Abstractions of Sequences, Functions and Operators

Louis Rustenholz[1,3], in collaboration with
Pedro López-García[2,3] and Manuel V. Hermenegildo[1,3]

[1]Universidad Politécnica de Madrid (UPM), Spain
[2]Spanish Council for Scientific Research (CSIC), Spain
[3]IMDEA Software Institute, Spain

CSV, June 5th, 2025
Università Ca' Foscari, Venezia

Introduction

Introduction

- Infer information about **mathematical functions** given by a **recursive definition**, i.e. by an operator/equation.

→ Role of **order theory and abstract interpretation**.

- Share some order theory facts, Galois connections, ... discovered along the way.

- Present (domain specific?) abstract domains built using **functions as the basic object**.

Intuition

Idea to be explored

Functions $X \to Y$ may be simpler objects than arbitrary sets of points $\mathcal{P}(X \times Y)$ (relations).

Of course, not in general, but...

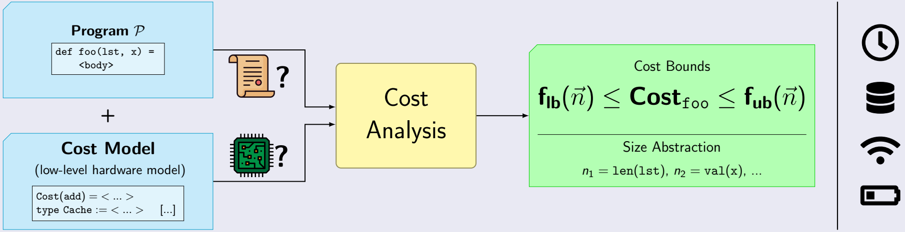
- Can be the case for families of functions we care to approximate
- ➔ Exploit local regularity
- ➔ Relate definition with that of simpler functions
- ➔ ...

Analysis viewpoint

We are interested in **numerical functions**, **constructed**/defined recursively **by an operator** $\Phi \in \big((X \to L) \to (X \to L)\big)$, **i.e. by an equation**.

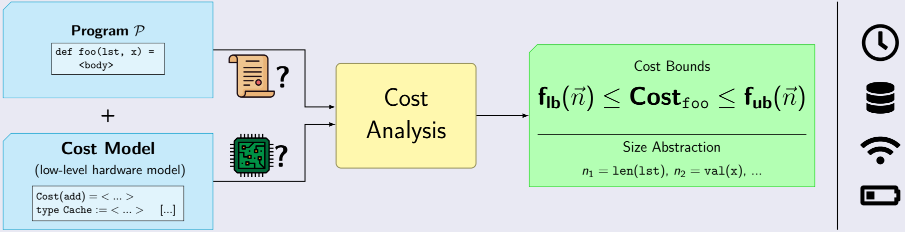
Origin and Motivation: Cost Analysis

Cost Analysis: Bounds on Resource Consumption



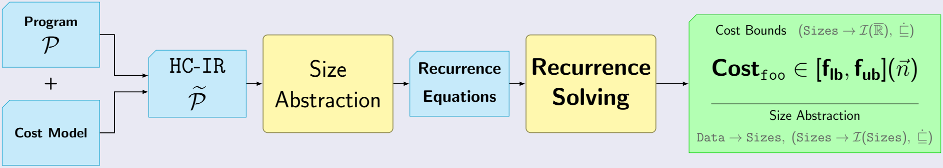
Origin and Motivation: Cost Analysis

Cost Analysis: Bounds on Resource Consumption



- Pipeline implemented in CiaoPP (and other analysers).

Recurrence-based cost analysis



Intuition

Analysis viewpoint

We are interested in **numerical functions**, **constructed**/defined recursively **by an operator** $\Phi \in \big((X \to L) \to (X \to L)\big)$, **i.e. by an equation**.

- **Function defined recursively** in a declarative language,
- **Input/output of a basic block** in an imperative language,
- Cost function obtained as the **solution of a recurrence equation**,
- Solution of a **differential equation**, ...

Intuition

Analysis viewpoint

We are interested in **numerical functions**, **constructed**/defined recursively **by an operator** $\Phi \in \big((X \to L) \to (X \to L)\big)$, **i.e. by an equation**.

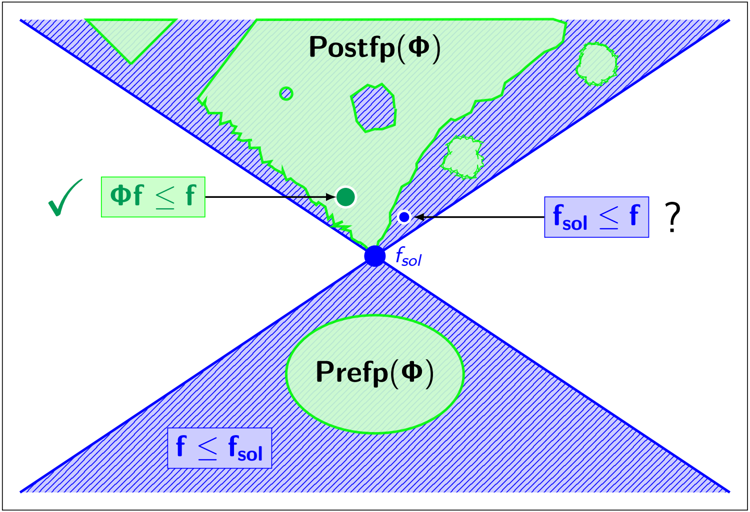
- **Function defined recursively** in a declarative language,
- **Input/output of a basic block** in an imperative language,
- Cost function obtained as the **solution of a recurrence equation**,
- Solution of a **differential equation**, ...

➔ Can we build abstract lattices of functions, and abstract these operators/equations $\big(\Phi \rightsquigarrow \Phi^\sharp\big)$ to **produce bounds** $f_{sol} \mathrel{\dot\sqsubseteq} \hat{f}$ by **abstract Kleene iteration**?

Order theory framework of equations [SAS24]

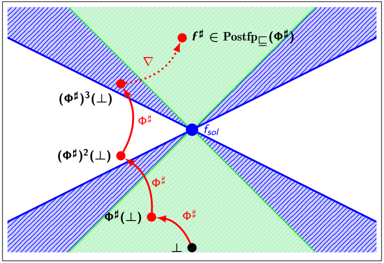
Equations $\leftrightarrow$ Operators Solutions $\leftrightarrow$ Fixpoints Bounds $\leftarrow$ Pre/Postfixp

✓ $\Phi \mathbf{f} \leq \mathbf{f}$

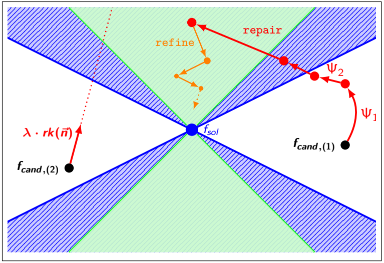


$\mathbf{f_{sol}} \leq \mathbf{f}$?

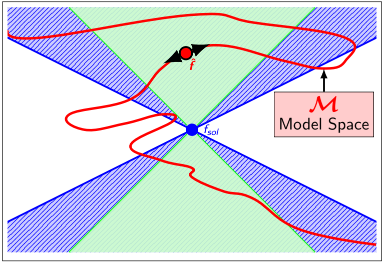
Equation solving as pre/postfixpoint search [SAS24]



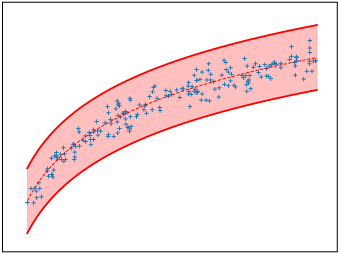
Abstract Interpretation



Geometry-based expression **Repair**

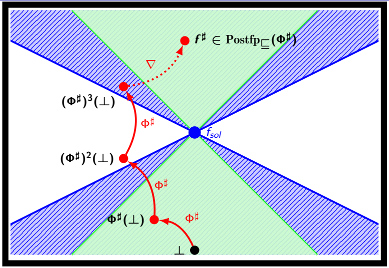


Search on subvarieties: **Templates**, ∀-elim

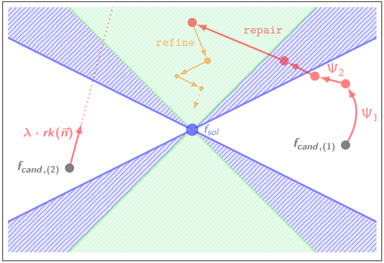


Constrained **Optimisation**,
with *provability constraints*

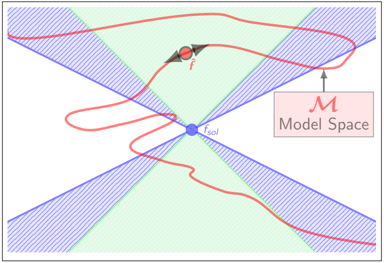
Equation solving as pre/postfixpoint search [SAS24]



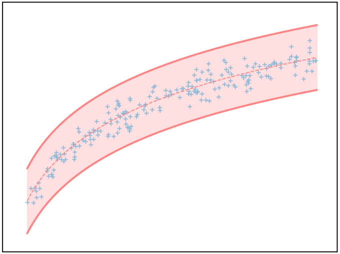
Abstract Interpretation



Geometry-based expression **Repair**



Search on subvarieties: **Templates**, ∀-elim



Constrained **Optimisation**,
with *provability constraints*

(Abstract) Lattices of Functions and Galois Connections between them

Lattices of Functions

Pointwise lattice structure

For X a set and $(L, \leq, \vee, \wedge)$ a complete lattice, the set of functions $X \to L$ has a very natural lattice structure $(X \to L, \dot{\leq}, \dot{\vee}, \dot{\wedge})$, with

$$f \mathrel{\dot{\leq}} g \overset{\Delta}{\iff} \forall x \in X,\ f(x) \leq g(x),$$

$$f \mathrel{\dot{\vee}} g \triangleq (x \mapsto f(x) \vee g(x)), \qquad \dot{\bot} \triangleq (x \mapsto \bot),$$

$$f \mathrel{\dot{\wedge}} g \triangleq (x \mapsto f(x) \wedge g(x)), \qquad \dot{\top} \triangleq (x \mapsto \top).$$

Example (Numerical functions)

In $(\mathcal{D} \to \overline{\mathbb{R}}, \dot{\leq})$, join and meet are pointwise max/min. $\dot{\bot} = x \mapsto -\infty$.

Example (Set-valued functions)

In $(\mathcal{D} \to \mathcal{I}(\mathbb{R}), \dot{\sqsubseteq}_{\mathcal{I}})$, join/meet are pointwise union/intersection.

We are interested in objects which **can be viewed as functions**, constructed/defined as **fixpoints** of some **operator** $\Phi \in \big((X \to L) \to (X \to L)\big)$, which is **monotone** for such an order on functions (to apply Knaster-Tarski).

- **Function defined recursively** in a declarative language,
- **Input/output of a basic block** in an imperative language,
- Cost function obtained as the **solution of a recurrence equation**,
- Solution of a **differential equation**, ...

➔ Can we abstract these lattices $\big(D \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} D^\sharp\big)$ and these operators $\big(\Phi \rightsquigarrow \Phi^\sharp\big)$ to produce bounds $f_{sol} \mathrel{\dot{\sqsubseteq}} \hat{f} \in \mathrm{Postfp}(\Phi)$ by **abstract Kleene iteration**?

Overview: a landscape of abstractions

From symbolic functions to manageable numerical functions

Symbolic functions	Numerical-set functions	Abstract numerical-set functions	Numerical functions	Simple numerical functions ($\mathfrak{B} \subseteq (\mathbb{N}^k \to \overline{\mathbb{R}})$)

$$\begin{tikzcd}
\Phi_{\mathtt{Data}} \arrow[r, shift left, "\text{size abstraction}"] & \Phi_{\mathcal{P}} \arrow[l, shift left] \arrow[r, shift left, "\text{codomain abstraction}"] & \Phi_{\mathcal{I}} \arrow[l, shift left] \arrow[r, shift left, "\substack{\text{separability}\\(\leq\text{-monotonicity}),\\ \text{reductions, } \dots}"] & \Phi_{\mathbb{N}} \arrow[l, shift left] \arrow[r, shift left, "\mathfrak{B}\textbf{-bounds abstraction}"] & \Phi^{\sharp} \arrow[l, shift left]
\end{tikzcd}$$

$$\begin{tikzcd}
\mathtt{Data} \arrow[d] & \mathbb{N}^r \arrow[d] & \mathbb{N}^r \arrow[d] & \mathbb{N}^k \arrow[d] & \\
\mathcal{P}(\mathtt{Data} \times \mathbb{R}) & \mathcal{P}(\mathbb{N}^s \times \mathbb{R}) & \mathcal{I}(\mathbb{N})^s \times \mathcal{I}(\mathbb{R}) & \overline{\mathbb{R}} & D^{\sharp} = \mathcal{P}_{\uparrow}(\mathfrak{B})
\end{tikzcd}$$

Overview: a landscape of abstractions

From symbolic functions to manageable numerical functions

Symbolic functions	Numerical-set functions	Abstract numerical-set functions	Numerical functions	Simple numerical functions ($\mathfrak{B} \subseteq (\mathbb{N}^k \to \overline{\mathbb{R}})$)

$$\begin{tikzcd}
\Phi_{\texttt{Data}} \arrow[r, shift left, leftarrow, "\text{size abstraction}"] \arrow[r, shift right] & \Phi_{\mathcal{P}} \arrow[r, shift left, leftarrow, "\text{codomain abstraction}"] \arrow[r, shift right] & \Phi_{\mathcal{I}} \arrow[r, shift left, leftarrow, "\substack{\text{separability}\\(\leq\text{-monotonicity}),\\ \text{reductions, } \ldots}"] \arrow[r, shift right] \arrow[rr, dashed, bend right, shift right] & \Phi_{\mathbb{N}} \arrow[r, shift left, leftarrow, "\mathfrak{B}\textbf{-bounds abstraction}"] \arrow[r, shift right] & \Phi^{\sharp} \arrow[ll, dashed, bend left]
\end{tikzcd}$$

$$\begin{tikzcd}
\texttt{Data} \arrow[d] & \mathbb{N}^r \arrow[d] & \mathbb{N}^r \arrow[d] & \mathbb{N}^k \arrow[d, dashed] \\
\mathcal{P}(\texttt{Data} \times \mathbb{R}) & \mathcal{P}(\mathbb{N}^s \times \mathbb{R}) & \mathcal{I}(\mathbb{N})^s \times \mathcal{I}(\mathbb{R}) & \overline{\mathbb{R}}
\end{tikzcd}$$

$$D^{\sharp} = (\mathcal{P}_{\downarrow}(\mathfrak{B}) \times \mathcal{P}_{\uparrow}(\mathfrak{B}))^{s+1}$$

Simplified picture: intervals for numerically non-monotone systems, lb+ub, **systems** $\Phi \in \mathrm{End}\big(\prod_{\tau}(\tau_{in} \to \tau_{out})\big)$ vs single equation $\Phi \in \mathrm{End}(\tau_{in} \to \tau_{out})$, …

Constructing Galois connections in function space

Many ways to build connections on functions from connections on values.

Proposition (Codomain abstraction)

Let $(D, \leq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (D^\sharp, \sqsubseteq)$ *and* X *be a set. This lifts to* $(X \to D, \dot{\leq}) \underset{\dot{\alpha}}{\overset{\dot{\gamma}}{\leftrightarrows}} (X \to D^\sharp, \dot{\sqsubseteq})$*, with* $\dot{\alpha}(f) = \alpha \circ f$ *and* $\dot{\gamma}(f^\sharp) = \gamma \circ f^\sharp$.

