Non-Uniform Polynomial Time and Non-Wellfounded Parsimonious Proofs

FICS

Naples, 19 February 2024

Matteo Acclavio [1] Gianluca Curzi [2] Giulio Guerrieri [1]

[1]University of Sussex

[2]University of Gothenburg

What is this talk about?

- Parsimonious linear logic = subsystem of LL based on the principles of **parsimonious logic** [MT15, Maz15]

  linear logic modalities $!, ?$ as least and greatest fixed points

  $$!A \;\sim\; \textbf{streams} \qquad\qquad ?A \;\sim\; \textbf{lists}$$

- In [MT15] characterisation of complexity classes via parsimonious logic:

  $$\begin{array}{lcl} \mathbf{P} & = & \text{polynomial time decidability} \\ \mathbf{P}/\text{poly} & = & \textbf{non-uniform} \text{ version of } \mathbf{P} \end{array}$$

- **This talk:** non-wellfounded proof systems for parsimonious linear logic:

  $$\mathsf{wrPLL}_2^\infty \text{ (non-uniform)} \quad \text{vs} \quad \mathsf{rPLL}_2^\infty \text{ (uniform)}$$

- **Our main result:** complexity-theoretic **characterisations**:

  $$\mathsf{wrPLL}_2^\infty = \mathbf{P}/\text{poly} \quad \text{vs} \quad \mathsf{rPLL}_2^\infty = \mathbf{P}$$

What is this talk about?

- Parsimonious linear logic = subsystem of LL based on the principles of **parsimonious logic** [MT15, Maz15]

  linear logic modalities $!, ?$ as least and greatest fixed points

  $$!A \sim \textbf{streams} \qquad\qquad ?A \sim \textbf{lists}$$

- In [MT15] characterisation of complexity classes via parsimonious logic:

P	=	polynomial time decidability
P/poly	=	**non-uniform** version of **P**

- **This talk:** non-wellfounded proof systems for parsimonious linear logic:

  $$\mathsf{wrPLL}_2^\infty \text{ (non-uniform)} \quad \text{vs} \quad \mathsf{rPLL}_2^\infty \text{ (uniform)}$$

- **Our main result:** complexity-theoretic **characterisations**:

  $$\mathsf{wrPLL}_2^\infty = \mathbf{P}/\mathrm{poly} \quad \text{vs} \quad \mathsf{rPLL}_2^\infty = \mathbf{P}$$

What is this talk about?

- Parsimonious linear logic = subsystem of LL based on the principles of **parsimonious logic** [MT15, Maz15]

  linear logic modalities $!, ?$ as least and greatest fixed points

  $$!A \sim \textbf{streams} \qquad\qquad ?A \sim \textbf{lists}$$

- In [MT15] characterisation of complexity classes via parsimonious logic:

  $$\begin{array}{lcl} \mathbf{P} & = & \text{polynomial time decidability} \\ \mathbf{P}/\text{poly} & = & \textbf{non-uniform} \text{ version of } \mathbf{P} \end{array}$$

- **This talk:** non-wellfounded proof systems for parsimonious linear logic:

  $$\mathsf{wrPLL}_2^\infty \text{ (non-uniform)} \quad \text{vs} \quad \mathsf{rPLL}_2^\infty \text{ (uniform)}$$

- **Our main result:** complexity-theoretic **characterisations**:

  $$\mathsf{wrPLL}_2^\infty = \mathbf{P}/\text{poly} \quad \text{vs} \quad \mathsf{rPLL}_2^\infty = \mathbf{P}$$

What is this talk about?

- Parsimonious linear logic = subsystem of LL based on the principles of **parsimonious logic** [MT15, Maz15]

  linear logic modalities $!, ?$ as least and greatest fixed points

  $$!A \sim \textbf{streams} \qquad ?A \sim \textbf{lists}$$

- In [MT15] characterisation of complexity classes via parsimonious logic:

  $$\begin{array}{lcl} \mathbf{P} & = & \text{polynomial time decidability} \\ \mathbf{P}/\text{poly} & = & \textbf{non-uniform} \text{ version of } \mathbf{P} \end{array}$$

- **This talk:** non-wellfounded proof systems for parsimonious linear logic:

  $$\mathsf{wrPLL}_2^\infty \text{ (non-uniform)} \quad \text{vs} \quad \mathsf{rPLL}_2^\infty \text{ (uniform)}$$

- **Our main result:** complexity-theoretic **characterisations**:

  $$\mathsf{wrPLL}_2^\infty = \mathbf{P}/\text{poly} \quad \text{vs} \quad \mathsf{rPLL}_2^\infty = \mathbf{P}$$

Parsimonious linear logic

- From linear logic...

$$\textit{dig}\,\frac{\Gamma, ??A}{\Gamma, ?A} \qquad \textit{fp}\,\frac{\Gamma, A}{?\Gamma, !A} \qquad \textit{w}\,\frac{\Gamma}{\Gamma, ?A} \qquad \textit{abs}\,\frac{\Gamma, A, ?A}{\Gamma, ?A}$$

digging — functorial promotion — weakening — absorption

- **Key idea:** $!A$ as a type of (very special) **streams**

$$\textit{cut}\,\dfrac{\textit{fp}\,\dfrac{\overset{\mathcal{D}}{A}}{!A} \qquad \textit{abs}\,\dfrac{\otimes\,\dfrac{\textit{ax}\,\dfrac{}{A^\perp, A} \qquad \textit{ax}\,\dfrac{}{?A^\perp, ?A}}{A^\perp, ?A^\perp, A \otimes ?A}}{?A^\perp, A \otimes ?A}}{A \otimes !A} \quad \rightsquigarrow^* \quad \otimes\,\dfrac{\overset{\mathcal{D}}{A} \qquad \textit{fp}\,\dfrac{\overset{\mathcal{D}}{A}}{!A}}{A \otimes !A}$$

$$\mathsf{pop}\,\langle \mathcal{D}, \mathcal{D}, \dots, \mathcal{D}, \dots \rangle \qquad\qquad \mathcal{D} \otimes \langle \mathcal{D}, \mathcal{D}, \dots, \mathcal{D}, \dots \rangle$$

