

where the eigenvariable a occurs only as indicated.

Finally, there is the cut rule:

$$\frac{\Gamma \longrightarrow \Delta, A \quad \Sigma, A \longrightarrow \Pi}{\Gamma, \Sigma \longrightarrow \Delta, \Pi}\ cut$$

We may refer to *LK* for both the version defined above or just its propositional fragment.[25] The calculus *MLK* is obtained from the propositional fragment by removing the $\supset$ -l and $\supset$ -r rules.

A.2 Some background on bounded arithmetic

There are some missing preliminaries for Dfn. 2, in that we have not yet said what a Δ_i^b-formula is. For the purposes of this paper however, where we do not distinguish between S_2^i for varying i, it will be simpler to give a self-contained definition of S_2 and T_2 rather than presenting the (bounded) arithmetical hierarchy, especially considering that there is no such canonical hierarchy in the intuitionistic setting.

Definition 29 (Bounded formulae and Buss' theories). *A quantifer is* bounded *if it has the format $\forall x \leq t$ or $\exists x \leq t$ for some term t. The semantics of these quantifiers is determined by the bounded quantifier rules presented in Sect. 2.1 but, for simplicity, they can be interpreted as follows:*

$$\exists x \leq t.A \ \equiv\ \exists x.(x \leq t \wedge A) \qquad \forall x \leq t.A \ \equiv\ \forall x.(x \leq t \supset A)$$

Δ_0 is the set of bounded formulae, i.e. those whose quantifiers are all bounded. The theories S_2 and T_2 are defined as Δ_0-PIND and Δ_0-IND respectively.

For completeness we give a monotone version of Buss' axiom set *BASIC* below. This is essentially taken from [12], where its equivalence to the usual set of axioms is also stated.

Definition 30. *We define the system BASIC as the following set of initial sequents,*

1. $a = Sa \longrightarrow 0 = 1$
2. $\longrightarrow 0 \leq a$
3. $a \leq b \longrightarrow (a = b \vee Sa \leq b)$
4. $(a \leq b \wedge b \leq c) \longrightarrow a \leq c$
5. $(a \leq b \wedge b \leq a) \longrightarrow a = b$
6. $\longrightarrow a \leq b \vee b \leq a$
7. $\longrightarrow |0| = 0$
8. $S0 \leq a \longrightarrow |2a| = S|a|$
9. $\longrightarrow |S(2a)| = S|a|$
10. $a \leq b \longrightarrow |a| \leq |b|$

[25] This is sometimes also called '*PK*' in the literature, e.g. in [10] and [11].

11. $\longrightarrow |a\#b| = S(|a| \cdot |b|)$
12. $\longrightarrow 1\#1 = 2$
13. $\longrightarrow a\#b = b\#a$
14. $|a| = |c| + |d| \longrightarrow a\#b = (c\#b) \cdot (d\#b)$
15. $\longrightarrow a + 0 = a$
16. $\longrightarrow a + Sb = S(a + b)$
17. $\longrightarrow (a + b) + c = a + (b + c)$
18. (a) $a + b \leq a + c \longrightarrow b \leq c$
    (b) $b \leq c \longrightarrow a + b \leq a + c$
19. $\longrightarrow a \cdot 1 = a$
20. $\longrightarrow a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
21. $\longrightarrow a = \lfloor \frac{a}{2} \rfloor + \lfloor \frac{a}{2} \rfloor \vee a = S(\lfloor \frac{a}{2} \rfloor + \lfloor \frac{a}{2} \rfloor)$

where $1 := S0$, $2 := S1$ *and where the symbol* $\times$ *is frequently replaced by* $\cdot$ *or omitted altogether for presentation reasons.*

A.3 Equational theory for Mon and Nor

Regarding the statement in Sect. 2.2 that rewriting for Mon is conducted modulo AC and equations for the constants, we give a complete implementation of this theory below:

Commutativity	*Associativity*	*Constants*
$A \vee B = B \vee A$	$[A \vee B] \vee C = A \vee [B \vee C]$	$A \vee \bot = A$
$A \wedge B = B \wedge A$	$(A \wedge B) \wedge C = A \wedge (B \wedge C)$	$A \wedge \top = A$
		$\top \vee \top = \top$
		$\bot \wedge \bot = \bot$

A.4 Deep inference systems

We present briefly the deep inference systems related to Mon and Nor, which we did not have time to present in the main part of this paper. This discussion is essentially based on that appearing in previous work, namely [14].

The deep inference systems KS^+ and KS are obtained from Mon and Nor respectively by adding the 'identity' rule:

$$i\!\downarrow \frac{\top}{A \vee \bar{A}}$$

In this setting we no longer have an implication symbol $\supset$, but instead allow negation only on variables. The expression '$\bar{A}$' denotes the *negation normal form* of the negation of A, obtained by the following De Morgan laws,

$$\bar{\top} = \bot \quad \bar{\bot} = \top \quad \bar{\bar{p}} = p \quad \overline{A \vee B} = \bar{A} \wedge \bar{B} \quad \overline{A \wedge B} = \bar{A} \vee \bar{B}$$

so that $\bar{A}$ contains only negation on propositional variables (if at all).

A formula is now positive or monotone if it contains no negated variables.

A *proof* in a deep inference system is a derivation whose premiss is $\top$.