Constructing Galois connections in function space

Many ways to build connections on functions from connections on values.

Proposition (Codomain abstraction)

Let $(D, \leq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (D^\sharp, \sqsubseteq)$ *and* X *be a set. This lifts to* $(X \to D, \dot{\leq}) \underset{\dot{\alpha}}{\overset{\dot{\gamma}}{\leftrightarrows}} (X \to D^\sharp, \dot{\sqsubseteq})$, *with* $\dot{\alpha}(f) = \alpha \circ f$ *and* $\dot{\gamma}(f^\sharp) = \gamma \circ f^\sharp$.

Definition (Endomorphisms are monotone endofunctions)

Let $(L, \sqsubseteq)$ be a partial order. $\mathrm{End}_{\sqsubseteq}(L) := \{f : L \to L \mid \forall x \sqsubseteq y,\ f(x) \sqsubseteq f(y)\}$.

Proposition (End-lifting)

Let $(D, \leq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (D^\sharp, \sqsubseteq)$. *This lifts to a Galois connection*

$$
\begin{aligned}
(\mathrm{End}_{\leq}(D), \dot{\leq}) &\underset{\vec{\alpha}}{\overset{\vec{\gamma}}{\leftrightarrows}} (\mathrm{End}_{\sqsubseteq}(D^\sharp), \dot{\sqsubseteq}) \\
f &\longmapsto \alpha \circ f \circ \gamma, \\
\gamma \circ f^\sharp \circ \alpha &\longleftarrow f^\sharp.
\end{aligned}
$$

$$
\begin{tikzcd}
D \arrow[r, "f"] & D \arrow[d, "\alpha"] \\
D^\sharp \arrow[u, "\gamma"] \arrow[r, dashed, "f^\sharp = \vec{\alpha}(f)"] & D^\sharp
\end{tikzcd}
$$

Corollary

This can be iterated to **operators** *in* $\mathrm{End}^2 \approx ((\cdot \to \cdot) \to (\cdot \to \cdot))$, *and beyond.*

From mappings in value space to Galois connections in function space

Theorem (Domain abstraction)

Let $m : X \to A$ be an **arbitrary mapping***, and $(L, \sqsubseteq)$ be a complete lattice. Then, there is a Galois connection*

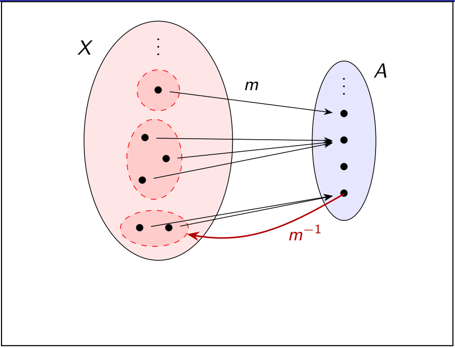
$$(X \to L, \dot{\sqsubseteq}) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (A \to L, \dot{\sqsubseteq})$$

$$f \longmapsto \left(a \mapsto \bigsqcup_{x \in m^{-1}(a)} f(x)\right)$$

$$\left(x \mapsto f^\sharp(m(x))\right) \longleftarrow f^\sharp,$$

which is an insertion for $m : X \twoheadrightarrow A$.

"$\alpha : f \mapsto \sqcup f \circ m^{-1}, \quad \gamma : f^\sharp \mapsto f^\sharp \circ m$"



Remark: Codomain abstraction is classical.
$(X \to D, \dot{\leq}) \leftrightarrows (X \to D^\sharp, \dot{\sqsubseteq})$

From mappings in value space to Galois connections in function space

Theorem (**Domain abstraction**)

Let $m : X \to A$ be an **arbitrary mapping***, and $(L, \sqsubseteq)$ be a complete lattice. Then, there is a Galois connection*

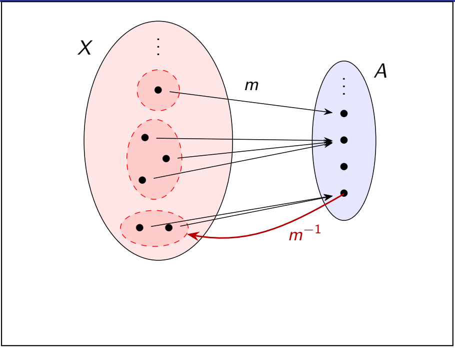
$$(X \to L, \dot{\sqsubseteq}) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (A \to L, \dot{\sqsubseteq})$$

$$f \longmapsto \left(a \mapsto \bigsqcup_{x \in m^{-1}(a)} f(x)\right)$$

$$\left(x \mapsto f^\sharp(m(x))\right) \longleftarrow f^\sharp,$$

which is an insertion for $m : X \twoheadrightarrow A$.

"$\alpha : f \mapsto \sqcup f \circ m^{-1}, \quad \gamma : f^\sharp \mapsto f^\sharp \circ m$"



Example (Size abstraction)

$$m : \mathtt{Data} \to \mathtt{Sizes}$$

e.g. `list-length` $\sqcup$ `tree-nbnodes` $\sqcup$ `int-value`...

Remark: Codomain abstraction is classical.

$$(X \to D, \dot{\leq}) \leftrightarrows (X \to D^\sharp, \dot{\sqsubseteq})$$

From mappings in value space to Galois connections in function space

Theorem (Domain abstraction)

Let $m : X \to A$ be an **arbitrary mapping**, *and $(L, \sqsubseteq)$ be a complete lattice. Then, there is a Galois connection*

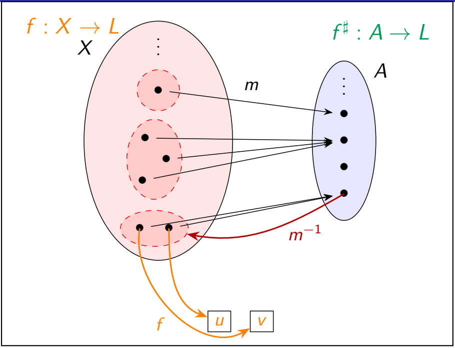
$$(X \to L, \dot{\sqsubseteq}) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (A \to L, \dot{\sqsubseteq})$$

$$f \longmapsto \Big(a \mapsto \bigsqcup_{x \in m^{-1}(a)} f(x)\Big)$$

$$\Big(x \mapsto f^\sharp(m(x))\Big) \longleftarrow f^\sharp,$$

which is an insertion for $m : X \twoheadrightarrow A$.

"$\alpha : f \mapsto \sqcup f \circ m^{-1}, \quad \gamma : f^\sharp \mapsto f^\sharp \circ m$"



Example (Size abstraction)

$$m : \texttt{Data} \to \texttt{Sizes}$$

e.g. `list-length` $\sqcup$ `tree-nbnodes` $\sqcup$ `int-value`...

Remark: Codomain abstraction is classical.

$$(X \to D, \dot{\leq}) \leftrightarrows (X \to D^\sharp, \dot{\sqsubseteq})$$

From mappings in value space to Galois connections in function space

Theorem (Domain abstraction)

Let $m : X \to A$ be an **arbitrary mapping**, *and $(L, \sqsubseteq)$ be a complete lattice.*
Then, there is a Galois connection

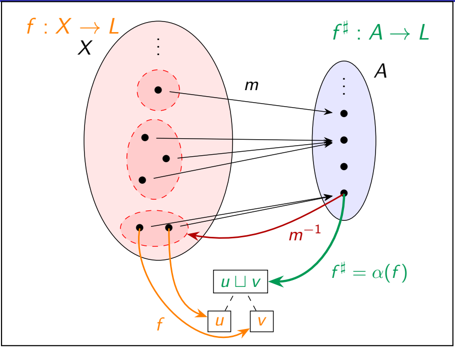
$$(X \to L, \dot{\sqsubseteq}) \xleftrightarrows[\alpha]{\gamma} (A \to L, \dot{\sqsubseteq})$$

$$f \longmapsto \Big(a \mapsto \bigsqcup_{x \in m^{-1}(a)} f(x)\Big)$$

$$\Big(x \mapsto f^\sharp(m(x))\Big) \longleftarrow f^\sharp,$$

which is an insertion for $m : X \twoheadrightarrow A$.

“ $\alpha : f \mapsto \sqcup f \circ m^{-1}, \quad \gamma : f^\sharp \mapsto f^\sharp \circ m$ ”



Example (Size abstraction)

$$m : \texttt{Data} \to \texttt{Sizes}$$

e.g. `list-length` $\sqcup$ `tree-nbnodes` $\sqcup$ `int-value`...

Remark: Codomain abstraction is classical.
$$(X \to D, \dot{\leq}) \leftrightarrows (X \to D^\sharp, \dot{\sqsubseteq})$$

From mappings in value space to Galois connections in function space

Theorem (Domain abstraction)

Let $m : X \to A$ be an **arbitrary mapping***, and $(L, \sqsubseteq)$ be a complete lattice. Then, there is a Galois connection*

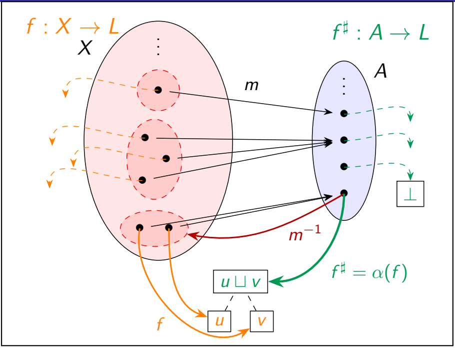
$$(X \to L, \dot{\sqsubseteq}) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (A \to L, \dot{\sqsubseteq})$$

$$f \longmapsto \Big(a \mapsto \bigsqcup_{x \in m^{-1}(a)} f(x)\Big)$$

$$\Big(x \mapsto f^\sharp(m(x))\Big) \longleftarrow f^\sharp,$$

which is an insertion for $m : X \twoheadrightarrow A$.

"$\alpha : f \mapsto \sqcup f \circ m^{-1}, \quad \gamma : f^\sharp \mapsto f^\sharp \circ m$"



Example (Size abstraction)

m : Data $\to$ Sizes

e.g. `list-length` $\sqcup$ `tree-nbnodes` $\sqcup$ `int-value`...

Remark: Codomain abstraction is classical.
$(X \to D, \dot{\leq}) \leftrightarrows (X \to D^\sharp, \dot{\sqsubseteq})$

From mappings in value space to Galois connections in function space

Theorem (**Domain abstraction**)

Let $m : X \to A$ be an **arbitrary mapping**, *and $(L, \sqsubseteq)$ be a complete lattice. Then, there is a Galois connection*

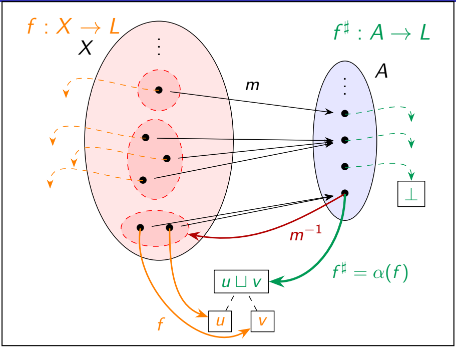
$$(X \to L, \dot{\sqsubseteq}) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (A \to L, \dot{\sqsubseteq})$$

$$f \longmapsto \left(a \mapsto \bigsqcup_{x \in m^{-1}(a)} f(x)\right)$$

$$\left(x \mapsto f^\sharp(m(x))\right) \longleftarrow f^\sharp,$$

which is an insertion for $m : X \twoheadrightarrow A$.

“$\alpha : f \mapsto \sqcup f \circ m^{-1}$, $\quad \gamma : f^\sharp \mapsto f^\sharp \circ m$”



Corollary (Powerset lifting)

$$(\mathcal{P}(X), \subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\mathcal{P}(A), \subseteq)$$

Corollary (For non-deterministic functions)

$$(X \to \mathcal{P}(X), \dot{\subseteq}) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (A \to \mathcal{P}(A), \dot{\subseteq})$$

(“$f \mapsto m^{-1} \circ f \circ m$ and $f^\sharp \mapsto m \circ f^\sharp \circ m^{-1}$”.)

Example (Size abstraction)

$$m : \texttt{Data} \to \texttt{Sizes}$$

e.g. `list-length` $\sqcup$ `tree-nbnodes` $\sqcup$ `int-value`...

- Can also combine such *size metrics*.
- Question of inference for a given program.
- ≈Dual approach: `Sized-Types` $\to \mathcal{P}(\texttt{Data})$.

Remark: Codomain abstraction is classical.
$(X \to D, \dot{\leq}) \leftrightarrows (X \to D^\sharp, \dot{\sqsubseteq})$

From mappings in value space to Galois connections in function space

Theorem (**Domain abstraction**)

Let $m : X \to A$ be an **arbitrary mapping**, *and $(L, \sqsubseteq)$ be a complete lattice.*
Then, there is a Galois connection

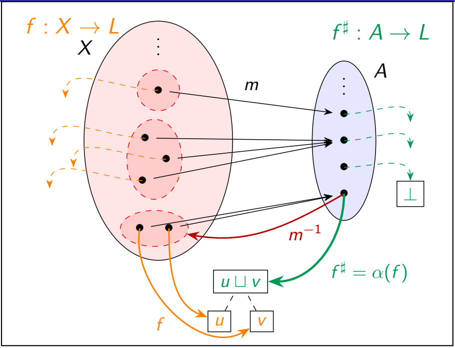
$$(X \to L, \dot\sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (A \to L, \dot\sqsubseteq)$$

$$f \longmapsto \Big(a \mapsto \bigsqcup_{x \in m^{-1}(a)} f(x)\Big)$$

$$\Big(x \mapsto f^\sharp(m(x))\Big) \longleftarrow f^\sharp,$$

which is an insertion for $m : X \twoheadrightarrow A$.

"$\alpha : f \mapsto \sqcup f \circ m^{-1}, \quad \gamma : f^\sharp \mapsto f^\sharp \circ m$"



Corollary (Powerset lifting)

$$(\mathcal{P}(X), \subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\mathcal{P}(A), \subseteq)$$

Corollary (For non-deterministic functions)

$$(X \to \mathcal{P}(X), \dot\subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (A \to \mathcal{P}(A), \dot\subseteq)$$

("$f \mapsto m^{-1} \circ f \circ m$ and $f^\sharp \mapsto m \circ f^\sharp \circ m^{-1}$".)

Example (Dimensionality reduction)

$$m : \mathbb{N}^3 \to \mathbb{N}^2$$
$$(x, y, z) \mapsto (x + y, z)$$

Important questions:
- Inference for a given program Φ,
- Computation of $\Phi^\sharp$, ...

Remark: Codomain abstraction is classical.
$$(X \to D, \dot\leq) \leftrightarrows (X \to D^\sharp, \dot\sqsubseteq)$$

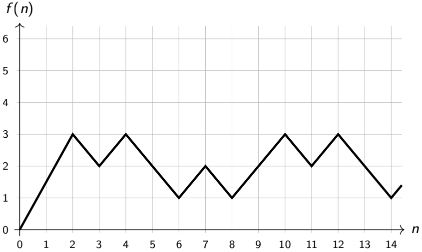
$\mathfrak{B}$-bounds — abstracting functions with simpler functions

A class of abstract domains for functions

For $\mathfrak{B} \subseteq (\mathcal{D} \to \overline{\mathbb{R}})$, let $D^\sharp := (\mathcal{P}(\mathfrak{B}), \supseteq)$ be the domain of $\mathfrak{B}$-upper bounds.