Parsimonious linear logic

- ... to parsimonious linear logic (PLL_2)

$$\cancel{dig\ \dfrac{\Gamma, ??A}{\Gamma, ?A}} \qquad fp\ \dfrac{\Gamma, A}{?\Gamma, !A} \qquad w\ \dfrac{\Gamma}{\Gamma, ?A} \qquad abs\ \dfrac{\Gamma, A, ?A}{\Gamma, ?A}$$

digging	functorial promotion	weakening	absorption

- **Key idea:** $!A$ as a type of (very special) **streams**

$$cut\ \dfrac{fp\ \dfrac{\overset{\mathcal{D}}{A}}{!A} \qquad abs\ \dfrac{\otimes\ \dfrac{ax\ \dfrac{}{A^{\perp}, A} \qquad ax\ \dfrac{}{?A^{\perp}, ?A}}{A^{\perp}, ?A^{\perp}, A \otimes ?A}}{?A^{\perp}, A \otimes ?A}}{A \otimes !A} \quad \rightsquigarrow^* \quad \otimes\ \dfrac{\overset{\mathcal{D}}{A} \qquad fp\ \dfrac{\overset{\mathcal{D}}{A}}{!A}}{A \otimes !A}$$

$\mathsf{pop}\ \langle \mathcal{D}, \mathcal{D}, \dots, \mathcal{D}, \dots \rangle \qquad\qquad \mathcal{D} \otimes \langle \mathcal{D}, \mathcal{D}, \dots, \mathcal{D}, \dots \rangle$

Parsimonious linear logic

- ... to parsimonious linear logic (PLL_2)

$$\cancel{dig\ \frac{\Gamma, ??A}{\Gamma, ?A}} \qquad fp\ \frac{\Gamma, A}{?\Gamma, !A} \qquad w\ \frac{\Gamma}{\Gamma, ?A} \qquad abs\ \frac{\Gamma, A, ?A}{\Gamma, ?A}$$

digging	functorial promotion	weakening	absorption

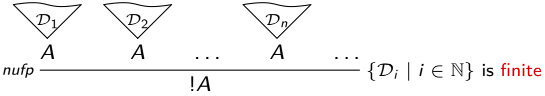
- **Key idea:** $!A$ as a type of (very special) **streams**

$$cut\ \cfrac{fp\ \cfrac{\overset{\mathcal{D}}{A}}{!A} \qquad abs\ \cfrac{\otimes\ \cfrac{ax\ \cfrac{}{A^{\perp}, A} \qquad ax\ \cfrac{}{?A^{\perp}, ?A}}{A^{\perp}, ?A^{\perp}, A \otimes ?A}}{?A^{\perp}, A \otimes ?A}}{A \otimes !A} \quad \rightsquigarrow^{*} \quad \otimes\ \cfrac{\overset{\mathcal{D}}{A} \qquad fp\ \cfrac{\overset{\mathcal{D}}{A}}{!A}}{A \otimes !A}$$

$\mathsf{pop}\,\langle \mathcal{D}, \mathcal{D}, \ldots, \mathcal{D}, \ldots \rangle$ $\qquad\qquad$ $\mathcal{D} \otimes \langle \mathcal{D}, \mathcal{D}, \ldots, \mathcal{D}, \ldots \rangle$

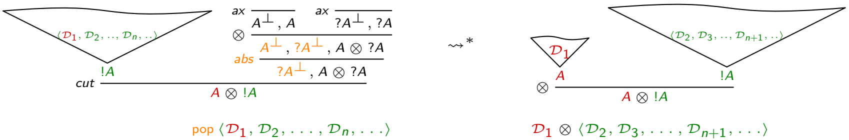
Non-uniform parsimonious linear logic

- Non-uniform PLL (nuPLL) = replace *fp* with the following:



$\sim \quad \langle \mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_n, \dots \rangle$

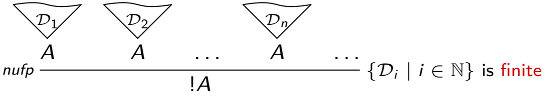
- **Improvement:** $!A$ as a type of streams over finite data of type A



- *nufp* introduces some proof-theoretical notion of **non-uniformity** that deeply interfaces with complexity-theoretic non-uniformity.

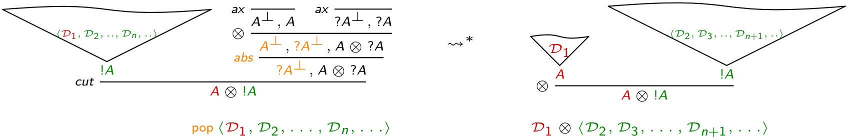
Non-uniform parsimonious linear logic

- Non-uniform PLL (nuPLL) = replace *fp* with the following:



$\sim \quad \langle \mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n, \ldots \rangle$

- **Improvement:** $!A$ as a type of streams over finite data of type A



- *nufp* introduces some proof-theoretical notion of **non-uniformity** that deeply interfaces with complexity-theoretic non-uniformity.

Non-uniformity via non-wellfoundedness

- **Goal:** non-wellfounded formulations of nuPLL and PLL
- (Partial) **recipe**:

  (1) Replace *fp* and *nufp* with conditional promotion (*cp*): $\mathsf{cp}\dfrac{\Gamma, A \quad ?\Gamma, !A}{?\Gamma, !A}$

  (2) From (wellfounded) **infinite branching** to **non-wellfounded** (finite branching):

$=$ non-wellfounded box

  (3) Progressing condition = logical consistency

  (4) Weak regularity = finiteness condition (∗) for non-wellfounded boxes

Non-uniformity via non-wellfoundedness

- **Goal:** non-wellfounded formulations of nuPLL and PLL
- (Partial) **recipe**:

(1) Replace *fp* and *nufp* with conditional promotion (*cp*): $\dfrac{\Gamma, A \quad ?\Gamma, !A}{?\Gamma, !A}\,cp$

(2) From (wellfounded) **infinite branching** to **non-wellfounded** (finite branching):

=
non-wellfounded box

(3) Progressing condition = logical consistency

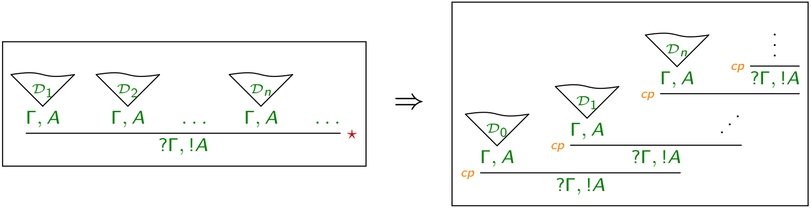
(4) Weak regularity = finiteness condition ($\star$) for non-wellfounded boxes

Non-uniformity via non-wellfoundedness

- **Goal:** non-wellfounded formulations of nuPLL and PLL
- (Partial) **recipe**:

(1) Replace *fp* and *nufp* with conditional promotion (*cp*):
$$\frac{\Gamma, A \quad ?\Gamma, !A}{?\Gamma, !A}\ cp$$

(2) From (wellfounded) **infinite branching** to **non-wellfounded** (finite branching):