Theorem 31 (Conservativity). *A* KS^+ *(or* KS*) derivation with monotone premiss and conclusion can be polynomially transformed to a* Mon *derivation (resp.* Nor*) derivation with the same premiss and conclusion.*

What is not clear, however is whether a KS^+ (or KS) *proof* of a monotone implication $A \supset B$, i.e. $\bar{A} \vee B$, can be polynomially transformed into a Mon (resp. Nor) derivation from A to B. This is in fact related to certain *interpolation* hypotheses for analytic deep inference.

A.5 Atomic flows and normalisation

We refer the reader to [14] and [13] for a background on atomic flows and the complexity of normalisation, and also to [16] for a comprehensive introduction to atomic flows in general.

B Appendix for Sect. 3

B.1 Paris-Wilkie translation of inductive definitions

We give the translation of inductive definitions below.

For an inductive definition,

$$R_A(\boldsymbol{x}) \longleftrightarrow A(R_A, \boldsymbol{x})$$

where $A(R, \boldsymbol{x})$ is,

$$\exists y_1 \leq s_1 . \forall z_1 \leq t_1 . \ \cdots \ . \exists y_d \leq s_d . \forall z_d \leq t_d . \begin{pmatrix} & B(R(\boldsymbol{t}^1), \dots, R(\boldsymbol{t}^k), \boldsymbol{x}) \\ \wedge & \bigwedge_{i=1}^{k} r(|\boldsymbol{t}^i|) < r(|\boldsymbol{x}|) \end{pmatrix}$$

for B positive and quantifier-free, we define $\langle R_A(\boldsymbol{n}) \rangle$ by induction on $r(\boldsymbol{n})$ as follows:

$$\bigvee_{i_1=0}^{v(s_1)} \bigwedge_{j_1=0}^{v(t_1)} \cdots \bigvee_{i_d=0}^{v(s_d)} \bigwedge_{j_d=0}^{v(t_d)} \bigvee_{r(|\boldsymbol{m}^i|) < r(|\boldsymbol{n}|)} \begin{pmatrix} & \left\langle B\left(\left\langle R_A(\boldsymbol{m}^1) \right\rangle, \dots, \left\langle R_A(\boldsymbol{m}^k) \right\rangle, \boldsymbol{n} \right) \right\rangle \\ \wedge & \bigwedge_{i=1}^{k} \left\langle \boldsymbol{t}^i[\boldsymbol{n}/\boldsymbol{x}] = \boldsymbol{m}^i \right\rangle \end{pmatrix}$$

It is not difficult to see that initial sequents from LFP^{polylog} are translated to simple monotone proofs of the appropriate format, consisting only of basic manipulations with the constant symbols.

B.2 Least fixed points in the intuitionistic setting

We cannot in general achieve a prenex normal form for all formulae in plain intuitionistic logic, and in particular not even monotone formulae due to the restriction on the ∀-r rule. However, it will suffice to work with monotone formulae already in prenex normal form for this work, although we point out that we could have been more careful about our clock in a way that did not rely on prenexing, e.g. by insisting that each $r(|\boldsymbol{t}^i|)$ occurs directly in conjunction with the associated occurrence of $R(\boldsymbol{t}^i)$.

Nonetheless, we in fact show that the ∀-r rule for positive formulae can be derived already in I_1S_2 in Sect. D when we prove that $MT_2 \subseteq I_1S_2$, and so that it is fine to assume that monotone formulae are in prenex normal form.

C Appendix for Sect. 4

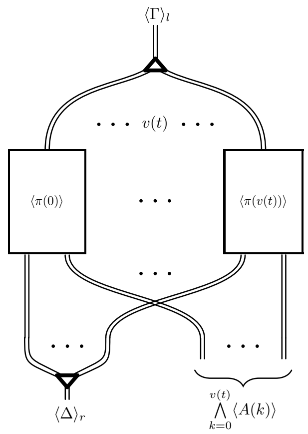
We give some further key cases to aid the reader in recognising how the Paris-Wilkie translation works for deep inference proofs, and further the effects on atomic flows.

A proof $\pi(a)$ extended by a ∀-r step,

$$\forall\text{-r}\ \frac{\Gamma, a \leq t \longrightarrow \Delta, A(a)}{\Gamma \longrightarrow \Delta, \forall x \leq t.A(x)}$$

is translated to the derivation below,

$$
\begin{array}{c}
{\scriptstyle =}\ \dfrac{{\scriptstyle c\uparrow}\ \dfrac{\langle\Gamma\rangle_l}{\langle\Gamma\rangle_l \wedge\cdots\wedge \langle\Gamma\rangle_l} \ \wedge\ {\scriptstyle =}\ \dfrac{\top}{\top\wedge\cdots\wedge\top}}{\langle\Gamma\rangle_l \wedge \langle 0 \leq t\rangle \qquad\qquad \langle\Gamma\rangle_l \wedge \langle v(t) \leq t\rangle} \\
{\scriptstyle\langle\pi(0)\rangle}\Big\| \qquad\quad \wedge\cdots\wedge \quad {\scriptstyle\langle\pi(v(t))\rangle}\Big\| \\
{\scriptstyle s}\ \dfrac{\langle\Delta\rangle_r \vee \langle A(0)\rangle \qquad\qquad \langle\Delta\rangle_r \vee \langle A(t)\rangle}{{\scriptstyle c\downarrow}\ \dfrac{\langle\Delta\rangle_r \vee\cdots\vee \langle\Delta\rangle_r}{\langle\Delta\rangle_r} \vee \bigwedge_{k=0}^{v(t)} \langle A(k)\rangle}
\end{array}
$$



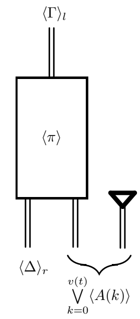
where we notice that $\langle n \leq t\rangle$ is $\top$ for all $n \leq v(t)$. With regards to the length of the flow, besides that inherited from each $\pi(i)$, notice that the $v(t)$ $ac\uparrow$ nodes at the top (for each atom occurrence in $\langle\Gamma\rangle_l$) can be implemented by a complete (almost) balanced tree of depth $\lceil \log v(t)\rceil$, and similarly for the $ac\downarrow$ nodes corresponding to $\langle\Delta\rangle_r$.

