



Programme

Michał J. Gajda, *Arithmetic of monotonic bounds*

We present an arithmetic of monotonic bounds with a syntax-directed, cut-free domination calculus (Definition 3). The calculus is sound for the semantic domination relation $\rho \preccurlyeq \sigma$ ($\llbracket \rho \rrbracket \leq \llbracket \sigma \rrbracket$ in every positive environment): $\rho \sqsubseteq \sigma \Rightarrow \rho \preccurlyeq \sigma$. We prove cut-elimination and give the complete rule set. The key results are the calculus's computational reach beyond primitive recursion [9] and its soundness. All results are mechanised in Agda.*

Definition 1 (Term syntax).

Types and terms over the positive cone $\mathbb{N}_{\geq 1} = \mathbb{N} \setminus \{0\}$:

$$\begin{aligned} \tau &::= \mathbf{1} \mid \tau \Rightarrow \tau \\ t &::= x \mid \mathbf{1} \mid \mathbb{S} t \mid \mathbb{P} t \mid \max t t \mid \lambda x. t \mid t \mid t \mid \mathbb{I} t t t \end{aligned}$$

$\mathbb{S}$ is successor, $\mathbb{P}$ the clamped predecessor (clamped at 1), an *explicit* λ binder, and iteration $\mathbb{I}(f, k, a)$ composes the function f k times from

start a : $\mathbb{I}(f,k,a) = f^k(a)$. Terms are intrinsically typed, so only well-typed terms are formed.

Definition 2 (Semantics).

Denotation is higher-order, given by:

$$\llbracket \iota \rrbracket = \mathbb{N}_{\geq 1}, \quad \llbracket \sigma \Rightarrow \tau \rrbracket = \llbracket \sigma \rrbracket \rightarrow \llbracket \tau \rrbracket.$$

Evaluation is defined by structural recursion:

$$\begin{aligned} \llbracket x \rrbracket &= \text{environment lookup} \\ \llbracket \mathbf{1} \rrbracket &= 1 \\ \llbracket \mathbb{S} \ t \rrbracket &= \llbracket t \rrbracket + 1 \\ \llbracket \mathbb{P} \ t \rrbracket &= \max(\llbracket t \rrbracket - 1, 1) \\ \llbracket \max \ t_1 \ t_2 \rrbracket &= \max(\llbracket t_1 \rrbracket, \llbracket t_2 \rrbracket) \\ \llbracket \lambda x. \ t \rrbracket &= \lambda v. \llbracket t \rrbracket [v/x] \\ \llbracket t_1 \ t_2 \rrbracket &= \llbracket t_1 \rrbracket (\llbracket t_2 \rrbracket) \\ \llbracket \mathbb{I} \ f \ k \ a \rrbracket &= \llbracket f \rrbracket^{\llbracket k \rrbracket} (\llbracket a \rrbracket) \end{aligned}$$

where $\llbracket f \rrbracket^{\llbracket k \rrbracket}$ means $\llbracket k \rrbracket$ -fold composition. Evaluation is total (intrinsic typing ensures well-foundedness).

Iteration at type $\iota \rightarrow \iota$ climbs the hyperoperation hierarchy: addition, multiplication, exponentiation, tetration, and the tower $tetr \ 2 \ b = 2 \uparrow\uparrow b$ are all single closed λ -terms derived from $\{\mathbb{S}, \mathbb{P}, \mathbf{1}, \mathbb{I}, \max\}$ (Table 1). The carrier has no additive $\mathbf{0}$: multiplication iterates from $\mathbf{1}$ and overshoots by one, with $\mathbb{P}$ correcting the offset; exponentiation iterates from $\mathbf{1}$, its genuine base $a^0 = 1$.

Table 1: The closed terms of the calculus: every bound, dominator and big-O witness is a single closed λ -term built from $\{\mathbb{S}, \mathbb{P}, \mathbf{1}, \mathbb{I}, \max\}$. $\mathbb{S}$, $\mathbb{P}$, $\max$ are the primitive constants (shown in function form); the rest are derived.

term	type	definition	explanation
$\mathbb{S}$	$\iota \rightarrow \iota$	$\lambda x. \ \mathbb{S} \ x$	$n+1$
$\mathbb{P}$	$\iota \rightarrow \iota$	$\lambda x. \ \mathbb{P} \ x$	$\max(n-1, 1)$
$\max$	$\iota \rightarrow \iota \rightarrow \iota$	$\lambda a \ b. \ \max(a, b)$	$\max(a, b)$
add	$\iota \rightarrow \iota \rightarrow \iota$	$\lambda a \ b. \ \mathbb{I}(\mathbb{S}, b, a)$	$a+b$
mul	$\iota \rightarrow \iota \rightarrow \iota$	$\lambda a \ b. \ \mathbb{P}(\mathbb{I}(\lambda x. \ a+x, b, \mathbf{1}))$	$a \cdot b$
exp	$\iota \rightarrow \iota \rightarrow \iota$	$\lambda a \ b. \ \mathbb{I}(\lambda x. \ x \cdot a, b, \mathbf{1})$	a^b
$tetr$	$\iota \rightarrow \iota \rightarrow \iota$	$\lambda a \ b. \ \mathbb{I}(\lambda x. \ a^x, b, \mathbf{1})$	$a \uparrow\uparrow b$