=
non-wellfounded box

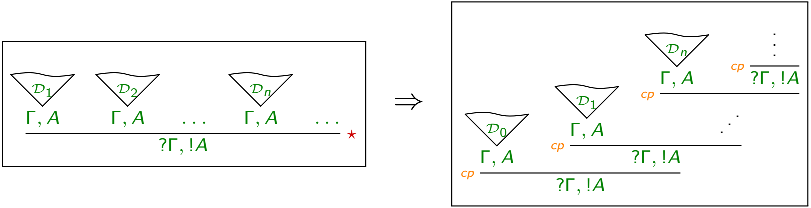
(3) Progressing condition = logical consistency

(4) Weak regularity = finiteness condition (⋆) for non-wellfounded boxes

Non-uniformity via non-wellfoundedness

- **Goal:** non-wellfounded formulations of nuPLL and PLL
- (Partial) **recipe**:
  (1) Replace *fp* and *nufp* with conditional promotion (cp): $\dfrac{\Gamma, A \quad ?\Gamma, !A}{?\Gamma, !A}\,cp$

  (2) From (wellfounded) **infinite branching** to **non-wellfounded** (finite branching):



=
non-wellfounded box

  (3) Progressing condition = logical consistency

  (4) Weak regularity = finiteness condition ($\star$) for non-wellfounded boxes

Regularity vs weak regularity

- Weakly regular = finitely many distinct subproofs whose conclusions are left premises of *cp* rules

  Idea: streams have finite support ...

- Regular proof = finitely many distinct subproofs

  Idea: streams are periodic, so they only represent computable real numbers

$\langle \mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n, \ldots \rangle \quad \sim$

Regularity vs weak regularity

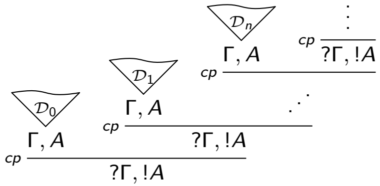
- Weakly regular = finitely many distinct subproofs whose conclusions are left premises of *cp* rules

  Idea: streams have finite support ...

- Regular proof = finitely many distinct subproofs

  Idea: streams are periodic, so they only represent **computable** real numbers

$$\langle \mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n, \ldots \rangle \qquad \sim$$



Regularity vs weak regularity

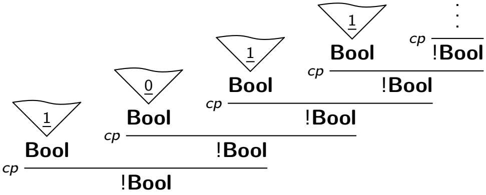
- Weakly regular = finitely many distinct subproofs whose conclusions are left premises of *cp* rules

  Idea: streams have finite support . . . so they only represent real numbers

- Regular proof = finitely many distinct subproofs

  Idea: streams are periodic, so they only represent **computable** real numbers

$$\langle \underline{1}, \underline{0}, \underline{1}, \underline{1}, \underline{0}, \underline{1}, \underline{1}, \underline{1}, \underline{0}, \ldots \rangle \quad =$$



Regularity vs weak regularity

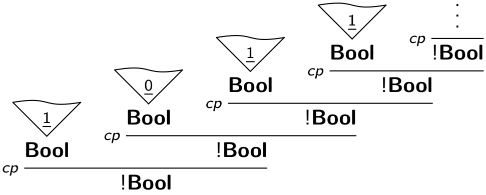
- Weakly regular = finitely many distinct subproofs whose conclusions are left premises of *cp* rules

  Idea: streams have finite support . . . so they only represent real numbers

- Regular proof = finitely many distinct subproofs

  Idea: streams are periodic, so they only represent **computable** real numbers

$\langle \underline{1}, \underline{0}, \underline{1}, \underline{1}, \underline{0}, \underline{1}, \underline{1}, \underline{1}, \underline{0}, \ldots \rangle \quad =$



Regularity vs weak regularity

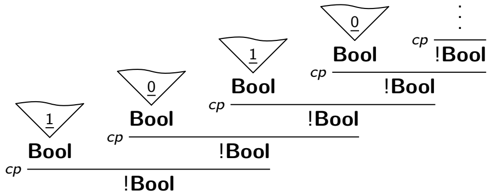
- Weakly regular = finitely many distinct subproofs whose conclusions are left premises of *cp* rules

  Idea: streams have finite support . . . so they only represent real numbers

- Regular proof = finitely many distinct subproofs

  Idea: streams are periodic, so they only represent **computable** real numbers

$\langle \underline{1}, \underline{0}, \underline{1}, \underline{0}, \underline{1}, \underline{0}, \ldots \rangle \quad =$



Two non-wellfounded proof systems

- Two non-wellfounded proof systems for parsimonious linear logic:

  wrPLL^{∞} = weakly regular progressing [...] proofs

  rPLL^{∞} = regular progressing [...] proofs

- Relating inductive and non-wellfounded systems:

	inductive	non-wellfounded
non-uniform	nuPLL	wrPLL^{∞}
uniform	PLL	rPLL^{∞}

Two non-wellfounded proof systems

- Two non-wellfounded proof systems for parsimonious linear logic:

  wrPLL^∞ = weakly regular progressing [. . .] proofs

  rPLL^∞ = regular progressing [. . .] proofs

- Relating inductive and non-wellfounded systems:

	inductive	**non-wellfounded**
non-uniform	nuPLL	wrPLL^∞
uniform	PLL	rPLL^∞

Cyclic implicit complexity

- Implicit computational complexity (ICC) = characterise complexity classes by means of languages/calculi **without** explicit reference to machine models or external resource bounds.
  - Originates with the Bellantoni and Cook's paper on **safe recursion** [BC92].
  - Borrows techniques from recursion theory, proof/type theory, model theory ...
  - Pervasive notion of **stratification**: data are organized into *strata*
  - **Example: light linear logics** = weaker versions of linear logic modality ! that induce a bound on cut-elimination [Gir87, DJ03, Laf04]
- Cyclic Implicit Complexity (CIC)
  - introduced in joint works with Anupam Das [CD22, CD23b], aiming at studying the principles of ICC using the tools of non-wellfounded and cyclic proof-theory
  - characterisation of **P** and **ELEMENTARY** [CD22], and **P**/poly [CD23b]
  - **This talk: CIC applied to LL!**

Cyclic implicit complexity

- Implicit computational complexity (ICC) = characterise complexity classes by means of languages/calculi **without** explicit reference to machine models or external resource bounds.
  - Originates with the Bellantoni and Cook's paper on **safe recursion** [BC92].
  - Borrows techniques from recursion theory, proof/type theory, model theory ...
  - Pervasive notion of **stratification**: data are organized into *strata*
  - **Example:** **light linear logics** = weaker versions of linear logic modality ! that induce a bound on cut-elimination [Gir87, DJ03, Laf04]

- Cyclic Implicit Complexity (CIC)
  - introduced in joint works with Anupam Das [CD22, CD23b], aiming at studying the principles of ICC using the tools of non-wellfounded and cyclic proof-theory
  - characterisation of P and **ELEMENTARY** [CD22], and P/poly [CD23b]
  - **This talk:** CIC applied to LL!