A proof π extended by a ∃-r step,

$$\exists\text{-r}\ \frac{\Gamma \longrightarrow \Delta, A(s)}{\Gamma, s \leq t \longrightarrow \Delta, \exists x \leq t.A(x)}$$

is translated to the derivation below,

$$\left[\langle \Delta \rangle_r \vee \begin{array}{c} = \dfrac{\langle \Gamma \rangle_l \wedge \top}{\langle \Gamma \rangle_l} \\ \langle \pi \rangle \Big\| \\ w\!\downarrow \dfrac{\langle A(s) \rangle}{\bigvee\limits_{k=0}^{v(t)} \langle A(k) \rangle} \end{array} \right]$$



assuming $v(s) \leq v(t)$, and so $\langle s \leq t \rangle$ is $\top$. Otherwise $\langle s \leq t \rangle$ is $\bot$ and the derivation is just an axiom $\bot \longrightarrow$ followed by several weakening steps w-l and w-r.

Proofs π_1 and π_2, of $\Gamma \longrightarrow \Delta, A$ and $\Sigma, A \longrightarrow \Pi$ respectively, composed by a *cut* step on A,

$$cut\ \frac{\Gamma \longrightarrow \Delta, A \quad \Sigma, A \longrightarrow \Pi}{\Gamma, \Sigma \longrightarrow \Delta, \Pi}$$

is translated as follows:

$$\begin{array}{l} \begin{array}{c} \langle \Gamma \rangle_l \\ \langle \pi_1 \rangle \Big\| \\ \end{array} \wedge \langle \Sigma \rangle_l \\ s\, \dfrac{\langle \Delta \rangle_r \vee \langle A \rangle \qquad\qquad}{\qquad \langle A \rangle \wedge \langle \Sigma \rangle_l} \\ \langle \Delta \rangle_r \vee \begin{array}{c} \langle \pi_2 \rangle \Big\| \\ \langle \Pi \rangle_r \end{array} \end{array}$$

This can be seen as a simpler version of the translation of an induction step, where the length of flows is increased by only a constant factor rather than a logarithmic factor.

In a similar way, the translations of $\vee$-l and $\wedge$-r steps are simpler versions of $\exists$-l and $\forall$-r steps, respectively, now increasing flow length by at most addition of a constant rather than addition of a logarithm.

The translations of $\wedge$-l and $\vee$-r steps do not affect flows (or derivations) at all, due to our definition of $\langle \cdot \rangle_l$ and $\langle \cdot \rangle_r$.

Finally, the translation of structural steps affect length of a flow by at most addition of a constant. The case for weakening, w-l and w-r, can also be seen as a simpler version of the case for $\exists$-r above. We include a case for contraction, c-r, below for reference, whence the case for c-l is dual.

A proof π extended by a c-r step,

$$c\text{-r}\ \frac{\Gamma \longrightarrow \Delta, A, A}{\Gamma \longrightarrow \Delta, A}$$

is translated as follows:

$$\cfrac{\langle \Delta \rangle_r \quad \vee\, c\!\downarrow \cfrac{\begin{array}{c}\langle \Gamma \rangle_l \\ {\scriptstyle \langle \pi \rangle}\Big\| \end{array}}{\langle A \rangle \vee \langle A \rangle}}{\langle A \rangle}$$

D Appendix for Sect. 5

D.1 The intuitionistic calculus

The intuitionistic caclulus LJ is obtained from LK by insisting that the RHS of each sequent in a proof consists of at most one formula. For this we are also forced to alter the $\vee$-l rule slightly as follows:

$$\vee\text{-l}\ \frac{\Gamma, A \longrightarrow \Delta \quad \Sigma, B \longrightarrow \Delta}{\Gamma, \Sigma, A \vee B \longrightarrow \Delta}$$

Notice that this changes slightly the translation of $\vee$-l steps from arithmetic to propositional logic, but affects the length of flows by only a constant factor. In particular, it can be simulated by the previous $\vee$-l rule by application of a contraction step c-r.

The reason for considering intuitionistic theories is because we wish to reason about manipulations of monotone proofs. In order to do this we incorporate a certain amount of 'negativity' in our theories, namely using implication between two positive formulae to assert the existence of a monotone derivation between them. From here, intuitionistic logic is naturally seen as a logic of proofs in which we can conduct metamathematical reasoning.

D.2 The $\Delta_0^{I_j}$-hierarchy and a restriction to the vocabulary

In order to simplify the definition of the generalised Paris-Wilkie translation later on, we only work with formulae that are $(\forall, \supset)$-combinations of positive formulae. Adding conjunction is simple, and is subsumed by Currying and meta-level conjunction, while disjunction and existentials cause genuine difficulties that are far more cumbersome to deal with.

