Chair of Software Modeling and Verification (Computer Science 2)

RWTH AACHEN UNIVERSITY

A Framework for Online Optimization

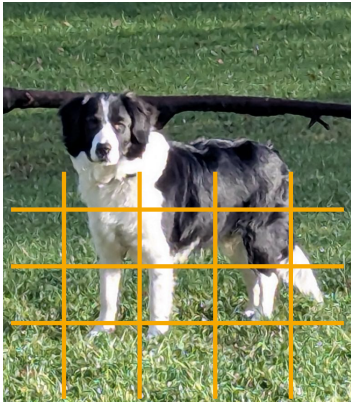
Ongoing Research

Emma Ahrens, Kevin Batz, Tobias Winkler, Joost-Pieter Katoen
RWTH Aachen University
26.11.2025

Online Optimization

Short-Sighted Cow*

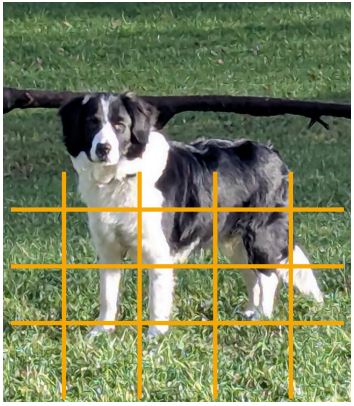
*Dog



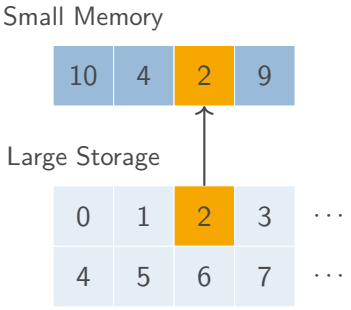
Online Optimization

Short-Sighted Cow*

*Dog



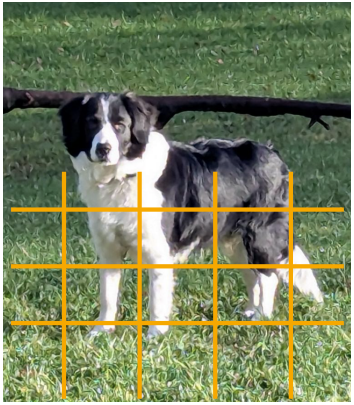
Paging Problem



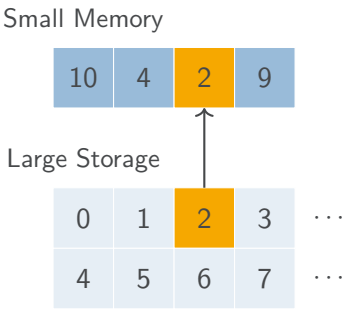
Online Optimization

Short-Sighted Cow*

*Dog

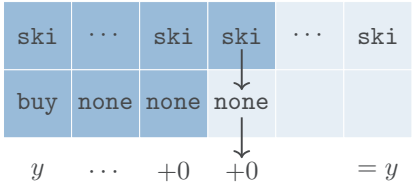


Paging Problem

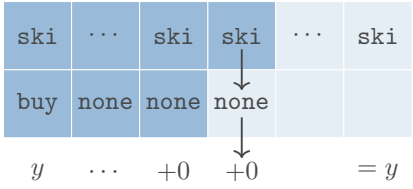


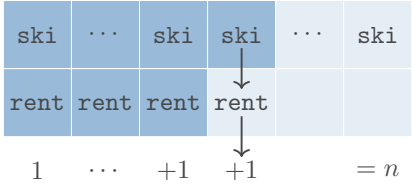
Ski-Rental Problem

Ski-Rental Problem

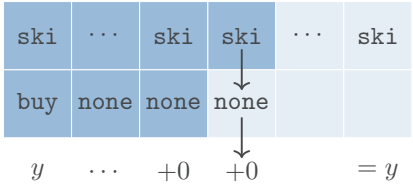


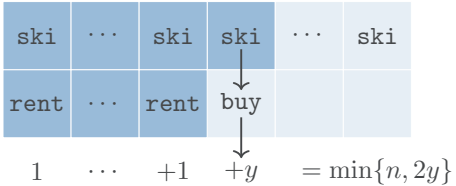
Ski-Rental Problem

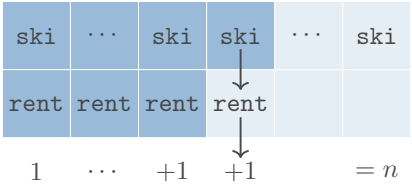




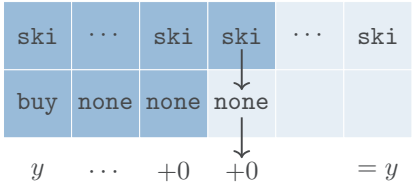
Ski-Rental Problem

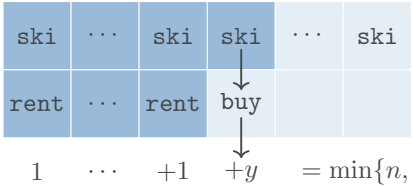


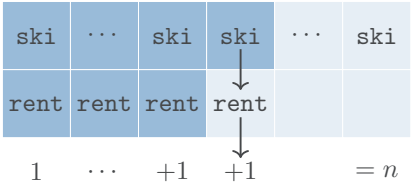


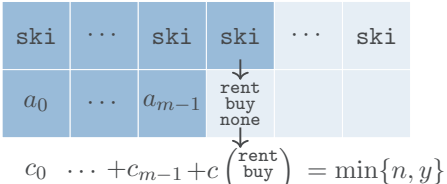


Ski-Rental Problem



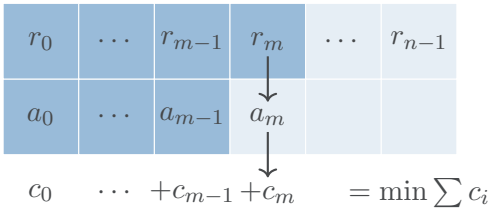






Online Optimization - Semantics

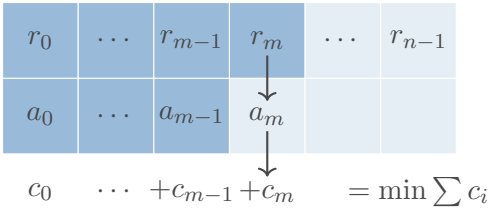
An online algorithm is one that receives a sequence of requests and, performs an immediate action in response to each request. Each sequence of requests and corresponding actions has an associated cost. *Richard M. Karp, 1992*



Online Optimization - Semantics

Optimization Problem

- requests $r_i \in \mathcal{R}$
- actions $a_i : \Sigma \to \Sigma \in \mathcal{A}$
- costs $c_i : \mathcal{R}^{m+1} \times \mathcal{A}^m \times \mathcal{A} \to \mathbb{N}$