$(\mathcal{D} \to \overline{\mathbb{R}}, \dot{\leq}) \underset{\alpha_{\mathfrak{B}}}{\overset{\gamma_{\mathfrak{B}}}{\leftrightarrows}} (\mathcal{P}(\mathfrak{B}), \supseteq)$. For $f : \mathcal{D} \to \overline{\mathbb{R}}$,

$$\alpha_{\mathfrak{B}}(f) = F^\sharp_{ub} \longleftrightarrow \bigwedge_{f^\sharp_i \in F^\sharp_{ub}} f \,\dot{\leq}\, f^\sharp_i.$$



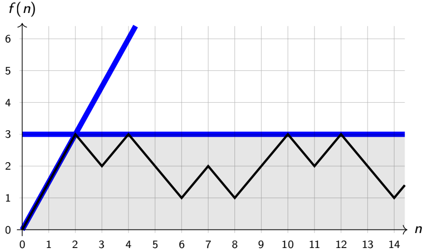
$\mathfrak{B}$-bounds — abstracting functions with simpler functions

A class of abstract domains for functions

For $\mathfrak{B} \subseteq (\mathcal{D} \to \overline{\mathbb{R}})$, let $D^\sharp := (\mathcal{P}(\mathfrak{B}), \supseteq)$ be the domain of $\mathfrak{B}$-upper bounds.

$(\mathcal{D} \to \overline{\mathbb{R}}, \dot{\leq}) \underset{\alpha_\mathfrak{B}}{\overset{\gamma_\mathfrak{B}}{\leftrightarrows}} (\mathcal{P}(\mathfrak{B}), \supseteq)$. For $f : \mathcal{D} \to \overline{\mathbb{R}}$,

$$\alpha_\mathfrak{B}(f) = F^\sharp_{ub} \longleftrightarrow \bigwedge_{f^\sharp_i \in F^\sharp_{ub}} f \mathrel{\dot{\leq}} f^\sharp_i .$$



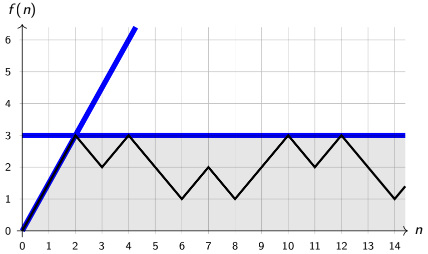
$\mathfrak{B}$-bounds — abstracting functions with simpler functions

A class of abstract domains for functions

For $\mathfrak{B} \subseteq (\mathcal{D} \to \overline{\mathbb{R}})$, let $D^\sharp := (\mathcal{P}(\mathfrak{B}), \supseteq)$ be the domain of $\mathfrak{B}$-upper bounds.

$(\mathcal{D} \to \overline{\mathbb{R}}, \dot{\leq}) \underset{\alpha_{\mathfrak{B}}}{\overset{\gamma_{\mathfrak{B}}}{\leftrightarrows}} (\mathcal{P}(\mathfrak{B}), \supseteq)$. For $f : \mathcal{D} \to \overline{\mathbb{R}}$,

$$\alpha_{\mathfrak{B}}(f) = F^\sharp_{ub} \longleftrightarrow \bigwedge_{f^\sharp_i \in F^\sharp_{ub}} f \mathrel{\dot{\leq}} f^\sharp_i .$$



Examples

- **Affine bounds.** $\mathfrak{B} = \{n \mapsto an + b \mid a, b \in \overline{\mathbb{R}}\}$ (one dim), $n \mapsto \vec{a} \cdot \vec{n} + b$ (multidim).
- **Polynomial bounds (bounded degree).** $n \mapsto \sum_{k \leq d} a_k n^k$ (monomial basis), $n \mapsto \sum a_k \binom{n}{k}$ (binomial basis), multidim versions, ...
- **Poly-exp.** $n \mapsto \sum a_{b,k} b^n n^k$, or $n \mapsto \sum a_{b,k} \binom{n}{k} \left\{ {n+1 \atop b+1} \right\}$ (with Stirling numbers of 2nd kind), ...
- **Arithmetico-geometric sequences**, **Regular expressions on numbers**, $\sum ab^n n^k \log(en + f)$...
- **Extra features:** initial exactness, piecewise behaviour (disjunctive versions), ...

Remark: functional version of a familiar concept (for relations) – constraint domains

For $E \in \mathcal{P}(\mathbb{R}^{\mathtt{Vars}})$,

$\alpha_B(E) = \mathtt{Box} \leftrightarrow \left(\forall x \in E,\ \bigwedge_i a_i \leq x_i \leq b_i\right)$,

$\alpha_P(E) = \mathtt{Polyhedron} \leftrightarrow \left(\forall x \in E,\ \bigwedge_i \sum_j \alpha_{i,j} x_j \leq \beta_i\right)$,

$\alpha_Z(E) = \mathtt{Zone} \leftrightarrow \left(\forall x \in E,\ \bigwedge_{i,j} x_i - x_j \leq c_{i,j}\right)$, ...

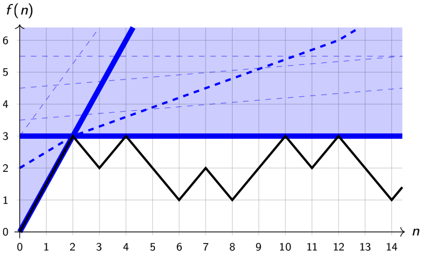
$\mathfrak{B}$-bounds — abstracting functions with simpler functions

A class of abstract domains for functions

For $\mathfrak{B} \subseteq (\mathcal{D} \to \overline{\mathbb{R}})$, let $D^\sharp := (\mathcal{P}(\mathfrak{B}), \supseteq)$ be the domain of $\mathfrak{B}$-upper bounds.

$(\mathcal{D} \to \overline{\mathbb{R}}, \dot{\leq}) \underset{\alpha_{\mathfrak{B}}}{\overset{\gamma_{\mathfrak{B}}}{\leftrightarrows}} (\mathcal{P}(\mathfrak{B}), \supseteq)$. For $f : \mathcal{D} \to \overline{\mathbb{R}}$,

$$\alpha_{\mathfrak{B}}(f) = F^\sharp_{ub} \longleftrightarrow \bigwedge_{f^\sharp_i \in F^\sharp_{ub}} f \mathrel{\dot{\leq}} f^\sharp_i.$$



Galois connection – $\mathfrak{B}$-ubs

$$(\mathcal{D} \to \overline{\mathbb{R}}, \dot{\leq}) \underset{\alpha_{\mathfrak{B}}}{\overset{\gamma_{\mathfrak{B}}}{\leftrightarrows}} (\mathcal{P}(\mathfrak{B}), \supseteq)$$

$$f \longmapsto \left\{ f^\sharp_{ub} \in \mathfrak{B} \,\middle|\, f \mathrel{\dot{\leq}} f^\sharp_{ub} \right\}$$

$$\left(\vec{n} \mapsto \min_{f^\sharp_{ub} \in F^\sharp_{ub}} f^\sharp_{ub}(\vec{n})\right) \longleftarrow F^\sharp_{ub}$$

In practice, we replace $(\mathcal{P}(\mathfrak{B}), \supseteq, \cap, \cup)$ by a more computable representation $(\mathcal{P}_{\uparrow,\mathrm{fin}}(\mathfrak{B}), \sqsubseteq^\sharp, \sqcup^\sharp, \sqcap^\sharp)$, with $A \sqsubseteq^\sharp B$ a sound approximation of $\uparrow A \supseteq \uparrow B$.

Remarks

- We can restrict to up-closed sets $\mathcal{P}_\uparrow(\mathfrak{B}) := \left\{ F \subseteq \mathfrak{B} \,\middle|\, \forall f \in F, \forall g \in \mathfrak{B},\ f \mathrel{\dot{\leq}} g \Rightarrow g \in F \right\}$.
- Up-closure $(\uparrow) = \alpha_{\mathfrak{B}} \circ \gamma_{\mathfrak{B}} : \mathcal{P}(\mathfrak{B}) \to \mathcal{P}_\uparrow(\mathfrak{B})$.
- Search for a finite number of **generators**
  $$\uparrow\{f^\sharp_1, \ldots, f^\sharp_k\} = \alpha_{\mathfrak{B}}(f),$$
  (or at least an overapproximation of $\alpha_{\mathfrak{B}}(f)$).
- → Add `normalise` and `widening` operators to keep the representation bounded.

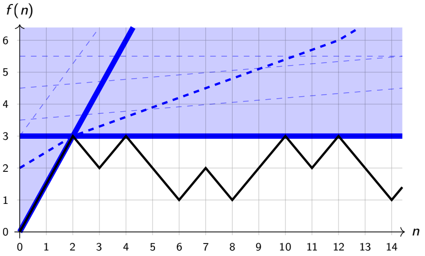
$\mathfrak{B}$-bounds — abstracting functions with simpler functions

A class of abstract domains for functions

For $\mathfrak{B} \subseteq (\mathcal{D} \to \overline{\mathbb{R}})$, let $D^\sharp := (\mathcal{P}(\mathfrak{B}), \supseteq)$ be the domain of $\mathfrak{B}$-upper bounds.

$(\mathcal{D} \to \overline{\mathbb{R}}, \dot{\leq}) \underset{\alpha_{\mathfrak{B}}}{\overset{\gamma_{\mathfrak{B}}}{\leftrightarrows}} (\mathcal{P}(\mathfrak{B}), \supseteq)$. For $f : \mathcal{D} \to \overline{\mathbb{R}}$,

$$\alpha_{\mathfrak{B}}(f) = F^\sharp_{ub} \longleftrightarrow \bigwedge_{f^\sharp_i \in F^\sharp_{ub}} f \mathrel{\dot{\leq}} f^\sharp_i.$$



With intervals – Galois connection with $\mathfrak{B}$-bounds: "flowpipes"

$$(\mathcal{D} \to \mathcal{I}(\mathbb{R}), \dot{\sqsubseteq}_{\mathcal{I}}) \underset{\alpha_{\mathfrak{B}}}{\overset{\gamma_{\mathfrak{B}}}{\leftrightarrows}} (\mathcal{P}(\mathfrak{B}), \supseteq) \times (\mathcal{P}(\mathfrak{B}), \supseteq)$$

$$\left(\vec{n} \mapsto \left[\max_{f^\sharp_{lb} \in F^\sharp_{lb}} f^\sharp_{lb}(\vec{n}),\ \min_{f^\sharp_{ub} \in F^\sharp_{ub}} f^\sharp_{ub}(\vec{n})\right]\right) \longleftarrow (F^\sharp_{lb},\ F^\sharp_{ub})$$

$$(f_{lb},\ f_{ub}) \longmapsto \left(\begin{array}{l} \left\{f^\sharp_{lb} \in \mathfrak{B} \,\middle|\, \forall \vec{n} \in \mathcal{D},\ f^\sharp_{lb}(\vec{n}) \leq f_{lb}(\vec{n})\right\}, \\ \left\{f^\sharp_{ub} \in \mathfrak{B} \,\middle|\, \forall \vec{n} \in \mathcal{D},\ f^\sharp_{ub}(\vec{n}) \geq f_{ub}(\vec{n})\right\} \end{array}\right)$$

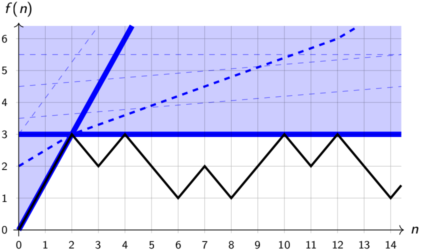
$\mathfrak{B}$-bounds — abstracting functions with simpler functions

A class of abstract domains for functions

For $\mathfrak{B} \subseteq (\mathcal{D} \to \overline{\mathbb{R}})$, let $D^\sharp := (\mathcal{P}(\mathfrak{B}), \supseteq)$ be the domain of $\mathfrak{B}$-upper bounds.

$(\mathcal{D} \to \overline{\mathbb{R}}, \dot{\leq}) \underset{\alpha_{\mathfrak{B}}}{\overset{\gamma_{\mathfrak{B}}}{\leftrightarrows}} (\mathcal{P}(\mathfrak{B}), \supseteq)$. For $f : \mathcal{D} \to \overline{\mathbb{R}}$,

$$\alpha_{\mathfrak{B}}(f) = F^\sharp_{ub} \longleftrightarrow \bigwedge_{f_i^\sharp \in F^\sharp_{ub}} f \,\dot{\leq}\, f_i^\sharp.$$



Proposition (convexity of constraint set)

When $\mathfrak{B}$ is convex,

all $\alpha(f)$ are convex *(in function space).*

Example

Most of the templates $\mathfrak{B}$ discussed before are convex, e.g.

$$\lambda \cdot \left(n \mapsto \sum a_k n^k\right) + (1-\lambda) \cdot \left(n \mapsto \sum b_k n^k\right)$$
$$= \left(n \mapsto \sum (\lambda a_k + (1-\lambda) b_k) n^k\right).$$

➔ This reduces the problem of finding a minimal set of extremal bounds (e.g. for **transfer function synthesis**) to **finding generators of a convex set.**

To make the problem tractable in practice, we often approximate the problem $\Phi(\gamma(f^\sharp)) \,\dot{\leq}\, g^\sharp$, and **work in parameter space**, using an order that is an incomplete abstraction of $\dot{\leq}$, e.g.

$$\left(\left(n \mapsto \sum a_k n^k\right) \sqsubseteq^\sharp \left(n \mapsto \sum b_k n^k\right)\right) \overset{\Delta}{\iff} \left(\forall k,\ a_k \leq b_k\right), \qquad \text{(despite } n^2 - 1 \,\dot{\geq}_{\mathbb{N}}\, 0\text{)}.$$

$\mathfrak{B}$-bounds: discussion

We have discussed Galois connections, and mentioned that the domain can support non-linear bounds.

Let's make this concrete, with fully-fledged abstract domains:

- add a **language**, (we will present an operator language)
- design **transfer functions**,
- showcase **abstract iteration**.

➔ Let's build some $\mathfrak{B}$-bounds domains!

Other applications of $\mathfrak{B}$-bound domains, beyond abstract iteration:

- Simplify analysis output for users: $f \leq g_{\texttt{precise}} \rightsquigarrow \bigwedge_i f \leq g_{i,\texttt{readable}}$,
- Help decide inequality of functions: $f \dot{\leq}^{(?)} g$,
- ...

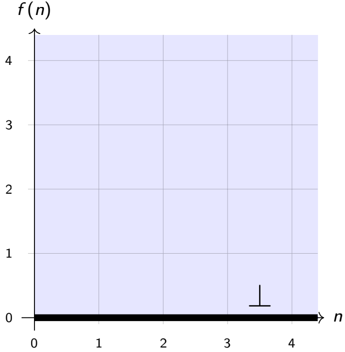
Abstract Domains of Sequences: A simple way towards non-linearity

Abstract iteration – what we want to get

Equation $\Phi(f) = \mathrm{ite}(n > 0, f(f(n-1)) + 1, 0)$, $D = \mathbb{N} \to \mathbb{N}_\infty$, $\mathfrak{B} = \mathtt{Affines}$.