Cyclic implicit complexity

- Implicit computational complexity (ICC) = characterise complexity classes by means of languages/calculi **without** explicit reference to machine models or external resource bounds.
  - Originates with the Bellantoni and Cook's paper on **safe recursion** [BC92].
  - Borrows techniques from recursion theory, proof/type theory, model theory ...
  - Pervasive notion of **stratification**: data are organized into *strata*
  - **Example:** **light linear logics** = weaker versions of linear logic modality ! that induce a bound on cut-elimination [Gir87, DJ03, Laf04]

- Cyclic Implicit Complexity (CIC)
  - introduced in joint works with Anupam Das [CD22, CD23b], aiming at studying the principles of ICC using the tools of non-wellfounded and cyclic proof-theory
  - characterisation of **P** and **ELEMENTARY** [CD22], and **P**/poly [CD23b]
  - **This talk:** CIC applied to LL!

Cyclic implicit complexity

- Implicit computational complexity (ICC) = characterise complexity classes by means of languages/calculi **without** explicit reference to machine models or external resource bounds.
  - Originates with the Bellantoni and Cook's paper on **safe recursion** [BC92].
  - Borrows techniques from recursion theory, proof/type theory, model theory ...
  - Pervasive notion of **stratification**: data are organized into *strata*
  - **Example:** **light linear logics** = weaker versions of linear logic modality ! that induce a bound on cut-elimination [Gir87, DJ03, Laf04]

- Cyclic Implicit Complexity (CIC)
  - introduced in joint works with Anupam Das [CD22, CD23b], aiming at studying the principles of ICC using the tools of non-wellfounded and cyclic proof-theory
  - characterisation of **P** and **ELEMENTARY** [CD22], and **P**/poly [CD23b]
  - **This talk:** CIC applied to LL!

Non-uniform complexity

- $\mathbf{P}/\text{poly}$ = class of problems decidable in **non-uniform** polynomial time

  Theorem: $A \in \mathbf{P}/\text{poly}$ iff A decided by family of polynomial size circuits

- $\mathbf{P}(\mathbb{R})$ = class of problems decidable in polynomial time by a Turing machine **querying bits of real numbers**"

  Theorem [Folklore]: $\mathbf{P}/\text{poly} = \mathbf{P}(\mathbb{R})$.

Non-uniform complexity

- $\mathbf{P}/\mathrm{poly}$ = class of problems decidable in **non-uniform** polynomial time

  Theorem: $A \in \mathbf{P}/\mathrm{poly}$ iff A decided by family of polynomial size circuits

- $\mathbf{P}(\mathbb{R})$ = class of problems decidable in polynomial time by a Turing machine **querying bits of real numbers**"

  Theorem [Folklore]: $\mathbf{P}/\mathrm{poly} = \mathbf{P}(\mathbb{R})$.

Computational meaning of regularity conditions

- Regular proofs = **finitely** many distinct subproofs

$$\boxed{\textit{regularity} \quad \approx \quad \textit{computability}, \ \textit{uniformity}}$$

- Weakly regular proofs = relaxation of regularity to represent real numbers

$$\boxed{\textbf{weak } \textit{regularity} \quad \approx \quad \textit{computability} \ + \ \textbf{query on bits of real numbers}}$$

... but since $\mathbf{P}(\mathbb{R}) = \mathbf{P}/\mathrm{poly}$ ...

$$\boxed{\textbf{weak } \textit{regularity} \quad \approx \quad \textit{non-uniformity}}$$

Computational meaning of regularity conditions

- Regular proofs $=$ **finitely** many distinct subproofs

$$\boxed{\textit{regularity} \quad \approx \quad \textit{computability}, \; \textit{uniformity}}$$

- Weakly regular proofs $=$ relaxation of regularity to represent real numbers

$$\boxed{\textbf{weak } \textit{regularity} \quad \approx \quad \textit{computability} \; + \; \textbf{query on bits of real numbers}}$$

... but since $\mathbf{P}(\mathbb{R}) = \mathbf{P}/\text{poly}$ ...

$$\boxed{\textbf{weak } \textit{regularity} \quad \approx \quad \textit{non-uniformity}}$$

Computational meaning of regularity conditions

- Regular proofs = **finitely** many distinct subproofs

$$\boxed{\textit{regularity} \quad \approx \quad \textit{computability}\,,\ \textit{uniformity}}$$

- Weakly regular proofs = relaxation of regularity to represent real numbers

$$\boxed{\textbf{weak}\ \textit{regularity} \quad \approx \quad \textit{computability} \ + \ \textbf{query on bits of real numbers}}$$

... but since $\mathbf{P}(\mathbb{R}) = \mathbf{P}/\mathrm{poly}$ ...

$$\boxed{\textbf{weak}\ \textit{regularity} \quad \approx \quad \textit{non-uniformity}}$$

Our characterisation results in a nutshell

Theorem:

	inductive	**non-wellfounded**
non-uniform	nuPLL_2	wrPLL_2^∞
uniform	PLL_2	rPLL_2^∞

Idea of the proof.

Our characterisation results in a nutshell

Theorem:

	inductive	non-wellfounded
P/poly	nuPLL_2	wrPLL_2^∞
P	PLL_2	rPLL_2^∞

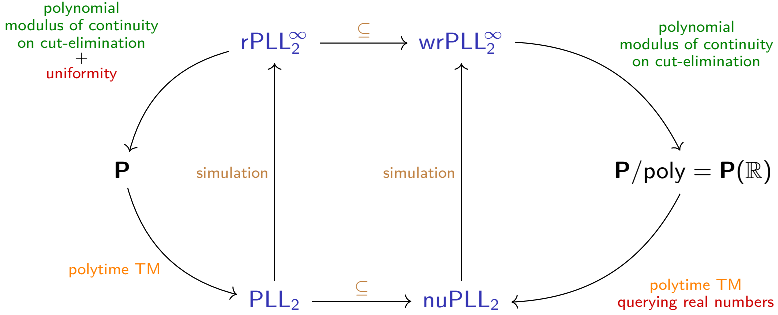
Idea of the proof.