term	type	definition	explanation
$tower$	$\iota \rightarrow \iota$	$\lambda b. tetr\ 2\ b$	$2\uparrow\uparrow b$
Φ	$(\iota \rightarrow \iota) \rightarrow (\iota \rightarrow \iota)$	$\lambda g. \lambda y. \mathbb{I}(g, \mathbb{S}\ y, g\ \mathbf{1})$	level step
$fAck$	$\iota \rightarrow \iota$	$\lambda n. \mathbb{I}(\Phi, n, \mathbb{S})\ n$	$\geq ack(n, n)$
Ψ	$(\iota \rightarrow \iota \rightarrow \iota) \rightarrow (\iota \rightarrow \iota \rightarrow \iota)$	$\lambda h. \lambda a\ b. \mathbb{I}(\lambda x. h\ a\ x, \mathbb{S}\ b, a)$	hyperop. step
Hyp	$\iota \rightarrow \iota$	$\lambda n. \mathbb{I}(\Psi, n, add)\ 3\ n$	$\geq ack(n, n)$
$c \cdot g$	ι	$\mathbb{P}(\mathbb{I}(\lambda x. x + g, c, \mathbf{1}))$	$c \cdot g$ (big-O witness)

The predicates used for the iteration rules below are:

- inflationary f : f is **inflationary** — $x \leq \llbracket f \rrbracket(x)$ for every x in every positive environment.
- monotone f : f is **monotone** — $x \leq y$ implies $\llbracket f \rrbracket(x) \leq \llbracket f \rrbracket(y)$.
- $f \triangleleft f'$: **strict domination** — $f \trianglelefteq f'$ but $f' \not\trianglelefteq f$.

The auxiliary judgments inflationary f and monotone f have syntactic inference rules (mechanised by structural recursion); the full calculus follows Abel [1] and Icard–Moss–Tune [5]:

inflationary $\mathbb{S}$ inflationary $\lambda x. x$ inflationary $\lambda x. \mathbb{I}\ f\ k\ x$ if inflationary f , inflationary k , inflatic
monotone $\mathbb{S}$ monotone $\mathbb{P}$ monotone $\max$

The domination calculus below is cut-free (transitivity is admissible via the successor-step and max-introduction rules).

Definition 3 (Domination calculus).

$$\begin{array}{c}
\rho \trianglelefteq \rho \text{ [refl]} \quad \mathbf{1} \trianglelefteq \rho \text{ [pos]} \quad \frac{\rho \trianglelefteq \sigma}{\rho \trianglelefteq \mathbb{S}\ \sigma} \text{ [sucStep]} \\
\\
\frac{\rho \trianglelefteq \sigma}{\mathbb{S}\ \rho \trianglelefteq \mathbb{S}\ \sigma} \text{ [sucM]} \\
\\
a \trianglelefteq \max(a, b) \text{ [maxL]} \quad b \trianglelefteq \max(a, b) \text{ [maxR]} \quad \frac{a \trianglelefteq b}{a \trianglelefteq \max(b, c)} \text{ [maxLI]} \\
\\
\frac{a \trianglelefteq c}{a \trianglelefteq \max(b, c)} \text{ [maxRI]} \\
\\
\frac{a \trianglelefteq c \quad b \trianglelefteq c}{} \text{ [maxE]} \quad \frac{a \trianglelefteq c \quad b \trianglelefteq d}{} \text{ [maxM]}
\end{array}$$

$$\max(a,b) \leq c$$

$$\max(a,b) \leq \max(c,d)$$

$$\frac{\text{inflationary } f' \quad \text{monotone } f' \quad f \leq f' \quad k \leq k' \quad a \leq a'}{\mathbb{I}(f,k,a) \leq \mathbb{I}(f',k',a')} \quad [\text{iterC}]$$

$$\frac{\text{inflationary } f' \quad \text{monotone } f' \quad f < f' \quad k \leq k' \quad a \leq a'}{\mathbb{I}(f,k,a) \leq \mathbb{I}(f',k',a')} \quad [\text{iterB}]$$

The calculus is syntax-directed: the head constructor of a guides rule selection; the auxiliary judgments inflationary f , monotone f , and $f \leq f'$ for function-type terms each have their own inference systems ($f \leq f'$ for function terms is defined analogously via λ -body comparison and transitivity).