Nonetheless, we point out that this fragment suffices to prove the various results in the converse direction, in particular Thm. 26, and so we still attain the required correspondence for this theory of arithmetic.[26]

We now give a formal definition of the $\Delta_0^{I_j}$-hierarchy, under this restriction to the vocabulary.

Definition 32 (Formula hierarchy). *Define $\Delta_0^{I_0}$ to be the set of positive bounded formulae, i.e. Δ_0^+. For $j > 0$ we define $\Delta_0^{I_j}$ as follows:*

[26] If the current work is accepted for publication at MFCS '15, then for a final version the statement of Thm. 26 would probably be reworded to take account of this.

- *If* $A \in \Delta_0^{I_{j-1}}$ *then* $A \in \Delta_0^{I_j}$.
- *If* $A \in \Delta_0^{I_{j-1}}$ *and* $B \in \Delta_0^{I_j}$ *then* $A \supset B \in \Delta_0^{I_j}$.
- *If* $A \in \Delta_0^{I_j}$ *then* $\forall x \leq t. \in \Delta_0^{I_j}$.

D.3 On prenexing of monotone formulae in I_1S_2

We show here that we can derive $\forall$-r in I_1S_2, so that it is sound to assume that all monotone formulae are in prenex normal form.

Theorem 33. *I_1S_2 proves $\forall x \leq t.(A(x) \vee B) \longrightarrow (\forall x \leq t.A(x) \vee B)$, where x does not occur free in B.*

Before giving the proof, let us recall from [8] and [12] that the law of excluded middle holds for all quantifier-free formulae free of nonarithmetic symbols already in I_0S_2:

Lemma 34 (Quantifier-free conservativity). *Arithmetic quantifier-free theorems of S_2 are theorems of I_0S_2.*

For this reason let us conservatively extend our language with notation for intervals and subtraction, to ease presentation, assuming that appropriate properties readily follow from their classical proofs.

Proof (of Thm. 33). We reason inside the theory I_1S_2. We aim to prove the following,

$$\forall y.\forall x \leq (t-y).\, [\forall z \in [x, x+y).(A(z) \vee B) \supset (\forall z \in [x, x+y).A(z) \vee B)] \quad (6)$$

for Δ_0^+ formulae $A(\cdot)$ and B by induction on y. Let $C(x,y)$ be such that the formula (6) is $\forall y.\forall x \leq (t-y).C(x,y)$. Notice that $\forall x \leq (t-y).C(x,y)$, the induction formula, is indeed in $\Delta_0^{I_1}$, even under the aforementioned restriction to our vocabulary.

The base case, when $y = 0$, is simple since the quantifiers in $C(x,y)$ collapse to enforcing $z = x$, whence $C(x,y)$ is just an identity.

Now assume, for some b, we have that $\forall x \leq (t-b).C(x,b)$, and let $a \leq (t-2b)$. We will attempt to show that $C(a, 2b)$.

Notice that from $a \leq (t-2b)$ we have that $a \leq (t-b)$ and $(a+b) \leq (t-b)$, by classical reasoning and Lemma 34 above. Therefore, by the inductive hypothesis we have that $C(a,b)$ and $C(a+b,b)$.

Now, let us assume the antecedent, say $C_1(a, 2b)$, of $C(a, 2b)$, i.e. $\forall z \in [a, a+2b).(A(z) \vee B)$, and attempt to deduce the succedent, say $C_2(a, 2b)$, i.e. $\forall z \in [a, a+2b).A(z) \vee B$. Notice that, due to the intuitionistic setting, we cannot at this point query the universal quantifier occurring in the succedent since it is underneath a disjunction.

Now, again by classical reasoning and Lemma 34, we have:

$$c \in [a, a+2b) \equiv c \in [a, a+b) \vee c \in [a+b, a+2b) \quad (7)$$

Consequently, from $C_1(a, 2b)$ we can deduce $\forall z \in [a, a+b).(A(z) \vee B)$ and $\forall z \in [a+b, a+2b)$, i.e. $C_1(a, b)$ and $C_1(a+b, b)$.

Finally, since we already have $C(a, b)$ and $C(a+b, b)$ from the inductive hypothesis, we can conclude $C_2(a, b)$ and $C_2(a+b, b)$. From here we can undistribute over the disjunction with B and can apply the other direction of (7) above to obtain $C_2(a, 2b)$ as required.

The inductive step for $y = 2b+1$ is similar, with one extra case. The theorem now follows from (6) by setting $y = t$ and conducting basic logical manipulations.

D.4 On the strength of intuitionistic induction

The usual simulation of propositional *MLK* in propositional *LJ* is carried over by converting the succedents of *MLK*-sequents to disjunctions of their formulae. We have already shown, above, that this approach also admits simulation of the $\forall$-r rule once $\Delta_0^{I_1}$-*PIND* is available, so in order to show that $MT_2 \subseteq I_1S_2$ it suffices to simulate the usual induction rules, with side formulas on the right, by its respective intuitionistic variants:

$$\textit{PIND}\ \frac{\Gamma, A(\lfloor \frac{a}{2} \rfloor) \longrightarrow A(a)}{\Gamma, A(0) \longrightarrow A(t)} \qquad\qquad \textit{IND}\ \frac{\Gamma, A(a) \longrightarrow A(Sa)}{\Gamma, A(0) \longrightarrow A(t)}$$