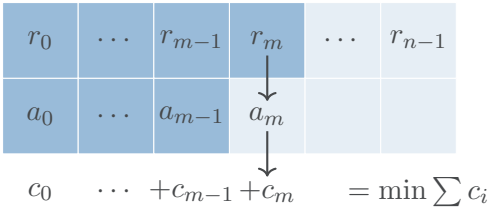


Online Optimization - Semantics

Optimization Problem

- requests $r_i \in \mathcal{R}$
- actions $a_i : \Sigma \to \Sigma \in \mathcal{A}$
- costs $c_i : \mathcal{R}^{m+1} \times \mathcal{A}^m \times \mathcal{A} \to \mathbb{N}$

Next Action $f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$



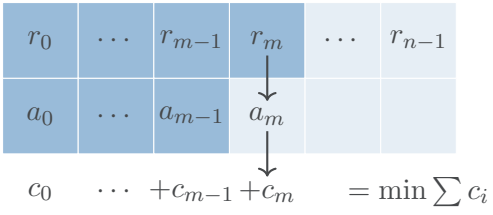
Online Optimization - Semantics

Optimization Problem

- requests $r_i \in \mathcal{R}$
- actions $a_i : \Sigma \to \Sigma \in \mathcal{A}$
- costs $c_i : \mathcal{R}^{m+1} \times \mathcal{A}^m \times \mathcal{A} \to \mathbb{N}$

Next Action $f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$

- specifies algorithm $\text{Alg} : \mathcal{R}^n \to 2^{\mathcal{A}^n}$



Online Optimization - Semantics

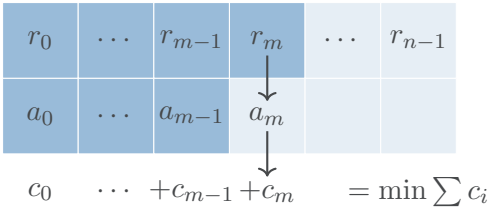
Optimization Problem

- requests $r_i \in \mathcal{R}$
- actions $a_i : \Sigma \to \Sigma \in \mathcal{A}$
- costs $c_i : \mathcal{R}^{m+1} \times \mathcal{A}^m \times \mathcal{A} \to \mathbb{N}$

Next Action $f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$

- specifies algorithm $\text{Alg} : \mathcal{R}^n \to 2^{\mathcal{A}^n}$

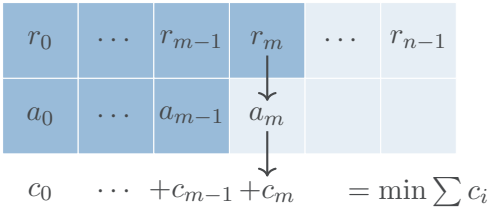
Competitive Ratio



Online Optimization - Semantics

Optimization Problem

- requests $r_i \in \mathcal{R}$
- actions $a_i : \Sigma \to \Sigma \in \mathcal{A}$
- costs $c_i : \mathcal{R}^{m+1} \times \mathcal{A}^m \times \mathcal{A} \to \mathbb{N}$



Next Action $f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$

- specifies algorithm $\mathsf{Alg} : \mathcal{R}^n \to 2^{\mathcal{A}^n}$

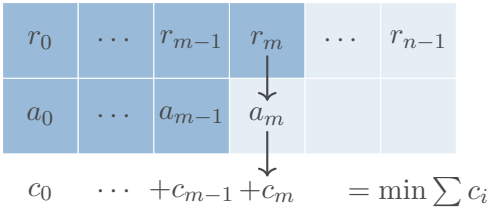
Competitive Ratio

- compares total costs $c : \mathcal{R}^n \times 2^{\mathcal{A}^n} \to \mathbb{N}^\infty, (R, \mathbf{A}) \mapsto \min_{A \in \mathbf{A}} \sum_{m<n} c(R, a_0 \cdots a_{m-1}, a_m)$

Online Optimization - Semantics

Optimization Problem

- requests $r_i \in \mathcal{R}$
- actions $a_i : \Sigma \to \Sigma \in \mathcal{A}$
- costs $c_i : \mathcal{R}^{m+1} \times \mathcal{A}^m \times \mathcal{A} \to \mathbb{N}$



Next Action $f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$

- specifies algorithm $\mathsf{Alg} : \mathcal{R}^n \to 2^{\mathcal{A}^n}$

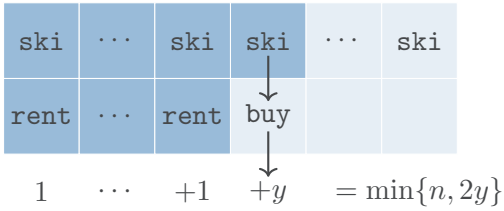
Competitive Ratio

- compares total costs $c : \mathcal{R}^n \times 2^{\mathcal{A}^n} \to \mathbb{N}^\infty, (R, \mathbf{A}) \mapsto \min_{A \in \mathbf{A}} \sum_{m<n} c(R, a_0 \cdots a_{m-1}, a_m)$
- of two algorithms via the *competitive ratio* $D := \sup_{R \in \mathcal{R}^n} \frac{c(R, \mathsf{On}(R,\epsilon))}{c(R, \mathsf{Off}(R,\epsilon))}$

Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$

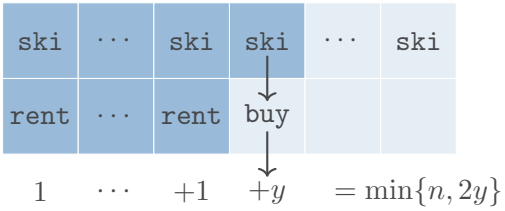


Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$

Next Action $f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$



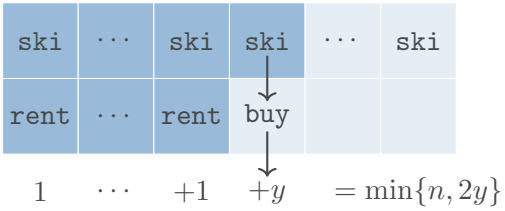
Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$

Next Action $f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$

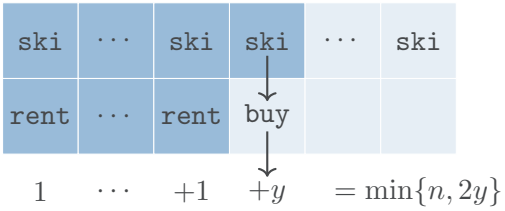
- specifies algorithm On : $\mathcal{R}^n \to 2^{\mathcal{A}^n}$, $R \mapsto \{(\texttt{rent}, \ldots, \texttt{rent}, \texttt{buy}, \texttt{none}, \ldots)\}$



Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$



Next Action $f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$

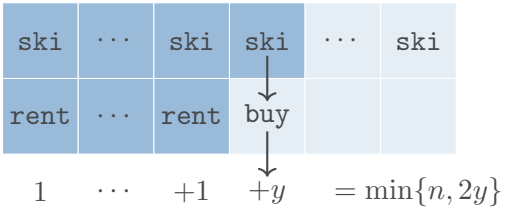
- specifies algorithm On : $\mathcal{R}^n \to 2^{\mathcal{A}^n}$, $R \mapsto \{(\texttt{rent}, \ldots, \texttt{rent}, \texttt{buy}, \texttt{none}, \ldots)\}$