➔ We deal with extremal bounds separately.

$$\begin{tikzcd} D^\sharp = \mathcal{P}_\uparrow(\mathfrak{B}) \arrow[r, bend left, dotted, "\Phi(\circ\gamma) \text{ on extremal bounds}"] & \mathcal{P}(D) \arrow[l, bend left, dotted, "\text{union of } \alpha\text{'s}"] \end{tikzcd}$$



$\perp^\sharp = \uparrow\{0n + 0\}$

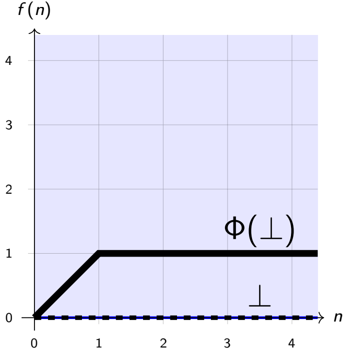
Abstract iteration – what we want to get

Equation $\Phi(f) = \mathrm{ite}(n > 0, f(f(n-1)) + 1, 0)$, $D = \mathbb{N} \to \mathbb{N}_\infty$, $\mathfrak{B} = \texttt{Affines}$.

➔ We deal with extremal bounds separately.

$$
\begin{tikzcd}
D^\sharp = \mathcal{P}_\uparrow(\mathfrak{B}) \arrow[rr, bend left, dotted, "\Phi(\circ\gamma) \text{ on extremal bounds}"] & & \mathcal{P}(D) \arrow[ll, bend left, dotted, "\text{union of } \alpha\text{'s}"]
\end{tikzcd}
$$

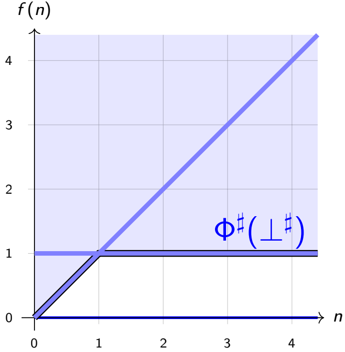
$$\perp^\sharp = \uparrow\{0n + 0\} \longrightarrow \{01^*\}$$



Abstract iteration – what we want to get

Equation $\Phi(f) = \mathrm{ite}(n > 0, f(f(n-1)) + 1, 0)$, $D = \mathbb{N} \to \mathbb{N}_\infty$, $\mathfrak{B} = \texttt{Affines}$.

➔ We deal with extremal bounds separately.



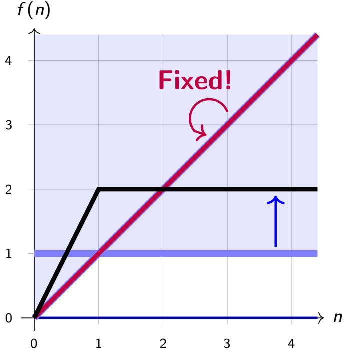
$$
\begin{tikzcd}
D^\sharp = \mathcal{P}_\uparrow(\mathfrak{B}) \arrow[rr, dotted, bend left, "\Phi(\circ\gamma)\text{ on extremal bounds}"] & & \mathcal{P}(D) \arrow[ll, dotted, bend left, "\text{union of }\alpha\text{'s}"]
\end{tikzcd}
$$

$$
\begin{tikzcd}
\bot^\sharp = \uparrow\{0n+0\} \arrow[rr] \arrow[d, dotted, "\alpha\circ\Phi\circ\gamma \;\sqsubseteq"] & & \{01^*\} \arrow[dll] \\
\uparrow\{{\color{purple}1n+0},\ 0n+1\} & &
\end{tikzcd}
$$

Abstract iteration – what we want to get

Equation $\Phi(f) = \mathrm{ite}(n > 0, f(f(n-1)) + 1, 0)$, $D = \mathbb{N} \to \mathbb{N}_\infty$, $\mathfrak{B} = \texttt{Affines}$.

➔ We deal with extremal bounds separately.



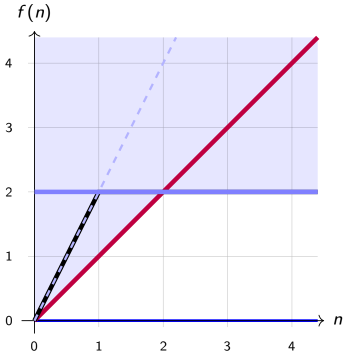
$$\begin{tikzcd} D^\sharp = \mathcal{P}_\uparrow(\mathfrak{B}) \arrow[rr, bend left, dotted, "\Phi(\circ\gamma) \text{ on extremal bounds}"] & & \mathcal{P}(D) \arrow[ll, bend left, dotted, "\text{union of } \alpha\text{'s}"] \end{tikzcd}$$

$$\begin{tikzcd} \perp^\sharp = \uparrow\{0n+0\} \arrow[r] \arrow[d, dotted, "\alpha\circ\Phi\circ\gamma" description, "\sqsupseteq"'] & \{01^*\} \arrow[dl] \\ \uparrow\{1n+0,\ 0n+1\} \arrow[r] & \{1n+0,\ 02^*\} \end{tikzcd}$$

Abstract iteration – what we want to get

Equation $\Phi(f) = \mathrm{ite}(n > 0, f(f(n-1)) + 1, 0)$, $D = \mathbb{N} \to \mathbb{N}_\infty$, $\mathfrak{B} = \texttt{Affines}$.

➔ We deal with extremal bounds separately.



$\Phi(\circ\gamma)$ on extremal bounds

$D^\sharp = \mathcal{P}_\uparrow(\mathfrak{B})$ ⇄ $\mathcal{P}(D)$

union of α's

$\bot^\sharp = \uparrow\{0n+0\} \longrightarrow \{01^*\}$

$\big\downarrow \alpha\circ\Phi\circ\gamma \;\; \sqsupseteq$

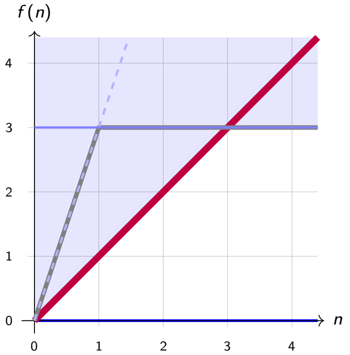
$\uparrow\{1n+0,\ 0n+1\} \longrightarrow \{1n+0,\ 02^*\}$

$$\uparrow\left\{ \begin{array}{l} 1n+0, \\ 2n+0,\ 0n+2 \end{array} \right\}$$

Abstract iteration – what we want to get

Equation $\Phi(f) = \mathrm{ite}(n > 0, f(f(n-1)) + 1, 0)$, $D = \mathbb{N} \to \mathbb{N}_\infty$, $\mathfrak{B} = \texttt{Affines}$.

➔ We deal with extremal bounds separately.



$$D^\sharp = \mathcal{P}_\uparrow(\mathfrak{B}) \underset{\text{union of } \alpha\text{'s}}{\overset{\Phi(\circ\gamma) \text{ on extremal bounds}}{\rightleftarrows}} \mathcal{P}(D)$$

$$\perp^\sharp = \uparrow\{0n+0\} \longrightarrow \{01^*\}$$

$$\alpha\circ\Phi\circ\gamma \;\sqsubseteq$$

$$\uparrow\{1n+0,\ 0n+1\} \longrightarrow \{1n+0,\ 02^*\}$$

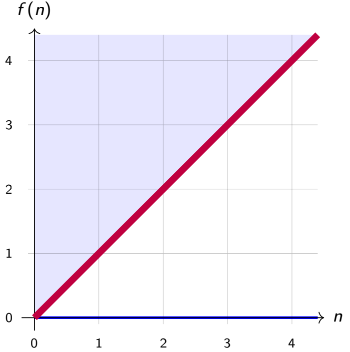
$$\uparrow\left\{\begin{array}{l} 1n+0, \\ \cancel{2n+0},\ 0n+2 \end{array}\right\} \longrightarrow \{1n+0,\ 03^*\}$$

$$\uparrow\left\{\begin{array}{l} 1n+0, \\ \cancel{3n+0},\ 0n+3 \end{array}\right\}$$

Abstract iteration – what we want to get

Equation $\Phi(f) = \mathrm{ite}(n > 0, f(f(n-1)) + 1, 0)$, $D = \mathbb{N} \to \mathbb{N}_\infty$, $\mathfrak{B} = \texttt{Affines}$.

➔ We deal with extremal bounds separately.



$\Phi(\circ\gamma)$ on extremal bounds

union of α's

$$D^\sharp = \mathcal{P}_\uparrow(\mathfrak{B}) \qquad \mathcal{P}(D)$$

$$\bot^\sharp = \uparrow\{0n+0\} \longrightarrow \{01^*\}$$

$\alpha\circ\Phi\circ\gamma$ $\sqsupseteq$

$$\uparrow\{1n+0,\ 0n+1\} \longrightarrow \{1n+0,\ 02^*\}$$

$$\uparrow\left\{\begin{array}{l} 1n+0, \\ \cancel{2n+0},\ 0n+2 \end{array}\right\} \longrightarrow \{1n+0,\ 03^*\}$$

$$\uparrow\left\{\begin{array}{l} 1n+0, \\ \cancel{3n+0},\ 0n+3 \end{array}\right\}$$

$$\uparrow\{1n+0,\ \top\} = \uparrow\{1n+0\}$$

Sequences

We start with an easy case: one dim., simple recursive structure (push/pop).

Syntax

`<expr>` ::=
  | `Cst <num>` | `'n'` | `'f'`
  | `<expr>` $\Diamond$ `<expr>`
  | `Pop <expr>` | `Push <num> <expr>`
  | ...

`<num>` ::= $c \in \overline{\mathbb{R}}$ $\Diamond \in \{+, -, \times\}$

Seq: A simple operator language

Example

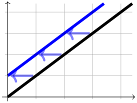
$$f(0) = 4,\ f(n) = 1/2 \cdot f(n-1) + 3$$
$$\longleftrightarrow \Phi : f \mapsto n \mapsto \mathrm{ite}(n = 0,\ 4,\ 1/2 \cdot f(n-1) + 3)$$
$$\longleftrightarrow \texttt{Push}\ 4\ \left(\texttt{Cst}(1/2) \times (\texttt{'f'} + \texttt{Cst}(3))\right)$$

Semantics

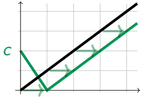
$$[\![\cdot]\!] : \texttt{Seq} \to \mathrm{End}(\mathbb{N} \to \overline{\mathbb{R}})$$

$$[\![\texttt{Cst}\ c]\!](f) = n \mapsto c$$
$$[\![\texttt{'n'}]\!](f) = n \mapsto n$$
$$[\![\texttt{'f'}]\!](f) = f$$
$$[\![e_1 \Diamond e_2]\!](f) = [\![e_1]\!](f)\ \dot{\Diamond}\ [\![e_2]\!](f)$$

$$\boxed{\begin{aligned} [\![\texttt{Pop}]\!](f) &= n \mapsto f(n+1) \\ [\![\texttt{Push}\ c]\!](f) &= n \mapsto \mathrm{ite}(n = 0, c, f(n-1)) \end{aligned}}$$



Pop



Push c

Sequences

We start with an easy case: one dim., simple recursive structure (push/pop).

Syntax

<expr> ::=
| Cst <num> | 'n' | 'f'
| <expr> $\Diamond$ <expr>
| Pop <expr> | Push <num> <expr>
| Shift <...> | Scale <...> | ...

<num> ::= $c \in \overline{\mathbb{R}}$ $\quad \Diamond \in \{+, -, \times\} \cup \{\circ\}$

Seq: A simple operator language

Example

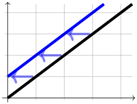
$$f(0) = 4,\ f(n) = 1/2 \cdot f(n-1) + 3$$
$$\longleftrightarrow \Phi : f \mapsto n \mapsto \mathrm{ite}(n = 0,\ 4,\ 1/2 \cdot f(n-1) + 3)$$
$$\longleftrightarrow \texttt{Push 4 } (\texttt{Cst}(1/2) \times (\texttt{'f'} + \texttt{Cst}(3)))$$

Semantics

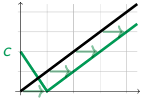
$$[\![\cdot]\!] : \texttt{Seq} \to \mathrm{End}(\mathbb{N} \to \overline{\mathbb{R}})$$

$$[\![\texttt{Cst } c]\!](f) = n \mapsto c$$
$$[\![\texttt{'n'}]\!](f) = n \mapsto n$$
$$[\![\texttt{'f'}]\!](f) = f$$
$$[\![e_1 \Diamond e_2]\!](f) = [\![e_1]\!](f) \dot{\Diamond} [\![e_2]\!](f)$$

$$\boxed{\begin{aligned}[\![\mathrm{Pop}]\!](f) &= n \mapsto f(n+1) \\ [\![\mathrm{Push}\ c]\!](f) &= n \mapsto \mathrm{ite}(n = 0, c, f(n-1))\end{aligned}}$$



Pop



Push c

Warm-up: affine bounds

$$\mathfrak{B} = \left\{ n \mapsto an + b \;\middle|\; a \in \mathbb{R}_+ \cup \{+\infty\},\ b \in \mathbb{R}_+ \right\} \cup \{\top_{\mathfrak{B}}\}$$

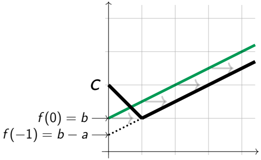
For simpler presentation, here only ub, and positive coefficients (need lb for subtraction and multiplication by negatives).

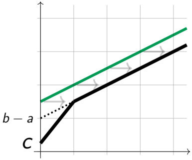
Here, **coordinate order** is **complete** $\left(a_1 n + b_1 \dot{\leq} a_2 n + b_2 \iff (a_1, b_1) \sqsubseteq_{\mathfrak{B}} (a_2, b_2)\right)$

$$\boxed{\begin{array}{l} (a_1, b_1) \sqsubseteq_{\mathfrak{B}} (a_2, b_2) \iff a_1 \leq a_2 \wedge b_1 \leq b_2, \qquad \bot_{\mathfrak{B}} = (0,0) \qquad \top_{\mathfrak{B}} \approx (\_, +\infty) \\ \qquad \bigsqcup\{(a_i, b_i)\} = \left(\max_i a_i,\ \max_i b_i\right), \qquad \bigsqcap\{(a_i, b_i)\} = \left(\min_i a_i,\ \min_i b_i\right), \end{array}}$$

For simplicity, define transfer functions $\mathfrak{B} \to \mathcal{P}(\mathfrak{B})$, then extend to $\mathcal{P}(\mathfrak{B}) \to \mathcal{P}(\mathfrak{B})$.

- Constants, addition, multiplication easily represented, e.g.
  $(a_1 n + b_1) + (a_2 n + b_2) = (a_1 + a_2) n + (b_1 + b_2)$
  $(a_1 n + b_1) \times (a_2 n + b_2) = \left(\underbrace{a_1 a_2 \infty}_{a_1 a_2 n^2} + a_1 b_2 + a_2 b_1\right) n + (b_1 b_2)$
- $[\![\mathrm{Pop}]\!]^\sharp((a, b)) = \{(a, b + a)\}$
- $[\![\mathrm{Push}\ c]\!]^\sharp((a, b)) = \begin{cases} \{(a, c)\} & \text{if } c \geq b - a \\ \{(a, b - a), (b - c, c)\} & \text{if } c < b - a \end{cases}$





Convex problem for $[\![\mathrm{Push}\ c]\!]^\sharp$

$$(n \mapsto un + v) \dot{\geq} [\![\mathrm{Push}\ c]\!](n \mapsto an + b)$$
$$\iff$$
$$\begin{cases} & v \geq c \\ \wedge & [\![\mathrm{Pop}]\!](n \mapsto un + v) \dot{\geq} (n \mapsto an + b) \end{cases}$$
$$\iff$$
$$\begin{cases} & v \geq c \\ \wedge & \forall n \in \mathbb{N},\ u \cdot (n+1) + v \geq an + b \end{cases}$$
$$\iff$$
$$\boxed{\begin{cases} & v \geq c \\ \wedge & u \geq a \\ \wedge & u + v \geq b \end{cases}}$$

Warm-up: affine bounds

$$\mathfrak{B} = \left\{ n \mapsto an + b \;\middle|\; a \in \mathbb{R}_+ \cup \{+\infty\},\ b \in \mathbb{R}_+ \right\} \cup \{\top_{\mathfrak{B}}\}$$

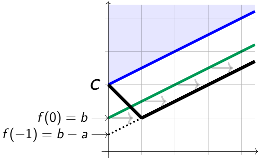
For simpler presentation, here only ub, and positive coefficients (need lb for subtraction and multiplication by negatives).