Unlike the case for $\forall$-r, to simulate classical Δ_0^+-*IND* (or Δ_0^+-*PIND*), it suffices to work in I_0T_2 (resp. I_0S_2). Below we give such a simulation, deriving from an upper sequent of a positive induction step, $\Gamma, A(a) \longrightarrow \Delta, A(Sa)$, the lower sequent,

$$\frac{\dfrac{\dfrac{\dfrac{\Gamma, A(a) \longrightarrow \bigvee \Delta \vee A(Sa)}{\Gamma \longrightarrow A(a) \supset (\bigvee \Delta \vee A(Sa))}\,{\supset}\text{-r} \qquad \dfrac{\pi}{A(a) \supset (\bigvee \Delta \vee A(Sa)), \bigvee \Delta \vee A(a) \longrightarrow \bigvee \Delta \vee A(Sa)}}{\Gamma, \bigvee \Delta \vee A(a) \longrightarrow \bigvee \Delta \vee A(Sa)}\,\textit{cut}}{\Gamma, \bigvee \Delta \vee A(0) \longrightarrow \bigvee \Delta \vee A(t)}\,\textit{IND}_i}{\Gamma, A(0) \longrightarrow \bigvee \Delta \vee A(t)}\,\vee\text{-inv}$$

where:

- The final '$\vee$-inv' step is obtained by cutting the LHS of its upper sequent against some proof of $A(0) \longrightarrow \bigvee \Delta \vee A(0)$, which must exist by completeness of *LJ* with respect to propositional logic, e.g. which can be taken to be the following:

$$\vee\text{-r}\ \frac{\textit{id}\ \dfrac{}{A(0) \longrightarrow A(0)}}{A(0) \longrightarrow \bigvee \Delta \vee A(0)}$$

  Notice that this use of *cut* is exemplary of the need for cuts anchored to induction steps.

– The proof π exists, again, by completeness of LJ with respect to propositional logic, e.g. we can take π to be the following:

$$\dfrac{\dfrac{\dfrac{}{\bigvee\Delta \longrightarrow \bigvee\Delta}\,id}{\bigvee\Delta \longrightarrow \bigvee\Delta \vee A(Sa)}\,\vee \qquad \dfrac{\dfrac{}{A(a) \longrightarrow A(a)}\,id \qquad \dfrac{}{\bigvee\Delta \vee A(Sa) \longrightarrow \bigvee\Delta \vee A(Sa)}\,id}{A(a), A(a) \supset (\bigvee\Delta \vee A(Sa)) \longrightarrow \bigvee\Delta \vee A(Sa)}\,\supset}{A(a) \supset (\bigvee\Delta \vee A(Sa)), \bigvee\Delta \vee A(a) \longrightarrow \bigvee\Delta \vee A(Sa)}\,\vee$$

The case for $PIND$ is entirely analogous to the case for IND given above.

Finally, in order to show that $I_0 T_2 \subseteq I_1 S_2$, and so Δ_0^+-IND is derivable in $I_1 S_2$, we use the same proof as that in [7] to show that $T_2^i \subseteq S_2^{i+1}$. The idea here is similar to the simulation of $\forall$-r for positive sequents in $I_1 S_2$ above, using a divide-and-conquer induction on a $\Delta_0^{I_1}$-formula instead of the usual induction on a Δ_0^+-formula.

We can now conclude the following:

Theorem 35 (Inclusions). *We have the following inclusions:*

1. $I_0 S_2 \subseteq MS_2 \subseteq MT_2$.
2. $I_0 T_2 \subseteq MT_2$.
3. $MT_2 \subseteq I_1 S_2$
4. $I_j S_2 \subseteq I_j T_2 \subseteq I_{j+1} S_2$.

The same results hold when LFP^{polylog} is included in each theory.

D.5 The generalised translation for $I_1 S_2$

For convenience we will switch back to the sequent calculus presentation of monotone proofs, MLK, whose tree-like variant is polynomially equivalent to Mon [18]. It is just as simple to conduct the translation to Mon, but already existing concepts and terminology for the sequent calculus makes it slightly easier to explain the translation; in particular, the existence of a sequent arrow at the meta-level makes it simple to translate implications between monotone formulae to monotone sequents. This way, our translation also makes it clear why $I_1 T_2$-proofs correspond to dag-like MLK, for which there is no standard definition of a corresponding system based on Mon.[27]

To simplify the translation further, we deal with only additive versions of the branching rules and use an identity rule with weakening 'built in' so that we need not consider any structural steps at all. I.e. we use the following rules in place of their previous analogues,

$$\dfrac{}{\Gamma \longrightarrow \top}\,\top \qquad \dfrac{}{\bot \longrightarrow A}\,\bot \qquad \dfrac{}{\Gamma, A \longrightarrow A}\,id$$

[27] However it is not difficult to construct such a definition, e.g. as a sequence of rewriting derivations, each permitted to utilise steps derived by previous derivations.

$$\textit{cut}\ \frac{\Gamma \longrightarrow A \quad \Gamma, A \longrightarrow B}{\Gamma \longrightarrow B} \qquad \supset\text{-l}\ \frac{\Gamma \longrightarrow A \quad \Gamma, B \longrightarrow C}{\Gamma, A \supset B \longrightarrow C}$$

and no longer have the rules w-l, w-r, c-l, c-r.

An overarching intuition for the general result (including for arbitrary $\Delta_0^{I_j}$) is given by the following result, allowing us to easily see how such formulae can be construed as sets, or conjunctions, of sequents of lower $\Delta_0^{I_j}$ complexity.