Competitive Ratio

Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$



Next Action $f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$

- specifies algorithm On : $\mathcal{R}^n \to 2^{\mathcal{A}^n}$, $R \mapsto \{(\texttt{rent}, \ldots, \texttt{rent}, \texttt{buy}, \texttt{none}, \ldots)\}$

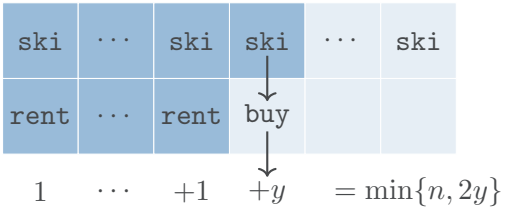
Competitive Ratio

- compares total costs $c = \min\{n, 2y\}$

Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$



Next Action $f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$

- specifies algorithm On : $\mathcal{R}^n \to 2^{\mathcal{A}^n}$, $R \mapsto \{(\texttt{rent}, \ldots, \texttt{rent}, \texttt{buy}, \texttt{none}, \ldots)\}$

Competitive Ratio

- compares total costs $c = \min\{n, 2y\}$
- of two algorithms via the *competitive ratio* $D := \sup_{R \in \mathcal{R}^n} \frac{\min\{n,2y\}}{\min n,y} \to 2$

Weighted Programming wGCL

$$
\begin{array}{rcl}
C & \to & x \coloneqq E \quad \mid \quad C_1 \mathbin{;} C_2 \quad \mid \quad \odot w \\
& \mid & \texttt{if}\,(\varphi)\,\{\,C_1\,\}\,\texttt{else}\,\{\,C_2\,\} \quad \mid \quad \texttt{while}\,(\varphi)\,\{C_1\} \quad \mid \quad \{\,C_1\,\} \oplus \{\,C_2\,\},
\end{array}
$$

where $x \in \texttt{Vars}$, E is an arithmetic expression, and $[\![w]\!] \in \mathbb{S}$

Weighted Programming wGCL

$$C \;\to\; x \coloneqq E \;\mid\; C_1 \fatsemi C_2 \;\mid\; \odot w$$
$$\mid\; \texttt{if}\,(\varphi)\,\{C_1\}\,\texttt{else}\,\{C_2\} \;\mid\; \texttt{while}\,(\varphi)\,\{C_1\} \;\mid\; \{C_1\} \oplus \{C_2\},$$

where $x \in \texttt{Vars}$, E is an arithmetic expression, and $[\![w]\!] \in \mathbb{S}$

Semiring $\mathbb{S} = (S, \oplus, \odot, \mathbb{0}, \mathbb{1})$ satisfies

- $(S, \oplus, \mathbb{0})$ is a commutative monoid, $(S, \odot, \mathbb{1})$ is a monoid,

Weighted Programming wGCL

$$C \;\to\; x \coloneqq E \;\mid\; C_1 \mathbin{;} C_2 \;\mid\; \odot w$$
$$\mid\; \texttt{if}\,(\varphi)\,\{C_1\}\,\texttt{else}\,\{C_2\} \;\mid\; \texttt{while}\,(\varphi)\,\{C_1\} \;\mid\; \{C_1\} \oplus \{C_2\},$$

where $x \in \texttt{Vars}$, E is an arithmetic expression, and $[\![w]\!] \in \mathbb{S}$

Semiring $\mathbb{S} = (S, \oplus, \odot, \mathbb{0}, \mathbb{1})$ satisfies

- $(S, \oplus, \mathbb{0})$ is a commutative monoid, $(S, \odot, \mathbb{1})$ is a monoid,
- multiplication $\odot$ distributes over addition $\oplus$, and

Weighted Programming wGCL

$$C \;\to\; x \coloneqq E \;\mid\; C_1 \mathbin{;} C_2 \;\mid\; \odot w \;\mid\; \texttt{if}\,(\varphi)\,\{C_1\}\,\texttt{else}\,\{C_2\} \;\mid\; \texttt{while}\,(\varphi)\,\{C_1\} \;\mid\; \{C_1\} \oplus \{C_2\},$$

where $x \in \texttt{Vars}$, E is an arithmetic expression, and $[\![w]\!] \in \mathbb{S}$

Semiring $\mathbb{S} = (S, \oplus, \odot, \mathbb{0}, \mathbb{1})$ satisfies

- $(S, \oplus, \mathbb{0})$ is a commutative monoid, $(S, \odot, \mathbb{1})$ is a monoid,
- multiplication $\odot$ distributes over addition $\oplus$, and
- $\mathbb{0}$ annihilates every element, more concretely $a \odot \mathbb{0} = \mathbb{0} \odot a = \mathbb{0}$.

Weighted Programming wGCL

$$
\begin{array}{rcl|l|l}
C & \to & x \coloneqq E & C_1 \mathbin{;} C_2 & \odot w \\
 & | & \texttt{if}\,(\varphi)\,\{C_1\}\,\texttt{else}\,\{C_2\} & \texttt{while}\,(\varphi)\,\{C_1\} & \{C_1\} \oplus \{C_2\},
\end{array}
$$

where $x \in \texttt{Vars}$, E is an arithmetic expression, and $[\![w]\!] \in \mathbb{S}$

Semiring $\mathbb{S} = (S, \oplus, \odot, \mathbb{0}, \mathbb{1})$ satisfies

- $(S, \oplus, \mathbb{0})$ is a commutative monoid, $(S, \odot, \mathbb{1})$ is a monoid,
- multiplication $\odot$ distributes over addition $\oplus$, and
- $\mathbb{0}$ annihilates every element, more concretely $a \odot \mathbb{0} = \mathbb{0} \odot a = \mathbb{0}$.
- No inverse elements!

Weighted Programming wGCL

$$
\begin{array}{rcl|l|l}
C & \to & x \coloneqq E & C_1 \mathbin{;} C_2 & \odot w \\
 & | & \texttt{if}\,(\varphi)\,\{C_1\}\,\texttt{else}\,\{C_2\} & \texttt{while}\,(\varphi)\,\{C_1\} & \{C_1\} \oplus \{C_2\},
\end{array}
$$

where $x \in \texttt{Vars}$, E is an arithmetic expression, and $[\![w]\!] \in \mathbb{S}$

Semiring $\mathbb{S} = (S, \oplus, \odot, \mathbb{0}, \mathbb{1})$ satisfies

- $(S, \oplus, \mathbb{0})$ is a commutative monoid, $(S, \odot, \mathbb{1})$ is a monoid,
- multiplication $\odot$ distributes over addition $\oplus$, and
- $\mathbb{0}$ annihilates every element, more concretely $a \odot \mathbb{0} = \mathbb{0} \odot a = \mathbb{0}$.
- No inverse elements!