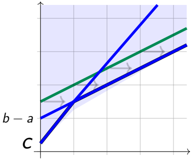
Here, **coordinate order** is **complete** $\left(a_1 n + b_1 \,\dot{\leq}\, a_2 n + b_2 \iff (a_1, b_1) \sqsubseteq_{\mathfrak{B}} (a_2, b_2)\right)$

$$\boxed{\begin{array}{c} (a_1, b_1) \sqsubseteq_{\mathfrak{B}} (a_2, b_2) \iff a_1 \leq a_2 \wedge b_1 \leq b_2, \qquad \bot_{\mathfrak{B}} = (0,0) \qquad \top_{\mathfrak{B}} \approx (\_, +\infty) \\ \bigsqcup\{(a_i, b_i)\} = \left(\max_i a_i,\ \max_i b_i\right), \qquad \bigsqcap\{(a_i, b_i)\} = \left(\min_i a_i,\ \min_i b_i\right), \end{array}}$$

For simplicity, define transfer functions $\mathfrak{B} \to \mathcal{P}(\mathfrak{B})$, then extend to $\mathcal{P}(\mathfrak{B}) \to \mathcal{P}(\mathfrak{B})$.

- Constants, addition, multiplication easily represented, e.g.
  $(a_1 n + b_1) + (a_2 n + b_2) = (a_1 + a_2)n + (b_1 + b_2)$
  $(a_1 n + b_1) \times (a_2 n + b_2) = \big(\underbrace{a_1 a_2 \infty}_{a_1 a_2 n^2} + a_1 b_2 + a_2 b_1\big)n + (b_1 b_2)$
- $[\![\mathrm{Pop}]\!]^\sharp((a,b)) = \{(a, b+a)\}$
- $[\![\mathrm{Push}\ c]\!]^\sharp((a,b)) = \begin{cases} \{(a,c)\} & \text{if } c \geq b - a \\ \{(a, b-a), (b-c, c)\} & \text{if } c < b - a \end{cases}$





Convex problem for $[\![\mathrm{Push}\ c]\!]^\sharp$

$$(n \mapsto un + v) \,\dot{\geq}\, [\![\mathrm{Push}\ c]\!](n \mapsto an + b)$$
$$\iff$$
$$\begin{cases} & v \geq c \\ \wedge & [\![\mathrm{Pop}]\!](n \mapsto un + v) \,\dot{\geq}\, (n \mapsto an + b) \end{cases}$$
$$\iff$$
$$\begin{cases} & v \geq c \\ \wedge & \forall n \in \mathbb{N},\ u \cdot (n+1) + v \geq an + b \end{cases}$$
$$\iff$$
$$\boxed{\begin{cases} & v \geq c \\ \wedge & u \geq a \\ \wedge & u + v \geq b \end{cases}}$$

Polynomial bounds

$$\mathfrak{B} = \left\{ n \mapsto \sum_{k \leq d} a_k n^k \;\middle|\; a_k \in \mathbb{R}_+ \right\} \cup \left\{ a_0 (+\infty)^* \;\middle|\; a_0 \in \mathbb{R}_+ \right\} \cup \{ \top_{\mathfrak{B}} \}$$

- Similar idea here: **polynomials are closed by simple arithmetic operations** ($+$, $\times$, $\circ$, ...), and **representations** of these operations **are easy to compute**.
- The coordinate-wise order becomes incomplete (e.g. $1 \cdot n^2 + 2 \cdot n \dot{\leq} 2 \cdot n^2 + 1 \cdot n$), but this is okay and we can still decide $\dot{\leq}$ if necessary.

Polynomial bounds

$$\mathfrak{B} = \left\{ n \mapsto \sum_{k \leq d} a_k n^k \;\middle|\; a_k \in \mathbb{R}_+ \right\} \cup \left\{ a_0(+\infty)^* \mid a_0 \in \mathbb{R}_+ \right\} \cup \{\top_{\mathfrak{B}}\}$$

- Similar idea here: **polynomials are closed by simple arithmetic operations** ($+$, $\times$, $\circ$, ...), and **representations** of these operations **are easy to compute**.
- The coordinate-wise order becomes incomplete (e.g. $1 \cdot n^2 + 2 \cdot n \dot{\lesssim} 2 \cdot n^2 + 1 \cdot n$), but this is okay and we can still decide $\dot{\lesssim}$ if necessary.
- Pop and Push are a bit more complex, but this is manageable.

Pop $\quad \displaystyle\sum_{k \leq d} a_k (n+1)^k = \sum_{k \leq d} \Bigg(\underbrace{\sum_{j \leq d} a_k \binom{j}{k}}_{p_k} \Bigg) n^k$

Push x $\quad$??

Convex problem for $[\![\texttt{Push } x]\!]^\sharp$

$$\left(n \mapsto \sum p_k n^k\right) \dot{\geq} [\![\texttt{Push } x]\!] \left(n \mapsto \sum a_k n^k\right)$$

$$\Longleftarrow$$

$$\begin{cases} & p_0 \geq x \\ \wedge & \forall k \in [\![0, d]\!],\ \sum_{j \leq d} p_k \binom{j}{k} \geq a_k \end{cases}$$

Polynomial bounds

$$\mathfrak{B} = \left\{ n \mapsto \sum_{k \leq d} a_k n^k \;\middle|\; a_k \in \mathbb{R}_+ \right\} \cup \left\{ a_0 (+\infty)^* \mid a_0 \in \mathbb{R}_+ \right\} \cup \{\top_{\mathfrak{B}}\}$$

- Similar idea here: **polynomials are closed by simple arithmetic operations** $(+, \times, \circ, \ldots)$, and **representations** of these operations **are easy to compute**.
- The coordinate-wise order becomes incomplete (e.g. $1 \cdot n^2 + 2 \cdot n \dot{\lesssim} 2 \cdot n^2 + 1 \cdot n$), but this is okay and we can still decide $\dot{\lesssim}$ if necessary.
- Pop and Push are a bit more complex, but this is manageable.

Pop $\quad \displaystyle\sum_{k \leq d} a_k (n+1)^k = \sum_{k \leq d} \underbrace{\left(\sum_{j \leq d} a_k \binom{j}{k} \right)}_{p_k} n^k$

Push $x \quad$??

Convex problem for $[\![\texttt{Push } x]\!]^\sharp$

$$\left(n \mapsto \sum p_k n^k\right) \dot{\geq} [\![\texttt{Push } x]\!] \left(n \mapsto \sum a_k n^k\right)$$

$$\Longleftarrow$$

$$\begin{cases} & p_0 \geq x \\ \wedge & \forall k \in [\![0, d]\!],\ \sum_{j \leq d} p_k \binom{j}{k} \geq a_k \end{cases}$$

➔ There is a nice trick, coming from works on type-based amortised cost analysis [HoffmannPhd11]: **Use the binomial basis instead of the monomial basis**, which has much simpler behaviour under shifts.

Polynomial bounds – Binomial basis

$$\mathfrak{B} = \left\{ n \mapsto \sum_{k \leq d} a_k \binom{n}{k} \,\middle|\, a_k \in \mathbb{R}_+ \right\} \cup \left\{ a_0 (+\infty)^* \,\middle|\, a_0 \in \mathbb{R}_+ \right\} \cup \{ \top_{\mathfrak{B}} \}$$

- Instead of the basis $1, n, n^2, n^3, \ldots$ of polynomials, use the basis $\binom{n}{0}, \binom{n}{1}, \binom{n}{2}, \binom{n}{3}, \ldots$,
  Examples: $\binom{n}{0} = 1$, $\binom{n}{1} = n$, $\binom{n}{2} = \frac{1}{2}n^2 - \frac{1}{2}n$, $\binom{n}{3} = \frac{1}{6}n^3 - \frac{1}{2}n^2 + \frac{1}{3}n$,
- We can convert between the two representation easily if needed
  (e.g. multiplication, composition, rescaling, etc., can be easier in monomial basis).
- Key identity: $$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}.$$
- The problems for `Pop` and `Push` become much easier.

`Pop` $$\sum_{k \leq d} a_k \binom{n+1}{k} = \sum_{k \leq d} \Big(\underbrace{a_k + a_{k+1}}_{p_k} \Big) \binom{n}{k}$$

`Push` x ...

Convex problem for $[\![\texttt{Push}\ x]\!]^\sharp$

$$\left(n \mapsto \sum p_k n^k\right) \dot{\geq} [\![\texttt{Push}\ x]\!] \left(n \mapsto \sum a_k n^k\right)$$

$$\Longleftarrow$$

$$\begin{cases} & p_0 \geq x \\ \wedge & p_d \geq a_d \\ \wedge \quad \forall k < d, & p_k + p_{k+1} \geq a_k \end{cases}$$

Polynomial bounds – Binomial basis

$$\mathfrak{B} = \left\{ n \mapsto \sum_{k \leq d} a_k \binom{n}{k} \;\middle|\; a_k \in \mathbb{R}_+ \right\} \cup \left\{ a_0 (+\infty)^* \mid a_0 \in \mathbb{R}_+ \right\} \cup \left\{ \top_{\mathfrak{B}} \right\}$$

- Instead of the basis $1, n, n^2, n^3, \ldots$ of polynomials, use the basis $\binom{n}{0}, \binom{n}{1}, \binom{n}{2}, \binom{n}{3}, \ldots,$
  Examples: $\binom{n}{0} = 1$, $\binom{n}{1} = n$, $\binom{n}{2} = \frac{1}{2}n^2 - \frac{1}{2}n$, $\binom{n}{3} = \frac{1}{6}n^3 - \frac{1}{2}n^2 + \frac{1}{3}n$,
- We can convert between the two representation easily if needed
  (e.g. multiplication, composition, rescaling, etc., can be easier in monomial basis).
- Key identity: $$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}.$$
- The problems for `Pop` and `Push` become much easier.

`Pop` $$\sum_{k \leq d} a_k \binom{n+1}{k} = \sum_{k \leq d} \Big(\underbrace{a_k + a_{k+1}}_{p_k} \Big) \binom{n}{k}$$

`Push` x e.g., for $d = 2$, $\{(p_2, p_1, p_0)\} =$

$$\begin{cases} \{(a_2, a_1 - a_2, x),\ (x + a_1 - a_0, a_0 - x, x)\} & \text{if } x > a_0 - a_1 + a_2 \\ \{(a_2, a_1 - a_2, a_0 - a_1 + a_2)\} & \text{if } x = a_0 - a_1 + a_2 \\ \{(a_2, a_1 - a_2, a_0 - a_1 + a_2),\ (a_2, a_0 - x, x)\} & \text{if } x < a_0 - a_1 + a_2 \end{cases}$$

Convex problem for $[\![\texttt{Push } x]\!]^\sharp$

$$\left(n \mapsto \sum p_k n^k\right) \dot{\geq} [\![\texttt{Push } x]\!] \left(n \mapsto \sum a_k n^k\right)$$

$$\Longleftarrow$$

$$\begin{cases} & p_0 \geq x \\ \wedge & p_d \geq a_d \\ \wedge & \forall k < d,\ p_k + p_{k+1} \geq a_k \end{cases}$$

Polynomial bounds – Binomial basis

$$\mathfrak{B} = \left\{ n \mapsto \sum_{k \le d} a_k \binom{n}{k} \;\middle|\; a_k \in \mathbb{R}_+ \right\} \cup \left\{ a_0 (+\infty)^* \;\middle|\; a_0 \in \mathbb{R}_+ \right\} \cup \{\top_{\mathfrak{B}}\}$$

- Instead of the basis $1, n, n^2, n^3, \dots$ of polynomials, use the basis $\binom{n}{0}, \binom{n}{1}, \binom{n}{2}, \binom{n}{3}, \dots,$
  Examples: $\binom{n}{0} = 1$, $\binom{n}{1} = n$, $\binom{n}{2} = \frac{1}{2}n^2 - \frac{1}{2}n$, $\binom{n}{3} = \frac{1}{6}n^3 - \frac{1}{2}n^2 + \frac{1}{3}n$,
- We can convert between the two representation easily if needed
  (e.g. multiplication, composition, rescaling, etc., can be easier in monomial basis).
- Key identity: $$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}.$$
- The problems for `Pop` and `Push` become much easier.

`Pop` $$\sum_{k \le d} a_k \binom{n+1}{k} = \sum_{k \le d} \Big(\underbrace{a_k + a_{k+1}}_{p_k}\Big) \binom{n}{k}$$

`Push` x e.g., for $d = 2$, $\{(p_2, p_1, p_0)\} =$

$$\begin{cases} \{(a_2, a_1 - a_2, x),\ (x + a_1 - a_0, a_0 - x, x)\} & \text{if } x > a_0 - a_1 + a_2 \\ \{(a_2, a_1 - a_2, a_0 - a_1 + a_2)\} & \text{if } x = a_0 - a_1 + a_2 \\ \{(a_2, a_1 - a_2, a_0 - a_1 + a_2),\ (a_2, a_0 - x, x)\} & \text{if } x < a_0 - a_1 + a_2 \end{cases}$$

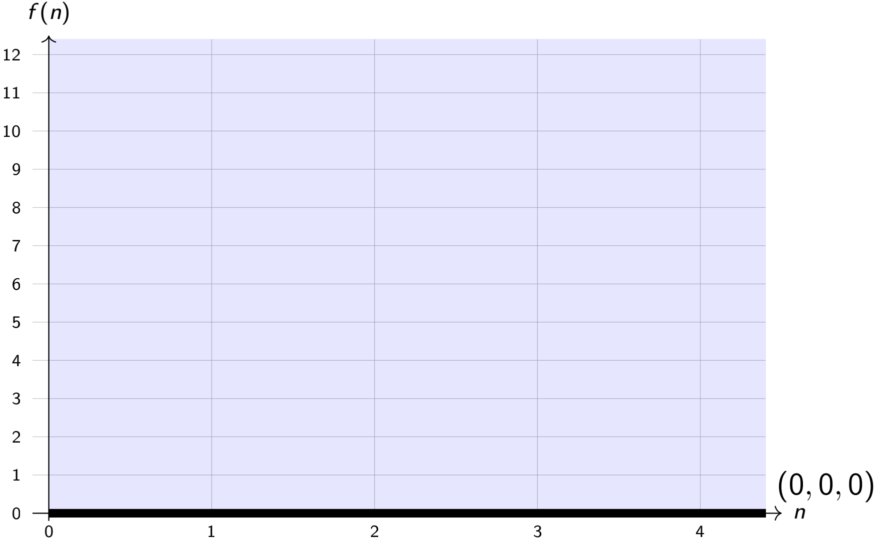
Convex problem for $[\![\texttt{Push}\ x]\!]^\sharp$

$$\left(n \mapsto \sum p_k n^k\right) \dot{\ge} [\![\texttt{Push}\ x]\!] \left(n \mapsto \sum a_k n^k\right)$$
$$\Longleftarrow$$
$$\begin{cases} & p_0 \ge x \\ \wedge & p_d \ge a_d \\ \wedge & \forall k < d,\ p_k + p_{k+1} \ge a_k \end{cases}$$