Our characterisation results in a nutshell

Theorem:

	inductive	non-wellfounded
$\mathbf{P}/\mathrm{poly}$	nuPLL_2	wrPLL_2^∞
$\mathbf{P}$	PLL_2	rPLL_2^∞

Idea of the proof.



Conclusion and future work

- **To sum up:**
  - We introduced two non-wellfounded proof systems rPLL^∞ and wrPLL^∞
  - We showed that the second-order extensions of rPLL^∞ and wrPLL^∞ characterise, respectively, **P** and **P**/poly

- **Ongoing work:**

	inductive	**non-wellfounded**
P/poly	nuPLL_2	wrPLL_2^∞
P	PLL_2	rPLL_2^∞

Conclusion and future work

- **To sum up:**
  - We introduced two non-wellfounded proof systems rPLL^∞ and wrPLL^∞
  - We showed that the second-order extensions of rPLL^∞ and wrPLL^∞ characterise, respectively, $\mathbf{P}$ and $\mathbf{P}/\mathsf{poly}$

- **Ongoing work:**

	inductive	**non-wellfounded**
$\mathbf{P}/\mathsf{poly}$	nuPLL_2	wrPLL_2^∞
$\mathbf{P}$	PLL_2	rPLL_2^∞

Conclusion and future work

- **To sum up:**
  - We introduced two non-wellfounded proof systems rPLL^∞ and wrPLL^∞
  - We showed that the second-order extensions of rPLL^∞ and wrPLL^∞ characterise, respectively, $\mathbf{P}$ and $\mathbf{P}/\mathrm{poly}$

- **Ongoing work:**

	inductive	**non-wellfounded**
$\mathbf{L}/\mathrm{poly}$	nuPLL	wrPLL^∞
$\mathbf{L}$	PLL	rPLL^∞

...i.e., we restate in a non-wellfounded setting the results in [Maz15, MT15].

Thank you! Questions?

References:

- Stephen J. Bellantoni and Stephen A. Cook, *A new recursion-theoretic characterization of the polytime functions*, Comput. Complex. **2** (1992), 97–110.
- David Baelde, Amina Doumane, and Alexis Saurin, *Infinitary proof theory: the multiplicative additive case*, CSL, 2016.
- David Baelde and Dale Miller, *Least and greatest fixed points in linear logic*, LPAR, 2007.
- Gianluca Curzi and Anupam Das, *Cyclic implicit complexity*, LICS (Christel Baier and Dana Fisman, eds.), 2022.
- ———, *Computational expressivity of (circular) proofs with fixed points*, LICS, 2023.
- ———, *Non-uniform complexity via non-wellfounded proofs*, CSL (Bartek Klin and Elaine Pimentel, eds.), 2023.
- Vincent Danos and Jean-Baptiste Joinet, *Linear logic and elementary time*, Inf. Comput. **183** (2003), no. 1, 123–137.
- Jean-Yves Girard, *Linear logic*, Theor. Comput. Sci. (1987).
- Yves Lafont, *Soft linear logic and polynomial time*, Theor. Comput. Sci. **318** (2004), no. 1-2, 163–180.
- Damiano Mazza, *Simple parsimonious types and logarithmic space*, CSL (Stephan Kreutzer, ed.), 2015.
- Damiano Mazza and Kazushige Terui, *Parsimonious types and non-uniform computation*, ICALP, 2015.
- Alexis Saurin, *A linear perspective on cut-elimination for non-wellfounded sequent calculi with least and greatest fixed points (extended version)*, 2023.
- Terese, *Term rewriting systems*, Cambridge tracts in theoretical computer science, 2003.

Appendix

Finite expandability

- Finitely expandable proof = any branch has finitely many *cut* and *abs* rules

  Example:

$$\mathit{cut}\ \dfrac{\mathit{ax}\ \dfrac{}{A^{\perp}, A} \qquad \mathit{cut}\ \dfrac{\mathit{ax}\ \dfrac{}{A^{\perp}, A} \qquad \mathit{cut}\ \dfrac{\vdots}{\Gamma, A}}{\Gamma, A}}{\Gamma, A} \qquad\qquad \mathit{abs}\ \dfrac{\mathit{abs}\ \dfrac{\mathit{abs}\ \dfrac{\vdots}{A, A, ?A}}{A, ?A}}{?A}$$

- **Theorem:** **decomposition** for finitely expandable and progressing proofs

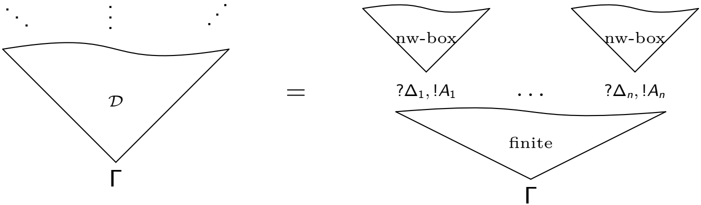
Finite expandability

- Finitely expandable proof = any branch has finitely many *cut* and *abs* rules

  Example:

$$\mathit{cut}\,\dfrac{\mathit{ax}\,\dfrac{}{A^{\perp}, A} \qquad \mathit{cut}\,\dfrac{\mathit{ax}\,\dfrac{}{A^{\perp}, A} \quad \mathit{cut}\,\dfrac{\vdots}{\Gamma, A}}{\Gamma, A}}{\Gamma, A} \qquad\qquad \mathit{abs}\,\dfrac{\mathit{abs}\,\dfrac{\mathit{abs}\,\dfrac{\vdots}{A, A, ?A}}{A, ?A}}{?A}$$

- **Theorem: decomposition** for finitely expandable and progressing proofs



Cut-elimination rules for non-wellfounded parsimonious logic

- **Cut-elimination rules** for the exponential modalities ! and ?:

$$
\dfrac{\dfrac{\Gamma, A \quad ?\Gamma, !A}{?\Gamma, !A}\,\mathsf{cp} \qquad \dfrac{A^\perp, \Delta, B \quad ?A^\perp, ?\Delta, !B}{?A^\perp, ?\Delta, !B}\,\mathsf{cp}}{?\Gamma, ?\Delta, !B}\,\mathsf{cut}
\rightsquigarrow
\dfrac{\dfrac{\Gamma, A \quad A^\perp, \Delta, B}{\Gamma, \Delta, B}\,\mathsf{cut} \qquad \dfrac{?\Gamma, !A \quad ?A^\perp, ?\Delta, !B}{?\Gamma, ?\Delta, !B}\,\mathsf{cut}}{?\Gamma, ?\Delta, !B}\,\mathsf{cp}
$$