Weighted Programming wGCL

$$C \;\to\; x \coloneqq E \;\mid\; C_1 \mathbin{;} C_2 \;\mid\; \odot w$$
$$\mid\; \texttt{if}\,(\varphi)\,\{C_1\}\,\texttt{else}\,\{C_2\} \;\mid\; \texttt{while}\,(\varphi)\,\{C_1\} \;\mid\; \{C_1\} \oplus \{C_2\},$$

where $x \in \texttt{Vars}$, E is an arithmetic expression, and $[\![w]\!] \in \mathbb{S}$

Semiring $\mathbb{S} = (S, \oplus, \odot, \mathbb{0}, \mathbb{1})$ satisfies

- $(S, \oplus, \mathbb{0})$ is a commutative monoid, $(S, \odot, \mathbb{1})$ is a monoid,
- multiplication $\odot$ distributes over addition $\oplus$, and
- $\mathbb{0}$ annihilates every element, more concretely $a \odot \mathbb{0} = \mathbb{0} \odot a = \mathbb{0}$.
- No inverse elements!

An example is the tropical semiring $(\mathbb{N}^\infty, \min, +, \infty, 0)$.

Weighted Programming wGCL

$$
\begin{array}{rcl}
C & \to & x \coloneqq E \quad \mid \quad C_1 \mathbin{;} C_2 \quad \mid \quad \odot w \\
& \mid & \texttt{if}\,(\varphi)\,\{C_1\}\,\texttt{else}\,\{C_2\} \quad \mid \quad \texttt{while}\,(\varphi)\,\{C_1\} \quad \mid \quad \{C_1\} \oplus \{C_2\},
\end{array}
$$

where $x \in \texttt{Vars}$, E is an arithmetic expression, and $[\![w]\!] \in \mathbb{S}$

Semiring $\mathbb{S} = (S, \oplus, \odot, \mathbb{0}, \mathbb{1})$ satisfies

- $(S, \oplus, \mathbb{0})$ is a commutative monoid, $(S, \odot, \mathbb{1})$ is a monoid,
- multiplication $\odot$ distributes over addition $\oplus$, and
- $\mathbb{0}$ annihilates every element, more concretely $a \odot \mathbb{0} = \mathbb{0} \odot a = \mathbb{0}$.
- No inverse elements!

An example is the tropical semiring $(\mathbb{N}^\infty, \min, +, \infty, 0)$.

Omitted: ω-complete partial order, ω-continuity

Reasoning about Competitive Ratio

problem-specific proofs by hand

$$D := \sup_{R \in \mathcal{R}^n} \frac{c(R, \mathsf{Alg}(R, \epsilon))}{c(R, \mathsf{Alg}'(R, \epsilon))} = \sup_{R \in \mathcal{R}^n} \frac{?}{?}$$

framework for semi-automatic verification

Weakest Preweightings Semantics

Calculus to prove properties of weighted programs:

$$\left\{ \texttt{wp}[\![\texttt{Prog}]\!](w) \right\} \ \texttt{Prog} \ \left\{ w \right\}$$

Weakest Preweightings Semantics

Calculus to prove properties of weighted programs:

$$\left\{ \texttt{wp}[\![\texttt{Prog}]\!](w) \right\} \ \texttt{Prog} \ \left\{ w \right\}$$

$$\mathbb{S} = (S, \oplus, \odot, \mathbf{0}, \mathbf{1}): \quad \texttt{wp}[\![\odot\, w]\!](\mathbf{1}) =$$

Weakest Preweightings Semantics

Calculus to prove properties of weighted programs:

$$\left\{ \; \mathtt{wp}[\![\mathtt{Prog}]\!](w) \; \right\} \; \mathtt{Prog} \; \left\{ \; w \; \right\}$$

$$\mathbb{S} = (S, \oplus, \odot, \mathbf{0}, \mathbf{1}) : \quad \mathtt{wp}[\![\odot \, w]\!](\mathbf{1}) = w$$

Weakest Preweightings Semantics

Calculus to prove properties of weighted programs:

$$\left\{ \ \mathtt{wp}[\![\mathtt{Prog}]\!](w) \ \right\} \ \mathtt{Prog} \ \left\{ \ w \ \right\}$$

$$\begin{aligned} \mathbb{S} = (S, \oplus, \odot, \mathbf{0}, \mathbf{1}): \quad & \mathtt{wp}[\![\odot\, w]\!](\mathbf{1}) = w \\ (\mathbb{N}^\infty, \min, +, \infty, 0): \quad & \mathtt{wp}[\![\odot\, y]\!](0) = \end{aligned}$$

Weakest Preweightings Semantics

Calculus to prove properties of weighted programs:

$$\left\{ \texttt{wp}[\![\texttt{Prog}]\!](w) \right\} \ \texttt{Prog} \ \left\{ w \right\}$$

$$\begin{aligned} \mathbb{S} = (S, \oplus, \odot, \mathbf{0}, \mathbf{1}): &\quad \texttt{wp}[\![\odot\, w]\!](\mathbf{1}) = w \\ (\mathbb{N}^\infty, \min, +, \infty, 0): &\quad \texttt{wp}[\![\odot\, y]\!](0) = y \end{aligned}$$

Weakest Preweightings Semantics

Calculus to prove properties of weighted programs:

$$\left\{ \texttt{wp}[\![\texttt{Prog}]\!](w) \right\} \ \texttt{Prog} \ \left\{ w \right\}$$

$$\begin{aligned} \mathbb{S} = (S, \oplus, \odot, \mathbf{0}, \mathbf{1}): \quad & \texttt{wp}[\![\odot\, w]\!](\mathbf{1}) = w \\ (\mathbb{N}^\infty, \min, +, \infty, 0): \quad & \texttt{wp}[\![\odot\, y]\!](0) = y \\ & \texttt{wp}[\![s \coloneqq 0 \fatsemi \odot [s = 1] \cdot 0]\!](0) = \end{aligned}$$

Weakest Preweightings Semantics

Calculus to prove properties of weighted programs:

$$\left\{\ \mathtt{wp}[\![\mathtt{Prog}]\!](w)\ \right\}\ \mathtt{Prog}\ \left\{\ w\ \right\}$$

$$\begin{aligned}
\mathbb{S} = (S, \oplus, \odot, \mathbf{0}, \mathbf{1}): \quad & \mathtt{wp}[\![\odot\, w]\!](\mathbf{1}) = w \\
(\mathbb{N}^\infty, \min, +, \infty, 0): \quad & \mathtt{wp}[\![\odot\, y]\!](0) = y \\
& \mathtt{wp}[\![s \coloneqq 0 \fatsemi \odot [s = 1] \cdot 0]\!](0) = \infty
\end{aligned}$$

Weakest Preweightings Semantics

Calculus to prove properties of weighted programs:

$$\left\{\ \mathtt{wp}[\![\mathtt{Prog}]\!](w)\ \right\}\ \mathtt{Prog}\ \left\{\ w\ \right\}$$

$$\begin{aligned}
\mathbb{S} = (S, \oplus, \odot, \mathbb{0}, \mathbb{1}): \quad & \mathtt{wp}[\![\odot\, w]\!](\mathbb{1}) = w \\
(\mathbb{N}^\infty, \min, +, \infty, 0): \quad & \mathtt{wp}[\![\odot\, y]\!](0) = y \\
& \mathtt{wp}[\![s \coloneqq 0 \ ;\ \odot [s = 1] \cdot 0]\!](0) = \infty
\end{aligned}$$