Proposition 36 (Normal form for $\Delta_0^{I_1}$). *Every $\Delta_0^{I_1}$-formula can be written in the form $\forall \boldsymbol{x} \leq \boldsymbol{t}.A \supset B$ for $A, B \in \Delta_0^+$.*

Proof. Simply notice that the following double sequents are provable in LJ,

$$\forall x.(A \wedge B(x)) \longleftrightarrow A \wedge \forall x.B(x) \qquad\qquad \forall x.(A \supset B(x)) \longleftrightarrow A \supset \forall x.B(x)$$

where x does not occur free in A above.

However, here, we consider only the case for $\Delta_0^{I_1}$-formulae, and proceed to give a translation $\langle\!\langle \cdot \rangle\!\rangle$ that generalises the previous $\langle \cdot \rangle$ translation.[28]

Definition 37 (Formula and sequent translation). *We define a translation $\langle\!\langle \cdot \rangle\!\rangle$ from closed $\Delta_0^{I_1}$ formulae to multisets of monotone sequents as follows:*

- *If $A \in \Delta_0^+$ then $\langle\!\langle A \rangle\!\rangle := \{ \quad \longrightarrow \langle A \rangle \}$.*
- *If $A \in \Delta_0^+$ and $B \in \Delta_0^{I_1}$ then:*

$$\langle\!\langle A \supset B \rangle\!\rangle \quad := \quad \{ \langle A \rangle , \Gamma \longrightarrow \Delta \quad : \quad \Gamma \longrightarrow \Delta \ \in \ \langle\!\langle B \rangle\!\rangle \}$$

- *If $A \in \Delta_0^{I_1}$ then $\langle\!\langle \forall x \leq t.A \rangle\!\rangle := \bigcup\limits_{k=0}^{v(t)} \langle\!\langle A \rangle\!\rangle$.*
- *For a cedent Γ of $\Delta_0^{I_1}$ formulae, $\langle\!\langle \Gamma \rangle\!\rangle := \bigcup\limits_{A \in \Gamma} \langle\!\langle A \rangle\!\rangle$.*

Since we will need to write *dag-like* proofs, taking care of complexity as we proceed, we turn to the 'Hilbert-Frege' notion of proofs-as-sequences rather than the presentation thusfar of sequent proofs as trees.

In what follows, we will use S and its decorations to vary over sequents.

Definition 38 (Sequent derivations). *An MLK derivation of a sequent S from sequents $S_1, \ldots, S_m$ is a list of sequents $S_1, \ldots, S_m, S_{m+1}, \ldots, S_n$ such that $S_n = S$ and, for $i > m$, each S_i is an axiom, or follows from previously occurring sequents by some rule of MLK.*

For such a derivation, we call $S_1, \ldots, S_m$ the premisses *and S the* conclusion. *If the set of premisses is empty, then we call the derivation a* proof.

[28] We use a different symbol because of the slight ontological difference that, for $A \in \Delta_0^+$, $\langle\!\langle A \rangle\!\rangle$ is defined to be the *sequent* $\longrightarrow \langle A \rangle$ instead of simply $\langle A \rangle$.

We will typically present sequent derivations vertically, writing $\begin{matrix} \boldsymbol{S} \\ \|\pi \\ S \end{matrix}$ for an derivation called π with premisses the (multi)set or list $\boldsymbol{S}$ and conclusion S.

Before proceeding to give the generalised Paris-Wilkie translation of I_1S_2 proofs, we first give a version of the *deduction theorem* that will be useful, e.g. for the $\supset$ -r case in our translation.

Lemma 39 (Deduction theorem). *An MLK derivation* $\begin{matrix} \longrightarrow \Gamma \\ \|\pi \\ \longrightarrow \Delta \end{matrix}$ *can be transformed to an MLK proof* π' *of* $\Gamma \longrightarrow \Delta$*, and vice-versa.*

Proof. The right-left direction is simple: π can be obtained by cutting the given proof against the given initial sequent:

$$\dfrac{\longrightarrow \Gamma \qquad \begin{matrix} \pi' \\ \vdots \\ \Gamma \longrightarrow \Delta \end{matrix}}{\longrightarrow \Delta}\ cut$$

For the left-right direction, we append the cedent Γ to the left-hand side of all sequents occurring in the derivation π:

$$\begin{matrix} \Gamma \longrightarrow \Gamma \\ \Big\|_{\Gamma,\pi} \\ \Gamma \longrightarrow \Delta \end{matrix}$$

It is not difficult to see that the derivation remains valid in *MLK* and, moreover, begins with a correct initial sequent.

We are now ready to define our generalised Paris-Wilkie translation for I_1S_2-proofs of closed $\Delta_0^{I_1}$-formulae. We do not give all the cases below, only the essential ones to understand the technicalities involved and where complexity is generated. As previously mentioned, we work with additive versions of all branching rules, for simplicity.

To show that the dag-like proofs in the image of the translation it suffices to just track the length of the dependency graph, which can be easily seen to be polylogarithmic. The complexity argument itself is similar to that used for atomic flows when $\langle\cdot\rangle$-translating MS_2-proofs, but notice that the translation is rather different: in atomic flows the edges trace atoms, while the dependency graph of a sequent proof traces entire sequents.