$$
\dfrac{\dfrac{\Gamma, A \quad ?\Gamma, !A}{?\Gamma, !A}\,\mathsf{cp} \qquad \dfrac{\Delta}{\Delta, ?A^\perp}\,\mathsf{w}}{?\Gamma, \Delta}\,\mathsf{cut}
\rightsquigarrow
\dfrac{\Delta}{\overline{?\Gamma, \Delta}}\,\mathsf{w}
$$

$$
\dfrac{\dfrac{\Gamma, A \quad ?\Gamma, !A}{?\Gamma, !A}\,\mathsf{cp} \qquad \dfrac{\Delta, A^\perp, ?A^\perp}{\Delta, ?A^\perp}\,\mathsf{abs}}{?\Gamma, \Delta}\,\mathsf{cut}
\rightsquigarrow
\dfrac{\dfrac{\Gamma, A \qquad \dfrac{?\Gamma, !A \quad \Delta, A^\perp, ?A^\perp}{?\Gamma, \Delta, A^\perp}\,\mathsf{cut}}{\Gamma, ?\Gamma, \Delta}\,\mathsf{cut}}{\overline{?\Gamma, \Delta}}\,\mathsf{abs}
$$

Cut-elimination rules preserve progressing, (weak) regularity, and finite expandability conditions

Our domain-theoretic approach

Starting from non-wellfonded proof $\mathcal{D}$:

- Special **infinitary rewriting strategies** σ that induce continuous functions over domains of (partially defined) non-wellfounded proofs
- **Productivity:** If $\mathcal{D}$ is progressing non-wellfounded proof then $f_\sigma(\mathcal{D})$ is (cut-free and) totally defined
- f_σ **preserves** progressing and (weak) regularity conditions.

Our domain-theoretic approach

Starting from non-wellfonded proof $\mathcal{D}$:

- Special **infinitary rewriting strategies** σ that induce continuous functions over domains of (partially defined) non-wellfounded proofs
- **Productivity:** If $\mathcal{D}$ is progressing non-wellfounded proof then $f_\sigma(\mathcal{D})$ is (cut-free and) totally defined
- f_σ **preserves** progressing and (weak) regularity conditions.

Our domain-theoretic approach

Starting from non-wellfonded proof $\mathcal{D}$:

- Special **infinitary rewriting strategies** σ that induce continuous functions over domains of (partially defined) non-wellfounded proofs
- **Productivity:** If $\mathcal{D}$ is progressing non-wellfounded proof then $f_\sigma(\mathcal{D})$ is (cut-free and) totally defined
- f_σ **preserves** progressing and (weak) regularity conditions.

Approximating non-wellfounded proofs

(1) A new rule: the **hypothesis**

$$hyp \, \frac{}{\Gamma}$$

(2) Open proof = non-wellfounded proof that might contain *hyp*

(3) Normal open proof = proofs where *cut* rules are irreducible
(e.g., cut between hypotheses)

(4) $\mathsf{oPLL}^{\infty}(\Gamma)$ = domain of open proofs with endsequent Γ:

Approximating non-wellfounded proofs

(1) A new rule: the **hypothesis**

$$\mathit{hyp}\ \frac{\quad}{\Gamma}$$

(2) Open proof = non-wellfounded proof that might contain *hyp*

(3) Normal open proof = proofs where *cut* rules are irreducible
(e.g., cut between hypotheses)

(4) $\mathsf{oPLL}^{\infty}(\Gamma)$ = domain of open proofs with endsequent Γ:

Approximating non-wellfounded proofs

(1) A new rule: the **hypothesis**

$$hyp\ \frac{}{\Gamma}$$

(2) Open proof = non-wellfounded proof that might contain *hyp*

(3) Normal open proof = proofs where *cut* rules are irreducible
(e.g., cut between hypotheses)

(4) $\mathrm{oPLL}^{\infty}(\Gamma)$ = domain of open proofs with endsequent Γ:

Approximating non-wellfounded proofs

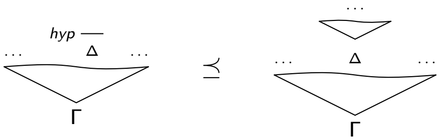
(1) A new rule: the **hypothesis**

$$\mathit{hyp}\ \frac{}{\Gamma}$$

(2) Open proof = non-wellfounded proof that might contain *hyp*

(3) Normal open proof = proofs where *cut* rules are irreducible
(e.g., cut between hypotheses)

(4) $\mathrm{oPLL}^{\infty}(\Gamma)$ = domain of open proofs with endsequent Γ:



Approximating non-wellfounded proofs

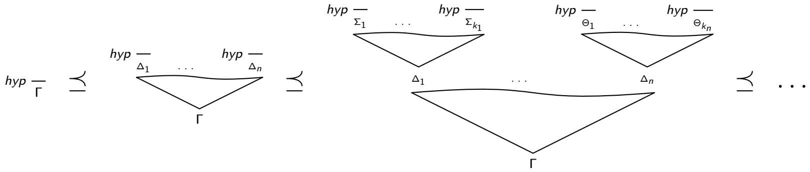
(1) A new rule: the **hypothesis**

$$\mathit{hyp}\ \frac{}{\Gamma}$$

(2) Open proof = non-wellfounded proof that might contain *hyp*

(3) Normal open proof = proofs where *cut* rules are irreducible
(e.g., cut between hypotheses)

(4) $\mathsf{oPLL}^{\infty}(\Gamma)$ = domain of open proofs with endsequent Γ:



Infinitary cut-elimination strategies

- Infinitary cut-elimination strategy (ices) := family $\sigma = (\sigma_{\mathcal{D}})_{\mathcal{D} \in \mathsf{oPLL}^\infty}$ where each $\sigma_{\mathcal{D}}$ is a countable sequence of proofs such that:

$$\mathcal{D} = \sigma_{\mathcal{D}}(0) \rightsquigarrow \sigma_{\mathcal{D}}(1) \rightsquigarrow \ldots \rightsquigarrow \sigma_{\mathcal{D}}(n) \rightsquigarrow \ldots$$

- Given an ices σ we define $f_\sigma : \mathsf{oPLL}^\infty(\Gamma) \to \mathsf{oPLL}^\infty(\Gamma)$ as

$$f_\sigma(\mathcal{D}) := \bigsqcup_{i=0}^{\ell(\sigma_{\mathcal{D}})} \mathsf{cf}(\sigma_{\mathcal{D}}(i))$$

where $\mathsf{cf}(\mathcal{D}_i)$ is the greatest cut-free approximation of $\mathcal{D}_i$ (w.r.t. $\preceq$)

- A ices σ is:
  - Maximal if, for any finite open proof, $\ell(\sigma_{\mathcal{D}})$ (is finite and) is normal
  - (Scott-)continuous is f_σ is

- **Maximal continuous infinitary cut-elimination strategies** (mc-ices).