lhs a	rhs b	side condition	rule
a	a		[refl]
1	b		[pos]
$\$ a_1$	$\$ b_1$		[sucM]
$\max(a_1, a_2)$	$\max(b_1, b_2)$		[maxM]
	$\neq$ $\max(b_1, b_2)$		[maxE]
$\mathbb{I}(f, k, a_0)$	$\mathbb{I}(f', k', a_0')$	inflationary f' , monotone f' , $f \leq f'$	[iterC]
	$\mathbb{I}(f', k', a_0')$	inflationary f' , monotone f' , $f < f'$	[iterB]
a_1 (where $b = \max(a_1, b_2)$)	$\max(a_1, b_2)$		[maxL]
a_2 (where $b = \max(b_1, a_2)$)	$\max(b_1, a_2)$		[maxR]
(any a)	$\$ b_1$		[sucStep]
	$\max(b_1, b_2)$	$a \leq b_1$	[maxLI]
	$\max(b_1, b_2)$	$a \leq b_2$	[maxRI]

The calculus $\leq$ extends the class of functions reachable beyond primitive recursion [2] while remaining deductively weaker than PRA in the sense of Schirrn and Niebergall [9]: the if-free, induction-free rule set cannot capture $n+1 \leq 2n$ without explicit max, a gap

inherited from the prior domination inference system [4]. Unlike first-order arithmetics (Presburger [7], Skolem [10][6], Robinson Q [8], or Buss's S_2^1 [3]), the present system is a purely syntactic calculus of term inequalities — not a theory with a model class — so questions of semantic completeness or arithmetic decidability do not directly apply.

Theorem 4 (Ackermann bound).

The closed terms $fAck$ and Hyp (Table 1) satisfy $fAck(n) \geq ack(n,n)$ and $Hyp(n) \geq ack(n,n)$ for every n on the positive carrier.

Hence the arithmetic reaches beyond all primitive-recursive functions (Ackermann 1928).

Theorem 5 (Soundness).

Every derivation $\rho \trianglelefteq \sigma$ in the domination calculus is valid in every positive environment: $\llbracket \rho \rrbracket \leq \llbracket \sigma \rrbracket$ whenever the environment assigns values ≥ 1 .

Theorem 6 (Transitivity elimination).

The domination calculus admits a transitivity rule: if $\rho \trianglelefteq \sigma$ and $\sigma \trianglelefteq \tau$ then $\rho \trianglelefteq \tau$. Transitivity is not primitive — it is admissible via the `sucStep`, `maxLI`, and `maxRI` rules together with the syntax-directed rule set.

This work generalises a prior domination inference system [4] and proves these results in Agda using a weak arithmetic.

* Agda reproducibility package: <https://doi.org/10.5281/zenodo.21418369>

[1] Abel, Andreas. 2006. *Polarized Subtyping for Sized Types*. In Computer Science – Theory and Applications (CSR 2006), 3967:381–92. LNCS. Springer. https://doi.org/10.1007/11753728_39.

[2] Ackermann, Wilhelm. 1928. *Zum Hilbertschen Aufbau Der Reellen Zahlen*. Mathematische Annalen 99 (1): 118–33. <https://doi.org/10.1007/BF01459088>.

[3] Buss, Samuel R. 1986. *The Polynomial Hierarchy and Intuitionistic Bounded Arithmetic*. In Structure in Complexity Theory, Proceedings of the Conference held at the University of California, Berkeley, California, USA, June 2-5, 1986, edited by Alan L. Selman, 223:77–103. Lecture Notes in Computer Science. Springer. https://doi.org/10.1007/3-540-16486-3/_91.

[4] Gajda, Michał J. 2024. *Consistent Ultrafinitist Logic*. In 29th International Conference on Types for Proofs and Programs (TYPES 2023), 303:5:1–20. LIPIcs.

- [5] Icard, Thomas, Lawrence S. Moss, and William Tune. 2017. *A Monotonicity Calculus and Its Completeness*. In Proceedings of the 15th Meeting on the Mathematics of Language (MoL 2017), 75–87. ACL. <https://doi.org/10.18653/v1/w17-3408>.
- [6] Mostowski, Andrzej. 1952. *On Direct Products of Theories*. Journal of Symbolic Logic 17 (1): 1–31. <https://doi.org/10.2307/2267454>.
- [7] Presburger, Mojżesz. 1930. *Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt*. In Sprawozdanie z I Kongresu Matematyków Słowiańskich, 92–101. Warsaw.
- [8] Robinson, Raphael M. 1950. *An Essentially Undecidable Axiom System*. Proceedings of the International Congress of Mathematicians, 729–30.
- [9] Schirn, Matthias, and Karl-Georg Niebergall. 2005. *Finitism = PRA? On a Thesis of W. W. Tait*. Reports on Mathematical Logic 39 (January): 3–24. <https://rml.tcs.uj.edu.pl/rml-39/a-sch-39.htm>.
- [10] Skolem, Thoralf. 1930. *Über gewisse Satzfunktionen in der Arithmetik*. Skrifter Utgitt Av Videnskapsselskapet i Kristiania, I. Klasse, no. 7.

Advanced math rendering powered by



JAF 45 WARSAW

e-mail: jaf45warsaw@uw.edu.pl