➔ **Extends well to higher dimensions, with multivariate polynomials.**

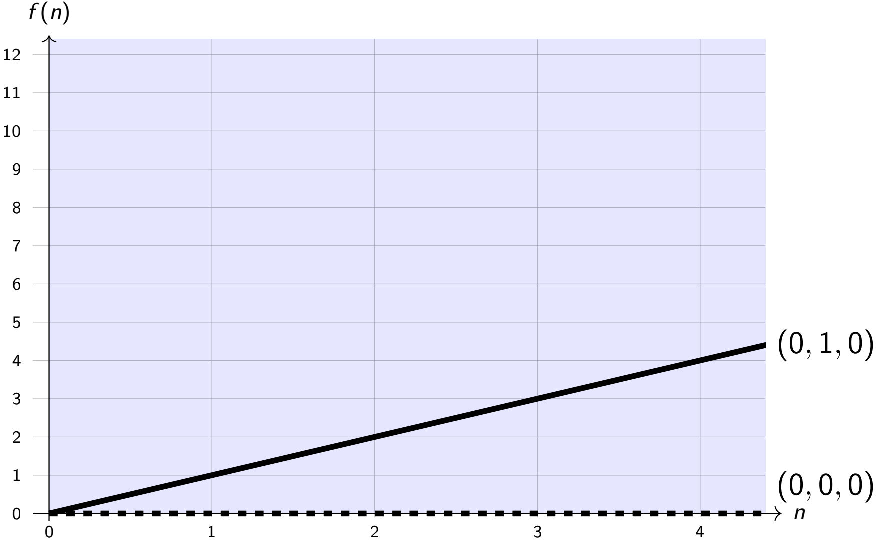
Polynomial bounds – Example

Very simple example: $\Phi(f)(n) = \mathrm{ite}(n = 0,\ 0,\ f(n-1) + n)$. $(f_{sol} = (1, 1, 0))$



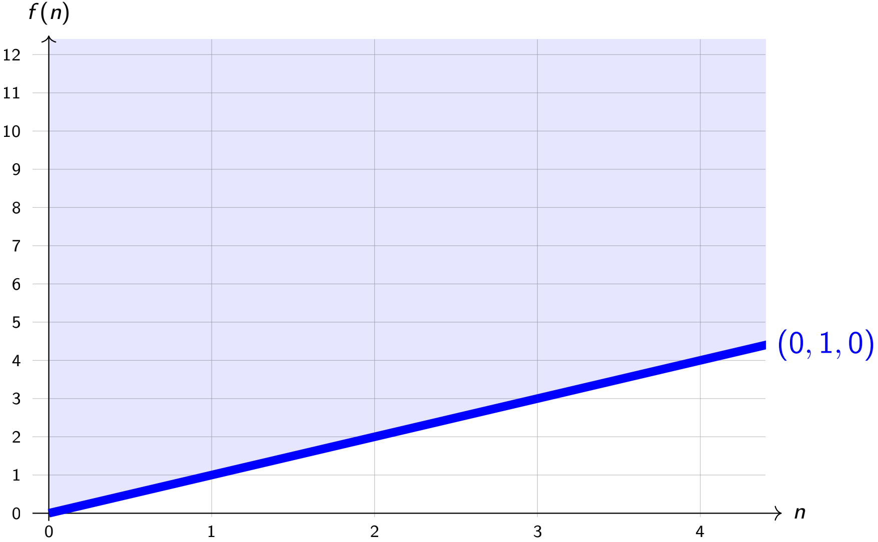
Polynomial bounds – Example

Very simple example: $\Phi(f)(n) = \mathrm{ite}(n = 0,\ 0,\ f(n-1) + n)$. $(f_{sol} = (1, 1, 0))$



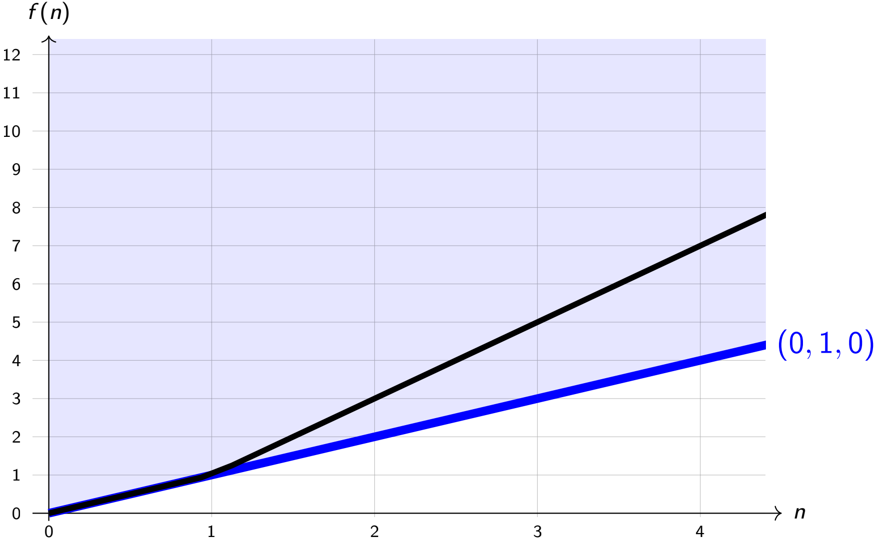
Polynomial bounds – Example

Very simple example: $\Phi(f)(n) = \mathrm{ite}(n = 0,\ 0,\ f(n-1) + n)$. $(f_{sol} = (1, 1, 0))$



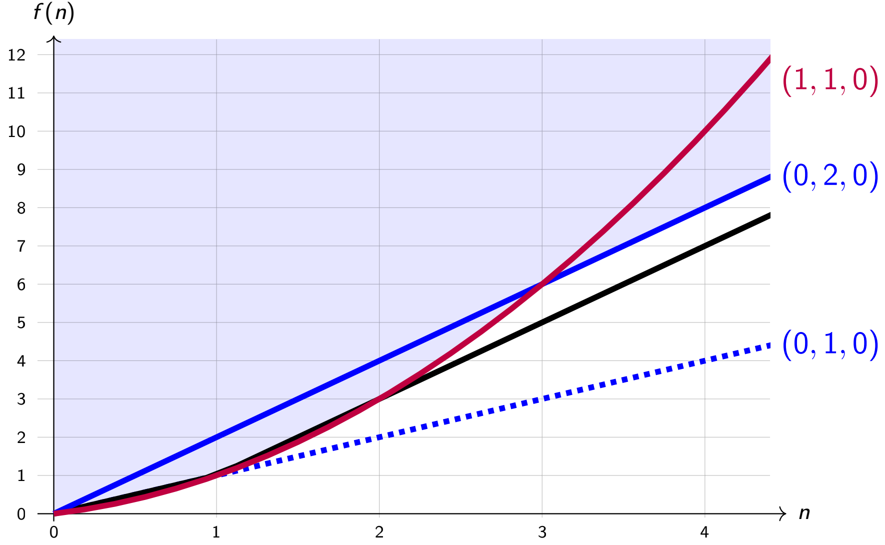
Polynomial bounds – Example

Very simple example: $\Phi(f)(n) = \mathrm{ite}(n = 0,\ 0,\ f(n-1) + n)$. $(f_{sol} = (1, 1, 0))$



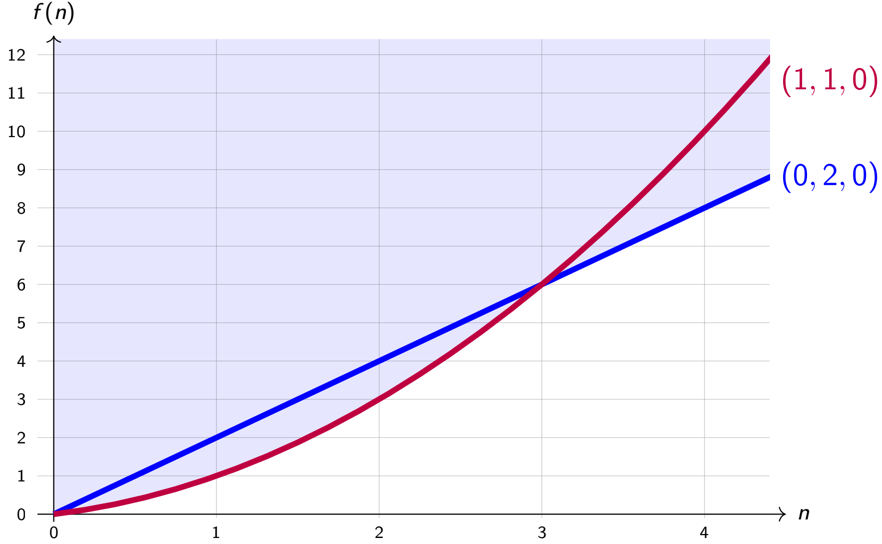
Polynomial bounds – Example

Very simple example: $\Phi(f)(n) = \mathrm{ite}(n = 0,\ 0,\ f(n-1) + n)$. $(f_{sol} = (1, 1, 0))$



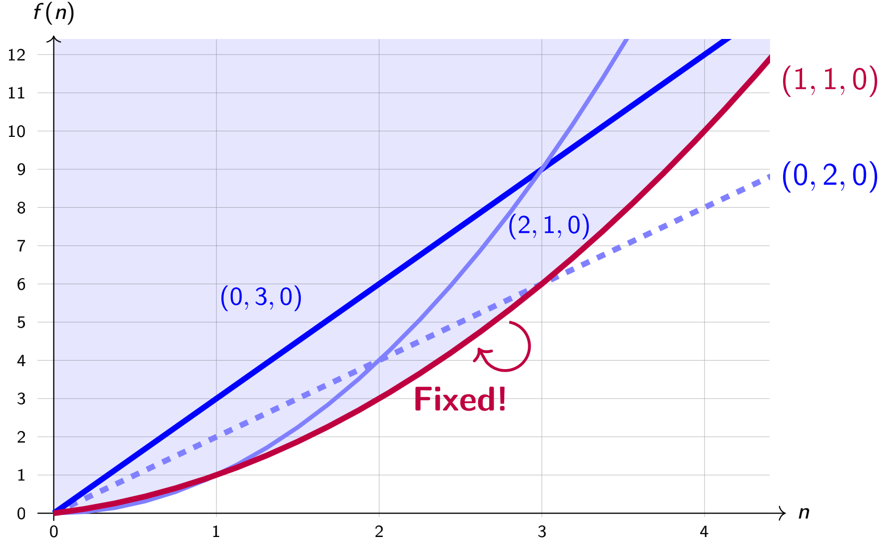
Polynomial bounds – Example

Very simple example: $\Phi(f)(n) = \mathrm{ite}(n = 0,\ 0,\ f(n-1) + n)$. $(f_{sol} = (1, 1, 0))$



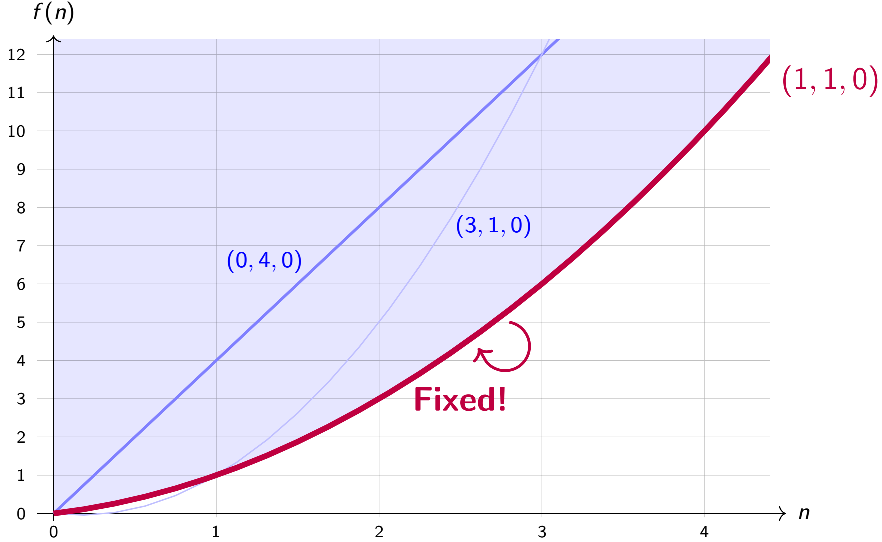
Polynomial bounds – Example

Very simple example: $\Phi(f)(n) = \mathrm{ite}(n = 0,\ 0,\ f(n-1) + n)$. $(f_{sol} = (1, 1, 0))$



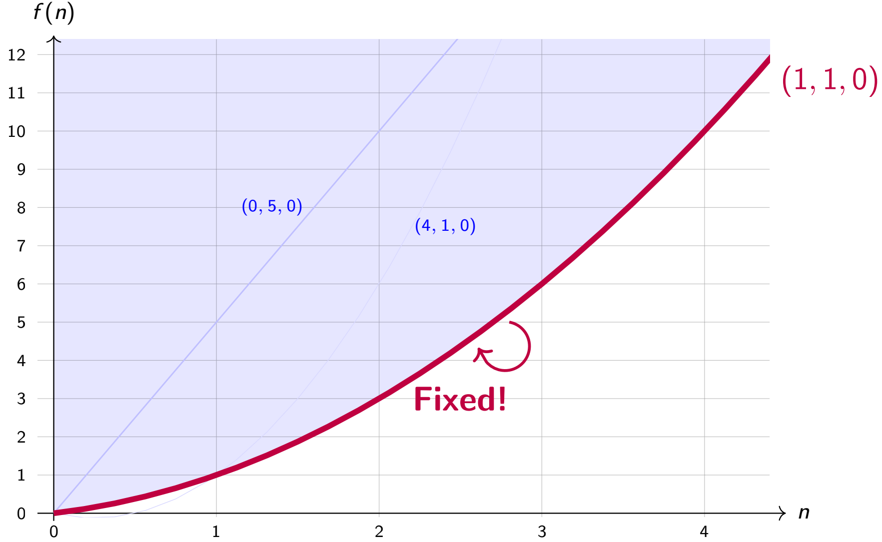
Polynomial bounds – Example

Very simple example: $\Phi(f)(n) = \mathrm{ite}(n = 0,\ 0,\ f(n-1) + n)$. $(f_{sol} = (1, 1, 0))$



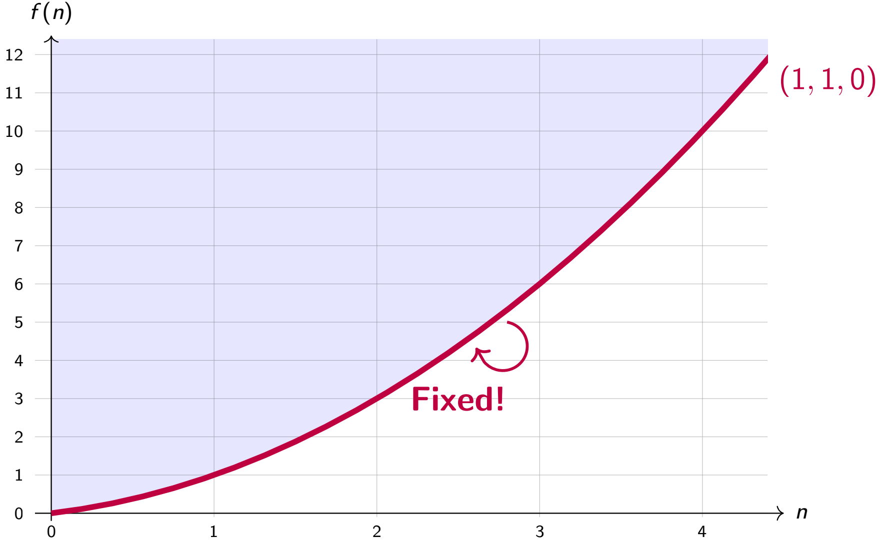
Polynomial bounds – Example

Very simple example: $\Phi(f)(n) = \mathrm{ite}(n = 0,\ 0,\ f(n-1) + n)$. $(f_{sol} = (1, 1, 0))$



Polynomial bounds – Example

Very simple example: $\Phi(f)(n) = \mathrm{ite}(n = 0,\ 0,\ f(n-1) + n)$. $(f_{sol} = (1, 1, 0))$



Beyond polynomials – exponential polynomials

Adding exponentials: the domain of **arithmetico-geometric sequences** [Feret05].

$$\beta : \mathbb{R}_+^5 \to \mathfrak{B} \subseteq (\mathbb{N} \to \mathbb{R}_+)$$

$$(M, a, b, a', b') \mapsto \left(n \mapsto [v \mapsto a \times v + b]\left([v \mapsto a' \times v + b']^{(n)}(M)\right)\right)$$

- Intended to be a classical "non-relational" value domain (for filters, floats,...): attach a function bound on each variable, parametric in the value of a local loop counter.
- Coordinate-wise $\sqsubseteq$, $\sqcup$, addition, ...
- Simple operations for scalar $+/\times$, `next` ($\sim$`Push`), projection, ...