Infinitary cut-elimination strategies

- Infinitary cut-elimination strategy (ices) := family $\sigma = (\sigma_{\mathcal{D}})_{\mathcal{D} \in \mathsf{oPLL}^{\infty}}$ where each $\sigma_{\mathcal{D}}$ is a countable sequence of proofs such that:

$$\mathcal{D} = \sigma_{\mathcal{D}}(0) \rightsquigarrow \sigma_{\mathcal{D}}(1) \rightsquigarrow \ldots \rightsquigarrow \sigma_{\mathcal{D}}(n) \rightsquigarrow \ldots$$

- Given an ices σ we define $f_{\sigma} \colon \mathsf{oPLL}^{\infty}(\Gamma) \to \mathsf{oPLL}^{\infty}(\Gamma)$ as

$$f_{\sigma}(\mathcal{D}) := \bigsqcup_{i=0}^{\ell(\sigma_{\mathcal{D}})} \mathsf{cf}(\sigma_{\mathcal{D}}(i))$$

where $\mathsf{cf}(\mathcal{D}_i)$ is the greatest cut-free approximation of $\mathcal{D}_i$ (w.r.t. $\preceq$)

- A ices σ is:
  - Maximal if, for any finite open proof, $\ell(\sigma_{\mathcal{D}})$ (is finite and) is normal
  - (Scott-)continuous is f_{σ} is

- **Maximal continuous infinitary cut-elimination strategies** (mc-ices).

Infinitary cut-elimination strategies

- Infinitary cut-elimination strategy (ices) := family $\sigma = (\sigma_{\mathcal{D}})_{\mathcal{D} \in \mathsf{oPLL}^\infty}$ where each $\sigma_{\mathcal{D}}$ is a countable sequence of proofs such that:

$$\mathcal{D} = \sigma_{\mathcal{D}}(0) \rightsquigarrow \sigma_{\mathcal{D}}(1) \rightsquigarrow \ldots \rightsquigarrow \sigma_{\mathcal{D}}(n) \rightsquigarrow \ldots$$

- Given an ices σ we define $f_\sigma \colon \mathsf{oPLL}^\infty(\Gamma) \to \mathsf{oPLL}^\infty(\Gamma)$ as

$$f_\sigma(\mathcal{D}) := \bigsqcup_{i=0}^{\ell(\sigma_{\mathcal{D}})} \mathsf{cf}(\sigma_{\mathcal{D}}(i))$$

  where $\mathsf{cf}(\mathcal{D}_i)$ is the greatest cut-free approximation of $\mathcal{D}_i$ (w.r.t. $\preceq$)

- A ices σ is:
  - Maximal if, for any finite open proof, $\ell(\sigma_{\mathcal{D}})$ (is finite and) is normal
  - (Scott-)continuous is f_σ is

- **Maximal continuous infinitary cut-elimination strategies** (mc-ices).

Infinitary cut-elimination strategies

- Infinitary cut-elimination strategy (ices) := family $\sigma = (\sigma_{\mathcal{D}})_{\mathcal{D} \in \mathsf{oPLL}^{\infty}}$ where each $\sigma_{\mathcal{D}}$ is a countable sequence of proofs such that:

$$\mathcal{D} = \sigma_{\mathcal{D}}(0) \rightsquigarrow \sigma_{\mathcal{D}}(1) \rightsquigarrow \ldots \rightsquigarrow \sigma_{\mathcal{D}}(n) \rightsquigarrow \ldots$$

- Given an ices σ we define $f_{\sigma} \colon \mathsf{oPLL}^{\infty}(\Gamma) \to \mathsf{oPLL}^{\infty}(\Gamma)$ as

$$f_{\sigma}(\mathcal{D}) := \bigsqcup_{i=0}^{\ell(\sigma_{\mathcal{D}})} \mathsf{cf}(\sigma_{\mathcal{D}}(i))$$

where $\mathsf{cf}(\mathcal{D}_i)$ is the greatest cut-free approximation of $\mathcal{D}_i$ (w.r.t. $\preceq$)

- A ices σ is:
  - Maximal if, for any finite open proof, $\ell(\sigma_{\mathcal{D}})$ (is finite and) is normal
  - (Scott-)continuous is f_{σ} is

- **Maximal continuous infinitary cut-elimination strategies** (mc-ices).

Continuous cut-elimination theorem

- **Existence of** mc-ices**:** intuitively, always apply a cut-elimination step to the leftmost reducible *cut* rule with minimal height.
- **Confluence:** if σ and σ' are mc-ices, then $f_\sigma = f_{\sigma'}$
- **Theorem (Continuous cut-elimination):** Given σ a mc-ices:
  (1) $\mathcal{D}$ is progressing then $f_\sigma(\mathcal{D})$ is *hyp*-free (*productivity*)
  (2) f_σ preserves progressing and finite expandability
  (3) If $\mathcal{D} \in \mathsf{wrPLL}^\infty$ then $f_\mathcal{D}(\mathcal{D}) \in \mathsf{wrPLL}^\infty$
  (4) If $\mathcal{D} \in \mathsf{rPLL}^\infty$ then $f_\mathcal{D}(\mathcal{D}) \in \mathsf{rPLL}^\infty$

Continuous cut-elimination theorem

- **Existence of** mc-ices**:** intuitively, always apply a cut-elimination step to the leftmost reducible *cut* rule with minimal height.
- **Confluence:** if σ and σ' are mc-ices, then $f_\sigma = f_{\sigma'}$
- **Theorem (Continuous cut-elimination):** Given σ a mc-ices:

  (1) $\mathcal{D}$ is progressing then $f_\sigma(\mathcal{D})$ is *hyp*-free (*productivity*)

  (2) f_σ preserves progressing and finite expandability

  (3) If $\mathcal{D} \in \mathsf{wrPLL}^\infty$ then $f_\mathcal{D}(\mathcal{D}) \in \mathsf{wrPLL}^\infty$

  (4) If $\mathcal{D} \in \mathsf{rPLL}^\infty$ then $f_\mathcal{D}(\mathcal{D}) \in \mathsf{rPLL}^\infty$

Productivity and preservation of progressing condition

Theorem:

(1) $\mathcal{D}$ is progressing then $f_\sigma(\mathcal{D})$ is *hyp*-free (*productivity*)

(2) f_σ preserves progressing and finite expandability

Proof idea.