Definition 40 (Translation of proofs). *A proof* π *of a sequent* $\Sigma \longrightarrow A$ *is translated by* $\langle\!\langle\cdot\rangle\!\rangle$ *to a multiset of derivations of the following format:*

$$\left\{ \begin{matrix} \langle\!\langle \Sigma \rangle\!\rangle \\ \Big\|_{\langle\!\langle \pi \rangle\!\rangle_S} \\ S \end{matrix} \quad : \quad S \in \langle\!\langle A \rangle\!\rangle \right\}$$

The translation is, once again, by induction on the height (or size) of a I_1S_2 proof. We give some key steps in the translation below.

If a proof π ends with a cut step as follows,

$$\dfrac{\begin{array}{c}\pi_1\\ \vdots\\ \Sigma \longrightarrow A\end{array} \qquad \begin{array}{c}\pi_2\\ \vdots\\ \Sigma, A \longrightarrow B\end{array}}{\Sigma \longrightarrow B}\,cut$$

then by the inductive hypothesis we have derivations $\begin{array}{c}\langle\!\langle\Sigma\rangle\!\rangle\\ \| {\scriptstyle\langle\!\langle\pi_1\rangle\!\rangle_S}\\ S\end{array}$ *for each* $S \in \langle\!\langle A\rangle\!\rangle$*, and derivations* $\begin{array}{c}\langle\!\langle\Sigma\rangle\!\rangle\\ \langle\!\langle A\rangle\!\rangle\\ \| {\scriptstyle\langle\!\langle\pi_2\rangle\!\rangle_{S'}}\\ S'\end{array}$ *for each* $S' \in \langle\!\langle B\rangle\!\rangle$.

Let $\langle\!\langle A\rangle\!\rangle = \{S_i \,:\, i \leq n\}$*. For each* $S' \in \langle\!\langle B\rangle\!\rangle$ *we define* $\langle\!\langle \pi\rangle\!\rangle_{S'}$ *as follows:*

$$\begin{array}{l}\langle\!\langle\Sigma\rangle\!\rangle\\ \quad\| {\scriptstyle\langle\!\langle\pi_1\rangle\!\rangle_{S_0}}\\ S_0\\ \quad\| {\scriptstyle\langle\!\langle\pi_1\rangle\!\rangle_{S_1}}\\ \quad\vdots\\ \quad\| {\scriptstyle\langle\!\langle\pi_1\rangle\!\rangle_{S_n}}\\ S_n\\ \quad\| {\scriptstyle\langle\!\langle\pi_2\rangle\!\rangle_{S'}}\\ S'\end{array}$$

If a proof π ends with an $\supset$ -l step as follows,

$$\dfrac{\begin{array}{c}\pi_1\\ \vdots\\ \Sigma \longrightarrow A\end{array} \qquad \begin{array}{c}\pi_2\\ \vdots\\ \Sigma, B \longrightarrow C\end{array}}{\Sigma, A \supset B \longrightarrow C}\,{\supset}\text{-}l$$

then notice that A must be positive, i.e. in Δ_0^+, since $A \supset B$ must be in $\Delta_0^{I_1}$. Therefore, by the inductive hypothesis, we have a derivation $\begin{array}{c}\langle\!\langle\Sigma\rangle\!\rangle\\ \| {\scriptstyle\langle\!\langle\pi_1\rangle\!\rangle_A}\\ \longrightarrow \langle A\rangle\end{array}$ *and derivations* $\begin{array}{c}\langle\!\langle\Sigma\rangle\!\rangle\\ \langle\!\langle B\rangle\!\rangle\\ \| {\scriptstyle\langle\!\langle\pi_2\rangle\!\rangle_S}\\ S\end{array}$ *for each* $S \in \langle\!\langle C\rangle\!\rangle$.

Let $\langle\!\langle B \rangle\!\rangle = \{\Gamma_i \longrightarrow \Delta_i \ : \ i \leq n\}$. For each $S \in \langle\!\langle C \rangle\!\rangle$ we define $\langle\!\langle \pi \rangle\!\rangle_S$ as follows,

$$
\begin{array}{c}
\langle\!\langle \Sigma \rangle\!\rangle \\
\langle A \rangle, \Gamma_0 \longrightarrow \Delta_0 \\
\| \, {\scriptstyle \langle\!\langle A \supset B \rangle\!\rangle} \\
\langle A \rangle, \Gamma_n \longrightarrow \Delta_n \\
\| \, {\scriptstyle \langle\!\langle \pi_1 \rangle\!\rangle_A} \\
\longrightarrow \langle A \rangle \\
\Gamma_0 \longrightarrow \Delta_0 \\
{\scriptstyle cut} \, \| \\
\Gamma_n \longrightarrow \Delta_n \\
\| \, {\scriptstyle \langle\!\langle \pi_2 \rangle\!\rangle_S} \\
S
\end{array}
$$

where the sequence of formulae marked cut is obtained by cutting $\longrightarrow \langle A \rangle$ above against each $\langle A \rangle, \Gamma_i \longrightarrow \Delta_i$ in $\langle\!\langle A \supset B \rangle\!\rangle$.