Beyond polynomials – exponential polynomials

Adding exponentials: the domain of **arithmetico-geometric sequences** [Feret05].

$$\beta : \mathbb{R}_+^5 \to \mathfrak{B} \subseteq (\mathbb{N} \to \mathbb{R}_+)$$

$$(M, a, b, a', b') \mapsto \left(n \mapsto [v \mapsto a \times v + b]\left([v \mapsto a' \times v + b']^{(n)}(M)\right)\right)$$

- Intended to be a classical "non-relational" value domain (for filters, floats,...): attach a function bound on each variable, parametric in the value of a local loop counter.
- Coordinate-wise $\sqsubseteq$, $\sqcup$, addition, ...
- Simple operations for scalar $+/\times$, `next` ($\sim$`Push`), projection, ...

`Shift`-friendly exponentials: **Stirling numbers of the** 2^{nd} **kind** [Kahn&Hoffmann20].

- Replace basis $1, 2^n, 3^n, \ldots$ by $\left\{ {n \atop 1} \right\}, \left\{ {n \atop 2} \right\}, \left\{ {n \atop 3} \right\}, \left\{ {n \atop 4} \right\}, \ldots$

$$\left\{ {n \atop k} \right\} \sim_{n\to\infty} \frac{k^n}{k!} \qquad \left\{ {n+1 \atop k} \right\} = k\left\{ {n \atop k} \right\} + \left\{ {n \atop k-1} \right\}$$

- Extends well to multivariate settings.

Beyond polynomials – exponential polynomials

Adding exponentials: the domain of **arithmetico-geometric sequences** [Feret05].

$$\beta : \mathbb{R}_+^5 \to \mathfrak{B} \subseteq (\mathbb{N} \to \mathbb{R}_+)$$
$$(M, a, b, a', b') \mapsto \left(n \mapsto [v \mapsto a \times v + b]\left([v \mapsto a' \times v + b']^{(n)}(M)\right)\right)$$

- Intended to be a classical "non-relational" value domain (for filters, floats,...): attach a function bound on each variable, parametric in the value of a local loop counter.
- Coordinate-wise $\sqsubseteq$, $\sqcup$, addition, ...
- Simple operations for scalar $+/\times$, `next` ($\sim$`Push`), projection, ...

`Shift`-friendly exponentials: **Stirling numbers of the 2**$^{\text{nd}}$ **kind** [Kahn&Hoffmann20].

- Replace basis $1, 2^n, 3^n, \ldots$ by $\left\{ {n \atop 1} \right\}, \left\{ {n \atop 2} \right\}, \left\{ {n \atop 3} \right\}, \left\{ {n \atop 4} \right\}, \ldots$

$$\left\{ {n \atop k} \right\} \sim_{n\to\infty} \frac{k^n}{k!} \qquad\qquad \left\{ {n+1 \atop k} \right\} = k\left\{ {n \atop k} \right\} + \left\{ {n \atop k-1} \right\}$$

- Extends well to multivariate settings.

Going beyond — **exponential polynomials**.

- Replace basis $n \mapsto b^n \cdot n^k$ by $n \mapsto \left\{ {n \atop b} \right\}\binom{n}{k}$
- $[\![\texttt{Pop}]\!]^\sharp$ still works well, and $[\![\texttt{Push c}]\!]^\sharp$ can be synthesised as before.

$$\left\{ {n+1 \atop b} \right\}\binom{n+1}{k} = b\left\{ {n \atop b} \right\}\binom{n}{k} + b\left\{ {n \atop b} \right\}\binom{n}{k-1} + \left\{ {n \atop b-1} \right\}\binom{n}{k} + \left\{ {n \atop b-1} \right\}\binom{n}{k-1}$$

Discussion: Transfer Function Synthesis, Widenings, ...

Partial conclusion

For such **sequence domains** (useful for loops, equations, recursive functions, streams, ...)

- ➔ **Non-linearity is** natural and reasonably **easy** to achieve.
- ➔ Embrace **regularity of functions** vs arbitrary relations.
- ➔ Important enabler: **convexity** properties of $\left\{b \in \mathfrak{B} \,\middle|\, \bigwedge_n f(n) \leq b(n)\right\}$ (vs $\left\{\phi \in \mathfrak{C} \,\middle|\, \bigwedge_p \phi(p)\right\}$? Could we actually reuse some of these ideas for relations?).

Discussion: Transfer Function Synthesis, Widenings, ...

Partial conclusion

For such **sequence domains** (useful for loops, equations, recursive functions, streams, ...)

- ➔ **Non-linearity is** natural and reasonably **easy** to achieve.
- ➔ Embrace **regularity of functions** vs arbitrary relations.
- ➔ Important enabler: **convexity** properties of $\left\{b \in \mathfrak{B} \,\middle|\, \bigwedge_n f(n) \leq b(n)\right\}$
  (vs $\left\{\phi \in \mathfrak{C} \,\middle|\, \bigwedge_p \phi(p)\right\}$? Could we actually reuse some of these ideas for relations?).

Widenings

We have not discussed widenings. They are crucial. Our idea is

- Use **stability** ∇ + **thresholds**, parameter-wise (within $\mathfrak{B}$),
- On $\mathcal{P}_{\uparrow}(\mathfrak{B})$: bound the number of constraints, **drop/join above max cardinality**,
- Perhaps **default to "fixed constraints"**? "local fixpoints".

➔ Can we do better? Is this appropriate?

Discussion: Transfer function synthesis

- Still, **design of transfer functions** for a new $\mathfrak{B}$ is the **main bottleneck** in creation of such abstract domains (more burdensome than **other postfixpoint search methods, e.g. optimisation-based**).
- **Transfer function synthesis** may be the right tool.
  We were able to generate several automatically via **CAS + SMT**, and more still in interactive loops.

➔ Can we streamline this synthesis process?

- **Extra precision**: generate **transfers for small combinations of basic constructs**, $[\![s_1 \circ s_2 \circ \ldots]\!]^\sharp$ vs $\{[\![s_i]\!]^\sharp\}$.

Abstract Domains of Functions

Introducing more complex features:
Multiple variables, disjunctivity,
non-trivial recursive structure...

Multivariate functions

When the boundaries form a vector space $\mathfrak{B} = \bigoplus_k \mathbb{R} \cdot f_k$ extended with top element<u>s</u>, as often before, it is easy to move on from $\mathbb{N} \to \overline{\mathbb{R}}$ to $\mathbb{N}^d \to \overline{\mathbb{R}}$.
Do a **(tensor) product**, and do products of coordinate-wise orders (for parameters).

$$\boxed{\mathfrak{B}^{\otimes d} := \bigotimes_{i<d} \bigoplus_k \mathbb{R} \cdot f_k^{(i)} = \bigoplus_{k_1,\ldots,k_d} \mathbb{R} \cdot f_{k_1}^{(1)} \otimes \ldots \otimes f_{k_d}^{(d)}}$$

Example

Monovariate polynomials $\mathbb{R}[n]$ in the monomial basis $P(n) = \sum_k a_k n^k$ simply give rise to multivariate polynomials $\mathbb{R}[x,y] = (\mathbb{R}[n])^{\otimes 2}$ in the monomial basis $P(x,y) = \sum a_{k_x,k_y} x^{k_x} y^{k_y}$.

Transfer functions ($[\![\Diamond]\!]^\sharp$, $[\![\texttt{Cst}\ \ \texttt{c}]\!]^\sharp$, etc.) are easily extended.
We can similarly define (and compose) $[\![\texttt{Pop}_k]\!]^\sharp$ and $[\![\texttt{Push}_k\ c]\!]^\sharp$ on each dimension,

$$\text{e.g., } \texttt{Push}_y\ c : \quad f(x,y,z) \rightsquigarrow \mathrm{ite}\big(y = 0,\ c,\ f(x, y-1, z)\big).$$

Example

For polynomials, $1 \cdot (x+1)^2 yz = (x^2 + 2x + 1)yz = 1 \cdot x^2 yz + 2 \cdot xyz + 1 \cdot yz$. Similarly, in the binomial basis, $\binom{x+1}{k_x}\binom{y}{k_y}\binom{z}{k_z} = \binom{x}{k_x}\binom{y}{k_y}\binom{z}{k_z} + \binom{x}{k_x-1}\binom{y}{k_y}\binom{z}{k_z}$, so more generally

$$[\![\texttt{Pop}_x]\!]^\sharp\left(\sum a_{(k_x,k_y,k_z)}\binom{x}{k_x}\binom{y}{k_y}\binom{z}{k_z}\right) = \left\{\sum \left(a_{(\mathbf{k_x},k_y,k_z)} + a_{(\mathbf{k_x+1},k_y,k_z)}\right)\binom{x}{k_x}\binom{y}{k_y}\binom{z}{k_z}\right\}.$$

Piecewise functions — more complex control flows

Work in Progress

Example

```
while (0<i&&i<n){
  if(b)
    i++;
  else
    i--;
}
```

Simple loop (from [AFloresPhd17])

$$\Phi(f)(i,n,b) = \begin{cases} 0 & \text{if } i = 0 & (\mathcal{D}_1) \\ 0 & \text{if } i \geq n & (\mathcal{D}_2) \\ 1 + f(i+1,n,b) & \text{if } 0 < i < n \wedge b \geq 1 & (\mathcal{D}_3) \\ 1 + f(i-1,n,b) & \text{if } 0 < i < n \wedge b = 0 & (\mathcal{D}_4) \end{cases}$$

Corresponding cost equation

Piecewise functions — more complex control flows

Work in Progress

Example

```
while (0<i&&i<n){
  if(b)
    i++;
  else
    i--;
}
```

Simple loop (from [AFloresPhd17])

$$\Phi(f)(i,n,b) = \begin{cases} 0 & \text{if } i = 0 & (\mathcal{D}_1) \\ 0 & \text{if } i \geq n & (\mathcal{D}_2) \\ 1 + f(i+1,n,b) & \text{if } 0 < i < n \wedge b \geq 1 & (\mathcal{D}_3) \\ 1 + f(i-1,n,b) & \text{if } 0 < i < n \wedge b = 0 & (\mathcal{D}_4) \end{cases}$$

Corresponding cost equation

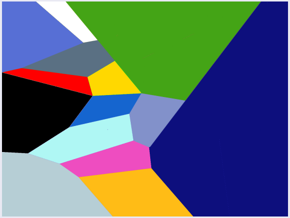
Piecewise function domains (closely related to BDT domains [Urban&Mine14])

Sets of $\mathfrak{B}$-**bounds by** $\mathcal{C}$-**cases**, where $\mathcal{C}$ is a constraint domains (e.g. polyhedra, etc.).

$$\mathbb{P}(\mathfrak{B},\mathcal{C}) \ni \{(F_1,c_1),\dots,(F_k,c_k)\}, \text{ s.t.}$$

- $F_i \in \mathcal{P}_{\uparrow}(\mathfrak{B})$,
- the $c_i \in \mathcal{C}$ partition the input space $\mathcal{D}$.

➔ Instead of one conjunction of bounds, piecewise conjunction of bounds.



Key primitive: ability to **merge cases**, i.e.
"Compute best $\mathfrak{B}$-approximation of $\mathfrak{B}$-cases by $\mathcal{C}$-parts."

Piecewise functions — more complex control flows

Work in Progress

Example

```
while (0<i&&i<n){
  if(b)
    i++;
  else
    i--;
}
```

Simple loop (from [AFloresPhd17])

$$\Phi(f)(i,n,b) = \begin{cases} 0 & \text{if } i = 0 & (\mathcal{D}_1) \\ 0 & \text{if } i \geq n & (\mathcal{D}_2) \\ 1 + f(i+1,n,b) & \text{if } 0 < i < n \wedge b \geq 1 & (\mathcal{D}_3) \\ 1 + f(i-1,n,b) & \text{if } 0 < i < n \wedge b = 0 & (\mathcal{D}_4) \end{cases}$$

Corresponding cost equation

$$\perp^\sharp = \begin{cases} \{0\} & (\mathcal{D}_1) \\ \{0\} & (\mathcal{D}_2) \\ \{0\} & (\mathcal{D}_3) \\ \{0\} & (\mathcal{D}_4) \end{cases}$$

Domains of piecewise functions

$\mathbb{P}(\mathfrak{B}, \mathcal{C}) \ni \{(F_1, c_1), \ldots, (F_k, c_k)\}$,

- $F_i \in \mathcal{P}_\uparrow(\mathfrak{B})$,
- the $c_i \in \mathcal{C}$ partition $\mathcal{D}$.

→ $\vec{n} \in \gamma(c_i) \Rightarrow \bigwedge_j f(\vec{n}) \leq f^\sharp_{i,j}(\vec{n})$

Key primitive:
"Compute best $\mathfrak{B}$-approximation of $\mathfrak{B}$-cases by $\mathcal{C}$-parts."

→ Refine at **interfaces between subdomains**, and **merge subcases**.

Piecewise functions — more complex control flows

Work in Progress

Example

```
while (0<i&&i<n){
  if(b)
    i++;
  else
    i--;
}
```

Simple loop (from [AFloresPhd17])

$$\Phi(f)(i,n,b) = \begin{cases} 0 & \text{if } i = 0 & (\mathcal{D}_1) \\ 0 & \text{if } i \geq n & (\mathcal{D}_2) \\ 1 + f(i+1,n,b) & \text{if } 0 < i < n \wedge b \geq 1 & (\mathcal{D}_3) \\ 1 + f(i-1,n,b) & \text{if } 0 < i < n \wedge b = 0 & (\mathcal{D}_4) \end{cases}$$

Corresponding cost equation

$$\perp^\sharp = \begin{cases} \{0\} & (\mathcal{D}_1) \\ \{0\} & (\mathcal{D}_2) \\ \{0\} & (\mathcal{D}_3) \\ \{0\} & (\mathcal{D}_4) \end{cases}$$

$$(\Phi^\sharp)(\perp^\sharp) = \begin{cases} \{0\} & (\mathcal{D}_1) \\ \{0\} & (\mathcal{D}_2) \\ \{1\} & (\mathcal{D}_3) \\ \{1\} & (\mathcal{D}_4) \end{cases}$$

Domains of piecewise functions

$\mathbb{P}(\mathfrak{B},\mathcal{C}) \ni \{(F_1,c_1),\dots,(F_k,c_k)\}$,

- $F_i \in \mathcal{P}_\uparrow(\mathfrak{B})$,
- the $c_i \in \mathcal{C}$ partition $\mathcal{D}$.