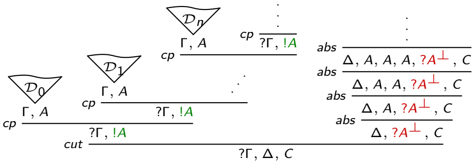
Productivity and preservation of progressing condition

Theorem:

(1) $\mathcal{D}$ is progressing then $f_\sigma(\mathcal{D})$ is *hyp*-free (*productivity*)

(2) f_σ preserves progressing and finite expandability

Proof idea.



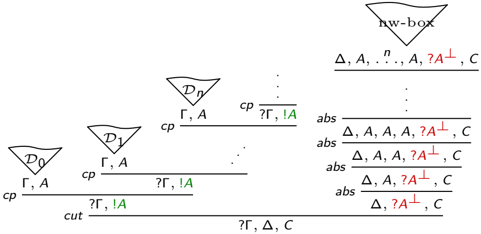
Productivity and preservation of progressing condition

Theorem:

(1) $\mathcal{D}$ is progressing then $f_\sigma(\mathcal{D})$ is *hyp*-free (*productivity*)

(2) f_σ preserves progressing and finite expandability

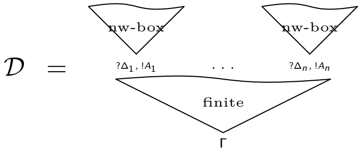
Proof idea.



Preservation of (weak) regularity

Theorem: If $\mathcal{D} \in \mathsf{wrPLL}^\infty$ then $f_{\mathcal{D}}(\mathcal{D}) \in \mathsf{wrPLL}^\infty$ (similarly for rPLL^∞)

Proof idea. We use decomposition:



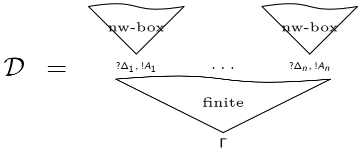
We define a transfinite cut-elimination sequence preserving (weak) regularity by induction on the "nesting" of non-wellfounded boxes:

We compress the transfinite sequence to an ω-long one [Sau23, Ter03]

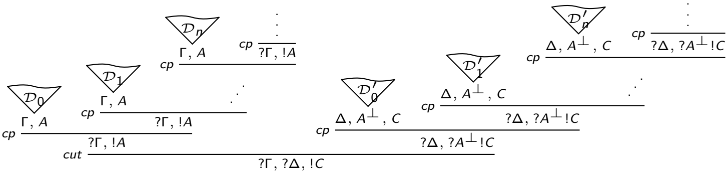
Preservation of (weak) regularity

Theorem: If $\mathcal{D} \in \mathsf{wrPLL}^\infty$ then $f_\mathcal{D}(\mathcal{D}) \in \mathsf{wrPLL}^\infty$ (similarly for rPLL^∞)

Proof idea. We use decomposition:



We define a transfinite cut-elimination sequence preserving (weak) regularity by induction on the "nesting" of non-wellfounded boxes:

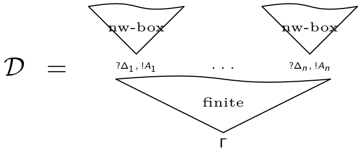


We compress the transfinite sequence to an ω-long one [Sau23, Ter03]

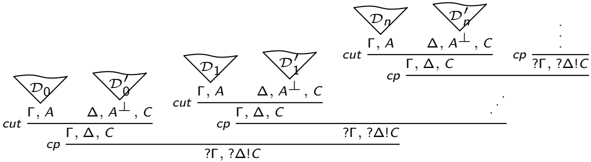
Preservation of (weak) regularity

Theorem: If $\mathcal{D} \in \mathsf{wrPLL}^\infty$ then $f_{\mathcal{D}}(\mathcal{D}) \in \mathsf{wrPLL}^\infty$ (similarly for rPLL^∞)

Proof idea. We use decomposition:



We define a transfinite cut-elimination sequence preserving (weak) regularity by induction on the "nesting" of non-wellfounded boxes:

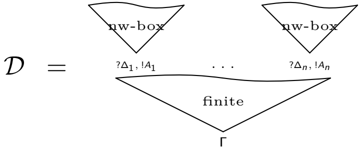


We compress the transfinite sequence to an ω-long one [Sau23, Ter03]

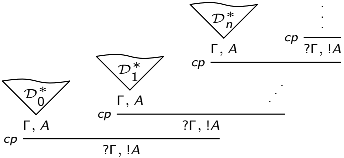
Preservation of (weak) regularity

Theorem: If $\mathcal{D} \in \mathsf{wrPLL}^\infty$ then $f_{\mathcal{D}}(\mathcal{D}) \in \mathsf{wrPLL}^\infty$ (similarly for rPLL^∞)

Proof idea. We use decomposition:



We define a transfinite cut-elimination sequence preserving (weak) regularity by induction on the “nesting” of non-wellfounded boxes:



We compress the transfinite sequence to an ω-long one [Sau23, Ter03]

5 Non-uniform complexity classes

Non-uniform complexity classes

- **FP** $=$ class of functions computable in polynomial time on a Turing machine.
- **FP**/poly is an extension of **FP** that intuitively has access to a 'small' amount of *advice*, determined only by the length of the input.
- **FP**/poly $=$ class of functions $f(\vec{x})$ for which there exists some strings $\alpha_{\vec{n}} \in \{0,1\}^*$ and a function $f'(x, \vec{x}) \in$ **FP** with:
  - $|\alpha_{\vec{n}}|$ is polynomial in $\vec{n}$.
  - $f(\vec{x}) = f'(\alpha_{|\vec{x}|}, \vec{x})$.
- Note, in particular, that **FP**/poly admits undecidable problems. E.g. the function $f(x) = 1$ just if $|x|$ is the code of a halting Turing machine (and 0 otherwise) is in **FP**/poly. Indeed, the point of the class **FP**/poly is to rather characterise a more non-uniform notion of computation.
- **Theorem**: $f(\vec{x}) \in$ **FP**/poly iff there are poly-size circuits computing $f(\vec{x})$.

- The class $\mathbf{FP}(\mathbb{R})$ consists of just the functions computable in polynomial time by a Turing machine with access to oracles from:

$$\mathbb{R} := \{f(x) : \mathbb{N} \to \{0, 1\} \mid |x| = |y| \implies f(x) = f(y)\}$$

- Note that the notation $\mathbb{R}$ is suggestive here, since its elements are essentially maps from lengths/positions to Booleans, and so may be identified with Boolean streams.

- **Theorem** [Folklore]: $\mathbf{FP}/\mathrm{poly} = \mathbf{FP}(\mathbb{R})$.