If a proof π ends with an $\supset$ -r step as follows,

$$
\begin{array}{c}
\pi' \\
\ddots \vdots \cdot^{\cdot^{\cdot}} \\
\supset\text{-}r \, \dfrac{\Sigma, A \longrightarrow B}{\Sigma \longrightarrow A \supset B}
\end{array}
$$

then, again, notice that A must be positive, i.e. in Δ_0^+, since $A \supset B$ must be in $\Delta_0^{I_1}$. Therefore, by the inductive hypothesis, we have derivations $\begin{array}{c} \langle\!\langle \Sigma \rangle\!\rangle \\ \longrightarrow \langle A \rangle \\ \| \, {\scriptstyle \langle\!\langle \pi' \rangle\!\rangle_S} \\ S \end{array}$ *for each $S \in \langle\!\langle B \rangle\!\rangle$. From here, we can obtain a definition of $\langle\!\langle \pi \rangle\!\rangle_S$ of the appropriate format by simply applying the deduction theorem, Lemma 39.*

The $\forall$ rules are rather simple. The rule $\forall$-l amounts to adding further premisses to an existing derivation, while $\forall$-r essentially follows from expanding out the definition of $\langle\!\langle \cdot \rangle\!\rangle$ on universally quantified formulae.

For the extension of a proof by an induction step, the definition of $\langle\!\langle \cdot \rangle\!\rangle$ is obtained by first converting the induction into finitely many instances of cut (the number determined by the value of the closed term in the succedent of the lower sequent), like in the definition of $\langle \cdot \rangle$, and then applying the definition of $\langle\!\langle \cdot \rangle\!\rangle$ for the case of cut steps.

We omit a complexity analysis here, for brevity but remark that the argument is not dissimilar to that for $\langle \cdot \rangle$ on MS_2. Essentially, the length of the dependency graph of an *MLK* proof in the image of $\langle\!\langle \cdot \rangle\!\rangle$ is bounded by a polylogarithm in the size of the arguments in the conclusion, due to the use of only polynomial induction steps.[29]

[29] The argument also bears semblance to that used in [20] for the theory U_1^1, where it is extension variables rather than dagness that needs to be 'unwound' in a proof.

Theorem 41 (Complexity of $\langle\!\langle\cdot\rangle\!\rangle$). *For an I_1S_2+LFP^{polylog} proof of a $\Delta_0^{I_1}$ sequent $\Sigma(\boldsymbol{x}) \longrightarrow A(\boldsymbol{x})$, there are MLK derivations $\begin{array}{c}\langle\!\langle \Sigma(\boldsymbol{n})\rangle\!\rangle \\ \| \\ S\end{array}$ for each $S \in \langle\!\langle A(\boldsymbol{n})\rangle\!\rangle$ of size quasipolynomial in $\boldsymbol{n}$ and whose dependency graph has length polylogarithmic in $\boldsymbol{n}$.*

From here, we obtain Thm. 23 as a corollary, by considering the special case of when the conclusion of a I_1S_2 is a positive sequent and then applying the deduction theorem, Lemma 39.

E Appendix for Sect. 6

E.1 Some comments on the formalisation and proof of reflection

The formalisation of logical concepts (formulae, proofs etc.), as well as the proof of reflection itself, is fairly standard, but we make some comments below that address a few key subtleties.

- In order to construct divide-and-conquer style arguments on Mon proofs, we represent them as rewriting derivations (or 'Calculus of Structures' style [4]). This way we may simply split derivations in half (or almost) at each inductive step.
- $\mathit{Fla}(x)$ and $\mathit{Der}_{\text{Mon}}(x, y, z)$ are both R-free and definable by Δ_0-formulae, as in [20]. However, we could also define derivations by using least fixed points, expecially given the point above. For example, if $\mathit{Fla}(x)$ is already defined, we could define:

$$\mathit{Der}_{\text{Mon}}(x, y, z) \quad := \quad \begin{array}{l} (\mathit{Fla}(x) \land x = y = z) \\ \lor\ \exists x_1, x_2, y', z' \le \lfloor \frac{2x}{3} \rfloor . \begin{pmatrix} \mathit{Der}_{\text{Mon}}(x_1, y, y') \\ \land\ \mathit{Der}_{\text{Mon}}(x_2, z', z) \\ \land\ \mathit{Step}_{\text{Mon}}(y', z') \end{pmatrix} \end{array}$$

  where $\mathit{Step}_{\text{Mon}}(x, y)$ has the format:

$$\mathit{Fla}(x) \land \mathit{Fla}(y) \land \text{"the expression } \frac{x}{y} \text{ codes a valid inference step of Mon"}$$

- The formula $\mathit{Tr}_R(x)$ is a little more complicated. In general there is no Δ_0-definition of arbitrary truth conditions (although ones for formulae of fixed depth are definable), and so we require least fixed points to deal with this. We would like to do something like the following:

$$\mathit{Tr}_R(x) := \begin{cases} R(y) & x = \ulcorner R(y) \urcorner \\ \mathit{Tr}_R(y) \lor \mathit{Tr}_R(z) & x = y \ulcorner \lor \urcorner z \\ \mathit{Tr}_R(y) \land \mathit{Tr}_R(z) & x = y \ulcorner \land \urcorner z \end{cases}$$

However, this is not necessarily a definition in LFP^{polylog}, since the depth of a propositional formula could be, in the worst case, linear in the size of a formula.
To this end, we rely on a monotone version of Spira's theorem, stating that all propositional formulae can be *balanced*, i.e. converted in polynomial time to a logically equivalent formula whose depth is logarithmic in its size. A similar approach is outlined in [20].
- The proof of reflection itself, Thm. 26, following similar proof ideas found in e.g. [20] and so we omit it. However, it is worth mentioning that, as stated in the proof sketch, we conduct a polynomial induction on a bounded form of Rfn_{MON}, as defined in (5), which is indeed in $\Delta_0^{I_1}$, and so the usual proof goes through fine.