➔ $\vec{n} \in \gamma(c_i) \Rightarrow \bigwedge_j f(\vec{n}) \leq f^\sharp_{i,j}(\vec{n})$

Key primitive:
"Compute best $\mathfrak{B}$-approximation of $\mathfrak{B}$-cases by $\mathcal{C}$-parts."

➔ Refine at **interfaces between subdomains**, and **merge subcases**.

Piecewise functions — more complex control flows

Work in Progress

Example

```
while (0<i&&i<n){
  if(b)
    i++;
  else
    i--;
}
```

Simple loop (from [AFloresPhd17])

$$\Phi(f)(i,n,b) = \begin{cases} 0 & \text{if } i = 0 & (\mathcal{D}_1) \\ 0 & \text{if } i \geq n & (\mathcal{D}_2) \\ 1 + f(i+1,n,b) & \text{if } 0 < i < n \wedge b \geq 1 & (\mathcal{D}_3) \\ 1 + f(i-1,n,b) & \text{if } 0 < i < n \wedge b = 0 & (\mathcal{D}_4) \end{cases}$$

Corresponding cost equation

$$\perp^\sharp = \begin{cases} \{0\} & (\mathcal{D}_1) \\ \{0\} & (\mathcal{D}_2) \\ \{0\} & (\mathcal{D}_3) \\ \{0\} & (\mathcal{D}_4) \end{cases}$$

$$(\Phi^\sharp)(\perp^\sharp) = \begin{cases} \{0\} & (\mathcal{D}_1) \\ \{0\} & (\mathcal{D}_2) \\ \{1\} & (\mathcal{D}_3) \\ \{1\} & (\mathcal{D}_4) \end{cases}$$

$$(\Phi^\sharp)^{(2)}(\perp^\sharp) = ?$$

Domains of piecewise functions

$\mathbb{P}(\mathfrak{B}, \mathcal{C}) \ni \{(F_1, c_1), \ldots, (F_k, c_k)\}$,

- $F_i \in \mathcal{P}_\uparrow(\mathfrak{B})$,
- the $c_i \in \mathcal{C}$ partition $\mathcal{D}$.

→ $\vec{n} \in \gamma(c_i) \Rightarrow \bigwedge_j f(\vec{n}) \leq f^\sharp_{i,j}(\vec{n})$

Key primitive:
"Compute best $\mathfrak{B}$-approximation of $\mathfrak{B}$-cases by $\mathcal{C}$-parts."

→ Refine at **interfaces between subdomains**, and **merge subcases**.

Piecewise functions — more complex control flows

Work in Progress

Example

```
while (0<i&&i<n){
  if(b)
    i++;
  else
    i--;
}
```

Simple loop (from [AFloresPhd17])

$$\Phi(f)(i,n,b) = \begin{cases} 0 & \text{if } i = 0 & (\mathcal{D}_1) \\ 0 & \text{if } i \geq n & (\mathcal{D}_2) \\ 1 + f(i+1,n,b) & \text{if } 0 < i < n \wedge b \geq 1 & (\mathcal{D}_3) \\ 1 + f(i-1,n,b) & \text{if } 0 < i < n \wedge b = 0 & (\mathcal{D}_4) \end{cases}$$

Corresponding cost equation

$$(\Phi^\sharp)(\bot^\sharp) = \begin{cases} \{0\} & (\mathcal{D}_1) \\ \{0\} & (\mathcal{D}_2) \\ \{1\} & (\mathcal{D}_3) \\ \{1\} & (\mathcal{D}_4) \end{cases}$$

$$(\Phi^\sharp)^{(2)}(\bot^\sharp) = \,?$$

$$(\Phi \circ \Phi^\sharp)(\bot^\sharp) = \begin{cases} 0 & (\mathcal{D}_1) \\ 0 & (\mathcal{D}_2) \\ \begin{cases} 2 & \text{if } (i,n,b) \in \mathcal{D}_3 \wedge (i+1,n,b) \in \mathcal{D}_3 \\ 1 & \text{if } (i,n,b) \in \mathcal{D}_3 \wedge (i+1,n,b) \in \mathcal{D}_2 \end{cases} \\ \begin{cases} 1 & \text{if } (i,n,b) \in \mathcal{D}_4 \wedge (i-1,n,b) \in \mathcal{D}_1 \\ 2 & \text{if } (i,n,b) \in \mathcal{D}_4 \wedge (i-1,n,b) \in \mathcal{D}_4 \end{cases} \end{cases}$$

Domains of piecewise functions

$\mathbb{P}(\mathfrak{B},\mathcal{C}) \ni \{(F_1,c_1),\ldots,(F_k,c_k)\}$,

- $F_i \in \mathcal{P}_\uparrow(\mathfrak{B})$,
- the $c_i \in \mathcal{C}$ partition $\mathcal{D}$.

→ $\vec{n} \in \gamma(c_i) \Rightarrow \bigwedge_j f(\vec{n}) \leq f^\sharp_{i,j}(\vec{n})$

Key primitive:
"Compute best $\mathfrak{B}$-approximation of $\mathfrak{B}$-cases by $\mathcal{C}$-parts."

➔ Refine at **interfaces between subdomains**, and **merge subcases**.

Piecewise functions — more complex control flows

Work in Progress

Example

```
while (0<i&&i<n){
  if(b)
    i++;
  else
    i--;
}
```

Simple loop (from [AFloresPhd17])

$$\Phi(f)(i,n,b) = \begin{cases} 0 & \text{if } i = 0 & (\mathcal{D}_1) \\ 0 & \text{if } i \geq n & (\mathcal{D}_2) \\ 1 + f(i+1,n,b) & \text{if } 0 < i < n \wedge b \geq 1 & (\mathcal{D}_3) \\ 1 + f(i-1,n,b) & \text{if } 0 < i < n \wedge b = 0 & (\mathcal{D}_4) \end{cases}$$

Corresponding cost equation

$$(\Phi \circ \Phi^\sharp)(\perp^\sharp) = \begin{cases} 0 & (\mathcal{D}_1) \\ 0 & (\mathcal{D}_2) \\ \begin{cases} 2 & \texttt{if } (i,n,b) \in \mathcal{D}_3 \wedge (i+1,n,b) \in \mathcal{D}_3 \\ 1 & \texttt{if } (i,n,b) \in \mathcal{D}_3 \wedge (i+1,n,b) \in \mathcal{D}_2 \end{cases} \\ \begin{cases} 1 & \texttt{if } (i,n,b) \in \mathcal{D}_4 \wedge (i-1,n,b) \in \mathcal{D}_1 \\ 2 & \texttt{if } (i,n,b) \in \mathcal{D}_4 \wedge (i-1,n,b) \in \mathcal{D}_4 \end{cases} \end{cases}$$

$$(\Phi^\sharp)^{(2)}(\perp^\sharp) = \begin{cases} \{0\} & (\mathcal{D}_1) \\ \{0\} & (\mathcal{D}_2) \\ \{n-i\} & (\mathcal{D}_3) \\ \{i\} & (\mathcal{D}_4) \end{cases}$$

Domains of piecewise functions

$\mathbb{P}(\mathfrak{B}, \mathcal{C}) \ni \{(F_1, c_1), \dots, (F_k, c_k)\}$,

- $F_i \in \mathcal{P}_\uparrow(\mathfrak{B})$,
- the $c_i \in \mathcal{C}$ partition $\mathcal{D}$.

➔ $\vec{n} \in \gamma(c_i) \Rightarrow \bigwedge_j f(\vec{n}) \leq f^\sharp_{i,j}(\vec{n})$

Key primitive:
"Compute best $\mathfrak{B}$-approximation of $\mathfrak{B}$-cases by $\mathcal{C}$-parts."

➔ Refine at **interfaces between subdomains**, and **merge subcases**.

Piecewise functions — more complex control flows

Work in Progress

Example

```
while (0<i&&i<n){
  if(b)
    i++;
  else
    i--;
}
```

Simple loop (from [AFloresPhd17])

$$\Phi(f)(i,n,b) = \begin{cases} 0 & \text{if } i = 0 & (\mathcal{D}_1) \\ 0 & \text{if } i \geq n & (\mathcal{D}_2) \\ 1 + f(i+1,n,b) & \text{if } 0 < i < n \wedge b \geq 1 & (\mathcal{D}_3) \\ 1 + f(i-1,n,b) & \text{if } 0 < i < n \wedge b = 0 & (\mathcal{D}_4) \end{cases}$$

Corresponding cost equation

$$(\Phi \circ \Phi^\sharp)(\bot^\sharp) = \begin{cases} 0 & (\mathcal{D}_1) \\ 0 & (\mathcal{D}_2) \\ \begin{cases} 2 & \texttt{if } (i,n,b) \in \mathcal{D}_3 \wedge (i+1,n,b) \in \mathcal{D}_3 \\ 1 & \texttt{if } (i,n,b) \in \mathcal{D}_3 \wedge (i+1,n,b) \in \mathcal{D}_2 \end{cases} \\ \begin{cases} 1 & \texttt{if } (i,n,b) \in \mathcal{D}_4 \wedge (i-1,n,b) \in \mathcal{D}_1 \\ 2 & \texttt{if } (i,n,b) \in \mathcal{D}_4 \wedge (i-1,n,b) \in \mathcal{D}_4 \end{cases} \end{cases}$$

$$(\Phi^\sharp)^{(2)}(\bot^\sharp) = \begin{cases} \{0\} & (\mathcal{D}_1) \\ \{0\} & (\mathcal{D}_2) \\ \{n-i\} & (\mathcal{D}_3) \\ \{i\} & (\mathcal{D}_4) \end{cases}$$

$\leftrightarrow f_{sol}(i,n,b)$!

Domains of piecewise functions

$\mathbb{P}(\mathfrak{B},\mathcal{C}) \ni \{(F_1,c_1),\ldots,(F_k,c_k)\}$,

- $F_i \in \mathcal{P}_\uparrow(\mathfrak{B})$,
- the $c_i \in \mathcal{C}$ partition $\mathcal{D}$.

➔ $\vec{n} \in \gamma(c_i) \Rightarrow \bigwedge_j f(\vec{n}) \leq f^\sharp_{i,j}(\vec{n})$

Key primitive:
"Compute best $\mathfrak{B}$-approximation of $\mathfrak{B}$-cases by $\mathcal{C}$-parts."

➔ Refine at **interfaces between subdomains**, and **merge subcases**.

Continuous systems: example of ODE

Work in Progress

Abstract functions become particularly interesting in cases where we cannot (or do not want to) compute input/output samples, e.g. for **continuous systems**, with a function $f : \mathbb{R}_+ \to \overline{\mathbb{R}}$ defined as the solution of a **differential equation**. ➔ Towards $\mathfrak{B} \in \mathcal{P}(\mathbb{R}_+ \to \overline{\mathbb{R}})$?

$$v : \mathbb{R}_+ \to \mathbb{R}, \qquad v(0) = v_0, \qquad \dot{v} = -\alpha \cdot v^2 - \beta \cdot v + \gamma$$

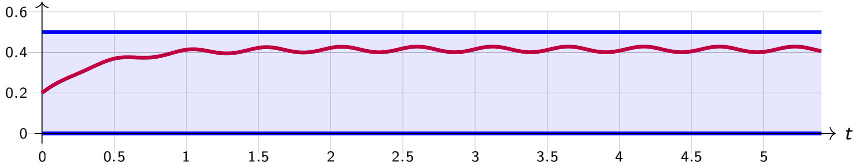
Non-standard analysis with infinitesimal ϵ, within **hyperreals** ${}^*\mathbb{R}$?

Or more simply a family of Euler schemas.

$$\boxed{\begin{aligned}&\Phi \in \mathrm{End}\left({}^*\mathbb{R}_+ \to \mathcal{I}({}^*\mathbb{R}), \dot{\sqsubseteq}_{\mathcal{I}}\right)\\ &:= f \mapsto t \mapsto \begin{cases}[v_0, v_0] & \text{if } t < \epsilon\\ f(t-\epsilon) - \alpha \cdot \epsilon \cdot f(t-\epsilon) \cdot_{\mathcal{I}} f(t-\epsilon) - \beta \cdot \epsilon \cdot f(t-\epsilon) + \epsilon \cdot \gamma & \text{if } t \geq \epsilon\end{cases}\end{aligned}}$$

For $\hat{f} : t \mapsto [0, M]$, $\Phi \hat{f} \mathrel{\dot{\sqsubseteq}_{\mathcal{I}}} \hat{f}$ whenever $x_0 \in [0, M]$, $\alpha, \beta, \gamma \geq 0$ and $M \geq \gamma/\beta$.

➔ Can also be applied with time-dependent $\alpha(t), \beta(t), \gamma(t)$, which is typically not solvable.



Example with $\alpha(t) \in [0, 2]$, $\beta = 2$, $\gamma = 1$, $v_0 = 0.2$.

Thank you!

Questions?

Order theory framework of equations [SAS24]

Equations $\leftrightarrow$ Operators Solutions $\leftrightarrow$ Fixpoints Bounds $\leftarrow$ Pre/Postfixp

Example

Search $f : \mathbb{N}^2 \to \mathbb{R}$ such that

$$f(n,c) = \begin{cases} f(n-1,0)+n+300 & \text{if } n>0 \text{ and } c \geq 100, \\ f(n-1,c+1)+n & \text{if } n>0 \text{ and } c<100, \\ c & \text{if } n=0, \end{cases}$$

$\rightarrow$

Example

Search $f : \mathbb{N}^2 \to \mathbb{R}$ such that $f = \Phi f$, i.e. $f \in \mathrm{Fixp}(\Phi)$, where

$$\Phi : (\mathbb{N}^2 \to \mathbb{R}) \to (\mathbb{N}^2 \to \mathbb{R})$$

$$f \mapsto (n,c) \mapsto \begin{cases} f(n-1,0)+n+300 & \text{if } n>0 \text{ and } c \geq 100, \\ f(n-1,c+1)+n & \text{if } n>0 \text{ and } c<100, \\ c & \text{if } n=0. \end{cases}$$

For a complete lattice, order $\mathcal{D} \to \mathbb{R}$ pointwise, and extend to $\overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$

Theorem

Let $\Phi : (\mathcal{D} \to \overline{\mathbb{R}}) \to (\mathcal{D} \to \overline{\mathbb{R}})$ be a **monotone equation***.*

- *If $f \in \mathrm{Postfp}(\Phi)$, i.e. $\Phi f \leq f$, then $\mathrm{lfp}\,\Phi \leq f$.*
- *If $f \in \mathrm{Prefp}(\Phi)$, i.e. $f \leq \Phi f$, then $f \leq \mathrm{gfp}\,\Phi$.*

Insight: cost equations are typically monotone for this pointwise order,
(and terminating $\rightsquigarrow \mathrm{lfp}\,\Phi = \mathrm{gfp}\,\Phi$).