- parametrized over semiring $\mathbb{S}$

Weakest Preweightings Semantics

Calculus to prove properties of weighted programs:

$$\left\{\ \mathtt{wp}[\![\mathtt{Prog}]\!](w)\ \right\}\ \mathtt{Prog}\ \left\{\ w\ \right\}$$

$$\begin{aligned}
\mathbb{S} = (S, \oplus, \odot, \mathbb{0}, \mathbb{1}): \quad & \mathtt{wp}[\![\odot\, w]\!](\mathbb{1}) = w \\
(\mathbb{N}^\infty, \min, +, \infty, 0): \quad & \mathtt{wp}[\![\odot\, y]\!](0) = y \\
& \mathtt{wp}[\![s \coloneqq 0 \mathbin{;} \odot [s = 1] \cdot 0]\!](0) = \infty
\end{aligned}$$

- parametrized over semiring $\mathbb{S}$
- undecidable in general

Weakest Preweightings Semantics

Calculus to prove properties of weighted programs:

$$\left\{\ \mathtt{wp}[\![\mathtt{Prog}]\!](w)\ \right\}\ \mathtt{Prog}\ \left\{\ w\ \right\}$$

$$\begin{aligned}
\mathbb{S} = (S, \oplus, \odot, \mathbb{0}, \mathbb{1}): \quad & \mathtt{wp}[\![\odot\, w]\!](\mathbb{1}) = w \\
(\mathbb{N}^\infty, \min, +, \infty, 0): \quad & \mathtt{wp}[\![\odot\, y]\!](0) = y \\
& \mathtt{wp}[\![s \coloneqq 0 \ ;\ \odot [s = 1] \cdot 0]\!](0) = \infty
\end{aligned}$$

- parametrized over semiring $\mathbb{S}$
- undecidable in general
- semi-automatic verification tools (like Caesar) exist for some instances of weighted programming (e.g. for the probabilistic and the tropical semiring)

Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$

Off $= \texttt{while}\ (m < n)\ \{$

$\quad m \coloneqq m + 1$

$\}$

Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$

$$
\begin{aligned}
&\mathsf{Off} = \texttt{while}\ (m < n)\ \{\\
&\qquad \texttt{if}\ (\texttt{true}) \to \{\ \ \}\\
&\qquad \oplus\ (\texttt{true}) \to \{\ \ \}\\
&\qquad \oplus\ (\texttt{true}) \to \{\ \ \}\\
&\qquad m \coloneqq m + 1\\
&\}
\end{aligned}
$$

Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$

$$
\begin{aligned}
\mathsf{Off} = \texttt{while}\ &(m < n)\ \{ \\
&\texttt{if}\ (\texttt{true}) \to \{\ \odot 1\ \fatsemi\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot y\ \fatsemi\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot [s = 1] \cdot 0\ \fatsemi\ \} \\
&m \coloneqq m + 1 \\
\}&
\end{aligned}
$$

Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$

$$
\begin{aligned}
\mathsf{Off} = \texttt{while}\ &(m < n)\ \{ \\
&\texttt{if}\ (\texttt{true}) \to \{\ \odot 1\ \texttt{;}\ \texttt{skip}\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot y\ \texttt{;}\ s \coloneqq 1\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot [s = 1] \cdot 0\ \texttt{;}\ \texttt{skip}\ \} \\
&m \coloneqq m + 1 \\
\}&
\end{aligned}
$$

Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$

Next Action

$f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$

- specifies algorithm $\mathsf{On} : \mathcal{R}^n \to 2^{\mathcal{A}^n}$, $R \mapsto \{(\texttt{rent}, \ldots, \texttt{rent}, \texttt{buy}, \texttt{none}, \ldots)\}$

$$
\begin{aligned}
\mathsf{Off} = \texttt{while}\ &(m < n)\ \{ \\
&\texttt{if}\ (\texttt{true}) \to \{\ \odot 1\ \texttt{;}\ \texttt{skip}\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot y\ \texttt{;}\ s \coloneqq 1\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot [s = 1] \cdot 0\ \texttt{;}\ \texttt{skip}\ \} \\
&m \coloneqq m + 1 \\
\}&
\end{aligned}
$$

Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$

Next Action

$f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$

- specifies algorithm On : $\mathcal{R}^n \to 2^{\mathcal{A}^n}$, $R \mapsto \{(\texttt{rent}, \ldots, \texttt{rent}, \texttt{buy}, \texttt{none}, \ldots)\}$

$$
\begin{aligned}
\mathsf{Off} = \texttt{while}\ &(m < n)\ \{ \\
&\texttt{if}\ (\texttt{true}) \to \{\ \odot 1\ \texttt{;}\ \texttt{skip}\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot y\ \texttt{;}\ s \coloneqq 1\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot [s = 1] \cdot 0\ \texttt{;}\ \texttt{skip}\ \} \\
&m \coloneqq m + 1 \\
\}&
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{On} = \texttt{while}\ &(m < n)\ \{ \\
&\\
&\\
&\\
&\\
&m \coloneqq m + 1 \\
\}&
\end{aligned}
$$

Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$

Next Action

$f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$

- specifies algorithm $\text{On} : \mathcal{R}^n \to 2^{\mathcal{A}^n}$, $R \mapsto \{(\texttt{rent}, \ldots, \texttt{rent}, \texttt{buy}, \texttt{none}, \ldots)\}$

$$
\begin{aligned}
\text{Off} = \texttt{while}\ (m < n)\ \{ & \\
& \texttt{if}\ (\texttt{true}) \to \{\ \odot 1 \fatsemi \texttt{skip}\ \} \\
& \oplus\ (\texttt{true}) \to \{\ \odot y \fatsemi s := 1\ \} \\
& \oplus\ (\texttt{true}) \to \{\ \odot [s = 1] \cdot 0 \fatsemi \texttt{skip}\ \} \\
& m := m + 1 \\
\}
\end{aligned}
$$

$$
\begin{aligned}
\text{On} = \texttt{while}\ (m < n)\ \{ & \\
& \texttt{if}\ (\qquad) \to \{ \qquad \} \\
& \oplus\ (\qquad) \to \{ \qquad \} \\
& \oplus\ (\qquad) \to \{ \qquad \} \\
& m := m + 1 \\
\}
\end{aligned}
$$

Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$

Next Action

$f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$

- specifies algorithm On : $\mathcal{R}^n \to 2^{\mathcal{A}^n}$, $R \mapsto \{(\texttt{rent}, \ldots, \texttt{rent}, \texttt{buy}, \texttt{none}, \ldots)\}$

$$
\begin{aligned}
\mathsf{Off} = \texttt{while}\ &(m < n)\ \{ \\
&\texttt{if}\ (\texttt{true}) \to \{\ \odot 1\ \texttt{;}\ \texttt{skip}\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot y\ \texttt{;}\ s \coloneqq 1\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot [s = 1] \cdot 0\ \texttt{;}\ \texttt{skip}\ \} \\
&m \coloneqq m + 1 \\
\}&
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{On} = \texttt{while}\ &(m < n)\ \{ \\
&\texttt{if}\ (\quad\quad) \to \{\ \odot 1\ \texttt{;}\quad\quad\quad\quad\quad\quad\ \} \\
&\oplus\ (\quad\quad) \to \{\ \odot y\ \texttt{;}\quad\quad\quad\quad\quad\quad\ \} \\
&\oplus\ (\quad\quad) \to \{\ \odot [s = 1] \cdot 0\ \texttt{;}\quad\quad\quad\quad\ \} \\
&m \coloneqq m + 1 \\
\}&
\end{aligned}
$$

Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$

Next Action

$f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$

- specifies algorithm On : $\mathcal{R}^n \to 2^{\mathcal{A}^n}$, $R \mapsto \{(\texttt{rent}, \ldots, \texttt{rent}, \texttt{buy}, \texttt{none}, \ldots)\}$

$$
\begin{aligned}
\mathsf{Off} = \texttt{while}\ &(m < n)\ \{ \\
&\texttt{if}\ (\texttt{true}) \to \{\ \odot 1\ \texttt{;}\ \texttt{skip}\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot y\ \texttt{;}\ s \coloneqq 1\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot [s = 1] \cdot 0\ \texttt{;}\ \texttt{skip}\ \} \\
&m \coloneqq m + 1 \\
\}&
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{On} = \texttt{while}\ &(m < n)\ \{ \\
&\texttt{if}\ (\quad) \to \{\ \odot 1\ \texttt{;}\ \texttt{skip}\ \texttt{;}\ \quad\} \\
&\oplus\ (\quad) \to \{\ \odot y\ \texttt{;}\ s \coloneqq 1\ \texttt{;}\ \quad\} \\
&\oplus\ (\quad) \to \{\ \odot [s = 1] \cdot 0\ \texttt{;}\ \texttt{skip}\ \texttt{;}\ \quad\} \\
&m \coloneqq m + 1 \\
\}&
\end{aligned}
$$

Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$

Next Action

$f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$

- specifies algorithm $\mathsf{On} : \mathcal{R}^n \to 2^{\mathcal{A}^n}$, $R \mapsto \{(\texttt{rent}, \ldots, \texttt{rent}, \texttt{buy}, \texttt{none}, \ldots)\}$

$$
\begin{aligned}
\mathsf{Off} = \texttt{while}\ &(m < n)\ \{ \\
&\texttt{if}\ (\texttt{true}) \to \{\ \odot 1\ \texttt{;}\ \texttt{skip}\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot y\ \texttt{;}\ s \coloneqq 1\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot [s = 1] \cdot 0\ \texttt{;}\ \texttt{skip}\ \} \\
&m \coloneqq m + 1 \\
\}&
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{On} = \texttt{while}\ &(m < n)\ \{ \\
&\texttt{if}\ (\quad\quad) \to \{\ \odot 1\ \texttt{;}\ \texttt{skip}\ \texttt{;}\ c \coloneqq c + 1\ \} \\
&\oplus\ (\quad\quad) \to \{\ \odot y\ \texttt{;}\ s \coloneqq 1\ \texttt{;}\ c \coloneqq c + 1\ \} \\
&\oplus\ (\quad\quad) \to \{\ \odot [s = 1] \cdot 0\ \texttt{;}\ \texttt{skip}\ \texttt{;}\ \texttt{skip}\ \} \\
&m \coloneqq m + 1 \\
\}&
\end{aligned}
$$

Ski-Rental Problem

Optimization Problem

- requests $\mathcal{R} = \{\texttt{ski}\}$
- actions $\mathcal{A} = \{\texttt{rent}, \texttt{buy}, \texttt{none}\}$
- costs $c_{\texttt{rent}} = 1$, $c_{\texttt{buy}} = y$, $c_{\texttt{none}} = [s = 1] \cdot 0$

Next Action

$f^{m<n} : \mathcal{R}^{m+1} \times \mathcal{A}^m \to 2^{\mathcal{A}}$

- specifies algorithm On : $\mathcal{R}^n \to 2^{\mathcal{A}^n}$, $R \mapsto \{(\texttt{rent}, \ldots, \texttt{rent}, \texttt{buy}, \texttt{none}, \ldots)\}$

$$
\begin{aligned}
\mathsf{Off} = \texttt{while}\ &(m < n)\ \{ \\
&\texttt{if}\ (\texttt{true}) \to \{\ \odot 1\ \texttt{;}\ \texttt{skip}\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot y\ \texttt{;}\ s \coloneqq 1\ \} \\
&\oplus\ (\texttt{true}) \to \{\ \odot [s = 1] \cdot 0\ \texttt{;}\ \texttt{skip}\ \} \\
&m \coloneqq m + 1 \\
\}&
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{On} = \texttt{while}\ &(m < n)\ \{ \\
&\texttt{if}\ (c < y) \to \{\ \odot 1\ \texttt{;}\ \texttt{skip}\ \texttt{;}\ c \coloneqq c + 1\ \} \\
&\oplus\ (c = y) \to \{\ \odot y\ \texttt{;}\ s \coloneqq 1\ \texttt{;}\ c \coloneqq c + 1\ \} \\
&\oplus\ (c > y) \to \{\ \odot [s = 1] \cdot 0\ \texttt{;}\ \texttt{skip}\ \texttt{;}\ \texttt{skip}\ \} \\
&m \coloneqq m + 1 \\
\}&
\end{aligned}
$$

Ski-Rental Problem: Competitive Ratio

$$
\begin{aligned}
&\mathsf{Off} = \texttt{while}\ (m < n)\ \{\\
&\quad \texttt{if}\ (\texttt{true}) \to \{\, \odot 1\ \texttt{;}\ \texttt{skip}\,\}\\
&\quad \oplus\ (\texttt{true}) \to \{\, \odot y\ \texttt{;}\ s \coloneqq 1\,\}\\
&\quad \oplus\ (\texttt{true}) \to \{\, \odot [s = 1] \cdot 0\ \texttt{;}\ \texttt{skip}\,\}\\
&\quad m \coloneqq m + 1\\
&\}.
\end{aligned}
$$

$$
\begin{aligned}
&\mathsf{On} = \texttt{while}\ (m < n)\ \{\\
&\quad \texttt{if}\ (c < y) \to \{\, \odot 1\ \texttt{;}\ \texttt{skip}\ \texttt{;}\ c \coloneqq c + 1\,\}\\
&\quad \oplus\ (c = y) \to \{\, \odot y\ \texttt{;}\ s \coloneqq 1\ \texttt{;}\ c \coloneqq c + 1\,\}\\
&\quad \oplus\ (c > y) \to \{\, \odot [s = 1] \cdot 0\ \texttt{;}\ \texttt{skip}\ \texttt{;}\ \texttt{skip}\,\}\\
&\quad m \coloneqq m + 1\\
&\}.
\end{aligned}
$$

Ski-Rental Problem: Competitive Ratio

$$
\begin{aligned}
&\mathsf{Off} = \texttt{while}\ (m < n)\ \{\\
&\quad \texttt{if}\ (\texttt{true}) \to \{\, \odot 1 \mathbin{;} \texttt{skip} \,\}\\
&\quad \oplus\ (\texttt{true}) \to \{\, \odot y \mathbin{;} s \coloneqq 1 \,\}\\
&\quad \oplus\ (\texttt{true}) \to \{\, \odot [s = 1] \cdot 0 \mathbin{;} \texttt{skip} \,\}\\
&\quad m \coloneqq m + 1\\
&\}.
\end{aligned}
$$

$$
\begin{aligned}
&\mathsf{On} = \texttt{while}\ (m < n)\ \{\\
&\quad \texttt{if}\ (c < y) \to \{\, \odot 1 \mathbin{;} \texttt{skip} \mathbin{;} c \coloneqq c + 1 \,\}\\
&\quad \oplus\ (c = y) \to \{\, \odot y \mathbin{;} s \coloneqq 1 \mathbin{;} c \coloneqq c + 1 \,\}\\
&\quad \oplus\ (c > y) \to \{\, \odot [s = 1] \cdot 0 \mathbin{;} \texttt{skip} \mathbin{;} \texttt{skip} \,\}\\
&\quad m \coloneqq m + 1\\
&\}.
\end{aligned}
$$

$$\mathtt{wp}[\![\mathsf{Off}]\!](\mathfrak{1}) = n \oplus y = \min\{n, y\}$$

$$\mathtt{wp}[\![\mathsf{On}]\!](\mathfrak{1}) = n \oplus 2y = \min\{n, 2y\}$$

Ski-Rental Problem: Competitive Ratio

$$\mathsf{Off} = \texttt{while}\ (m < n)\ \{$$
$$\quad \texttt{if}\ (\texttt{true}) \to \{\, \odot 1 \mathbin{;} \texttt{skip} \,\}$$
$$\quad \oplus\ (\texttt{true}) \to \{\, \odot y \mathbin{;} s := 1 \,\}$$
$$\quad \oplus\ (\texttt{true}) \to \{\, \odot [s = 1] \cdot 0 \mathbin{;} \texttt{skip} \,\}$$
$$\quad m := m + 1$$
$$\}.$$

$$\mathsf{On} = \texttt{while}\ (m < n)\ \{$$
$$\quad \texttt{if}\ (c < y) \to \{\, \odot 1 \mathbin{;} \texttt{skip} \mathbin{;} c := c + 1 \,\}$$
$$\quad \oplus\ (c = y) \to \{\, \odot y \mathbin{;} s := 1 \mathbin{;} c := c + 1 \,\}$$
$$\quad \oplus\ (c > y) \to \{\, \odot [s = 1] \cdot 0 \mathbin{;} \texttt{skip} \mathbin{;} \texttt{skip} \,\}$$
$$\quad m := m + 1$$
$$\}.$$

$$\mathsf{wp}[\![\mathsf{Off}]\!](\mathfrak{1}) = n \oplus y = \min\{n, y\}$$

$$\mathsf{wp}[\![\mathsf{On}]\!](\mathfrak{1}) = n \oplus 2y = \min\{n, 2y\}$$

$$D = \sup_{R \in \mathcal{R}^n} \frac{\sigma_R(n \oplus 2y)}{\sigma_R(n \oplus y)} = \sup_{n>0, y>1} \frac{\min\{n, 2y\}}{\min\{n, y\}} = 2$$

From Costs to Expected Costs

Consider the expectation semiring $\mathbb{S}_{\mathbb{E}} = ([0,1] \times \mathbb{R}^{\infty}_{\geq 0}, \oplus, \odot, (0,0), (1,0))$

From Costs to Expected Costs

Consider the expectation semiring $\mathbb{S}_{\mathbb{E}} = ([0,1] \times \mathbb{R}^{\infty}_{\geq 0}, \oplus, \odot, (0,0), (1,0))$ with operations

$$(p,c) \oplus (q,d) = (p+q, c+d) \quad \text{and} \quad (p,c) \odot (q,d) = (p \cdot q, p \cdot d + q \cdot c).$$

From Costs to Expected Costs

Consider the expectation semiring $\mathbb{S}_{\mathbb{E}} = ([0,1] \times \mathbb{R}^{\infty}_{\geq 0}, \oplus, \odot, (0,0), (1,0))$ with operations

$$(p,c) \oplus (q,d) = (p+q, c+d) \quad \text{and} \quad (p,c) \odot (q,d) = (p \cdot q, p \cdot d + q \cdot c).$$

- $\odot\,(p,0)$ weighs an execution by probability p

From Costs to Expected Costs

Consider the expectation semiring $\mathbb{S}_{\mathbb{E}} = ([0,1] \times \mathbb{R}^{\infty}_{\geq 0}, \oplus, \odot, (0,0), (1,0))$ with operations

$$(p,c) \oplus (q,d) = (p+q, c+d) \quad \text{and} \quad (p,c) \odot (q,d) = (p \cdot q, p \cdot d + q \cdot c).$$

- $\odot\,(p,0)$ weighs an execution by probability p
- $\odot\,(0,c)$ adds the costs c to an exection

From Costs to Expected Costs

Consider the expectation semiring $\mathbb{S}_{\mathbb{E}} = ([0,1] \times \mathbb{R}^{\infty}_{\geq 0}, \oplus, \odot, (0,0), (1,0))$ with operations

$$(p,c) \oplus (q,d) = (p+q, c+d) \quad \text{and} \quad (p,c) \odot (q,d) = (p \cdot q, p \cdot d + q \cdot c).$$

- $\odot\,(p,0)$ weighs an execution by probability p
- $\odot\,(0,c)$ adds the costs c to an exection

From Costs to Expected Costs

Consider the expectation semiring $\mathbb{S}_{\mathbb{E}} = ([0,1] \times \mathbb{R}^{\infty}_{\geq 0}, \oplus, \odot, (0,0), (1,0))$ with operations

$$(p,c) \oplus (q,d) = (p+q, c+d) \quad \text{and} \quad (p,c) \odot (q,d) = (p \cdot q, p \cdot d + q \cdot c).$$

- $\odot\,(p,0)$ weighs an execution by probability p
- $\odot\,(0,c)$ adds the costs c to an exection

Often, randomized online algorithms perform better than deterministic online algorithms!

Ski-Rental Problem: Competitive Ratio of Randomized Algorithm

$$
\begin{aligned}
&\mathsf{Off} = \texttt{while}\ (m < n)\ \{\\
&\quad \texttt{if}\ (\texttt{true}) \to \{\, \odot 1 \mathbin{;} \texttt{skip} \,\}\\
&\quad \oplus\ (\texttt{true}) \to \{\, \odot y \mathbin{;} s \coloneqq 1 \,\}\\
&\quad \oplus\ (\texttt{true}) \to \{\, \odot [s = 1] \cdot 0 \mathbin{;} \texttt{skip} \,\}\\
&\quad m \coloneqq m + 1\\
&\}
\end{aligned}
$$

$$
\begin{aligned}
&\mathtt{Rand} = \texttt{while}\ (m < n)\ \{\\
&\quad \texttt{if}\ (c < y) \to \{\, \odot (1 - p_m, 1 - p_m) \mathbin{;} c \coloneqq c + 1 \,\}\\
&\quad \oplus\ (c < y) \to \{\, \odot (p_m, p_m \cdot y) \mathbin{;} s \coloneqq 1 \mathbin{;} c \coloneqq y \,\}\\
&\quad \oplus\ (c \geq y) \to \{\, \odot [s = 0] \cdot (0, \infty) \mathbin{;} \texttt{skip} \,\}\\
&\quad m \coloneqq m + 1\\
&\}.
\end{aligned}
$$

Ski-Rental Problem: Competitive Ratio of Randomized Algorithm

$$
\begin{aligned}
&\mathsf{Off} = \texttt{while}\ (m < n)\ \{\\
&\quad \texttt{if}\ (\texttt{true}) \to \{\, \odot 1 \ \texttt{;}\ \texttt{skip} \,\}\\
&\quad \oplus\ (\texttt{true}) \to \{\, \odot y \ \texttt{;}\ s \coloneqq 1 \,\}\\
&\quad \oplus\ (\texttt{true}) \to \{\, \odot [s = 1] \cdot 0 \ \texttt{;}\ \texttt{skip} \,\}\\
&\quad m \coloneqq m + 1\\
&\}
\end{aligned}
$$

$$
\begin{aligned}
&\mathsf{Rand} = \texttt{while}\ (m < n)\ \{\\
&\quad \texttt{if}\ (c < y) \to \{\, \odot (1 - p_m, 1 - p_m) \ \texttt{;}\ c \coloneqq c + 1 \,\}\\
&\quad \oplus\ (c < y) \to \{\, \odot (p_m, p_m \cdot y) \ \texttt{;}\ s \coloneqq 1 \ \texttt{;}\ c \coloneqq y \,\}\\
&\quad \oplus\ (c \geq y) \to \{\, \odot [s = 0] \cdot (0, \infty) \ \texttt{;}\ \texttt{skip} \,\}\\
&\quad m \coloneqq m + 1\\
&\}.
\end{aligned}
$$

$$\mathtt{wp}[\![\mathsf{Off}]\!](\mathfrak{1}) = n \oplus y = \min\{n, y\}$$

$$\mathtt{wp}[\![\mathsf{On}]\!](\mathfrak{1}) = E(n, y)$$

Ski-Rental Problem: Competitive Ratio of Randomized Algorithm

$$
\begin{aligned}
&\mathsf{Off} = \texttt{while}\ (m < n)\ \{\\
&\quad \texttt{if}\ (\texttt{true}) \to \{\, \odot 1\ \texttt{;}\ \texttt{skip} \,\}\\
&\quad \oplus\ (\texttt{true}) \to \{\, \odot y\ \texttt{;}\ s \coloneqq 1 \,\}\\
&\quad \oplus\ (\texttt{true}) \to \{\, \odot [s = 1] \cdot 0\ \texttt{;}\ \texttt{skip} \,\}\\
&\quad m \coloneqq m + 1\\
&\}
\end{aligned}
$$

$$
\begin{aligned}
&\mathtt{Rand} = \texttt{while}\ (m < n)\ \{\\
&\quad \texttt{if}\ (c < y) \to \{\, \odot (1 - p_m, 1 - p_m)\ \texttt{;}\ c \coloneqq c + 1 \,\}\\
&\quad \oplus\ (c < y) \to \{\, \odot (p_m, p_m \cdot y)\ \texttt{;}\ s \coloneqq 1\ \texttt{;}\ c \coloneqq y \,\}\\
&\quad \oplus\ (c \geq y) \to \{\, \odot [s = 0] \cdot (0, \infty)\ \texttt{;}\ \texttt{skip} \,\}\\
&\quad m \coloneqq m + 1\\
&\}.
\end{aligned}
$$

$$\mathtt{wp}[\![\mathsf{Off}]\!](\mathfrak{I}) = n \oplus y = \min\{n, y\}$$

$$
\begin{aligned}
&\mathtt{wp}[\![\mathsf{On}]\!](\mathfrak{I}) = E(n, y)\\
&= \textstyle\sum_{c=0}^{\min\{n,y\}-1} (y + c) \cdot \tilde{p}_c + \left(1 - \sum_{c=0}^{\min\{n,y\}-1} \tilde{p}_c\right) \cdot n.
\end{aligned}
$$

Ski-Rental Problem: Competitive Ratio of Randomized Algorithm

$$
\begin{aligned}
&\mathsf{Off} = \texttt{while}\ (m < n)\ \{\\
&\quad \texttt{if}\ (\texttt{true}) \to \{\, \odot 1 \ \texttt{;}\ \texttt{skip} \,\}\\
&\quad \oplus\ (\texttt{true}) \to \{\, \odot y \ \texttt{;}\ s \coloneqq 1 \,\}\\
&\quad \oplus\ (\texttt{true}) \to \{\, \odot [s = 1] \cdot 0 \ \texttt{;}\ \texttt{skip} \,\}\\
&\quad m \coloneqq m + 1\\
&\}
\end{aligned}
$$

$$
\begin{aligned}
&\mathtt{Rand} = \texttt{while}\ (m < n)\ \{\\
&\quad \texttt{if}\ (c < y) \to \{\, \odot (1 - p_m, 1 - p_m) \ \texttt{;}\ c \coloneqq c + 1 \,\}\\
&\quad \oplus\ (c < y) \to \{\, \odot (p_m, p_m \cdot y) \ \texttt{;}\ s \coloneqq 1 \ \texttt{;}\ c \coloneqq y \,\}\\
&\quad \oplus\ (c \geq y) \to \{\, \odot [s = 0] \cdot (0, \infty) \ \texttt{;}\ \texttt{skip} \,\}\\
&\quad m \coloneqq m + 1\\
&\}.
\end{aligned}
$$

$$\mathtt{wp}[\![\mathsf{Off}]\!](\mathfrak{1}) = n \oplus y = \min\{n, y\}$$

$$
\begin{aligned}
&\mathtt{wp}[\![\mathsf{On}]\!](\mathfrak{1}) = E(n, y)\\
&= \textstyle\sum_{c=0}^{\min\{n,y\}-1} (y + c) \cdot \tilde{p}_c + \left(1 - \sum_{c=0}^{\min\{n,y\}-1} \tilde{p}_c\right) \cdot n.
\end{aligned}
$$

$$D = \sup_{R \in \mathcal{R}^n} \frac{\mathtt{wp}[\![\mathtt{Prog}]\!](\mathfrak{1})(\sigma_R)}{\mathtt{wp}[\![\mathtt{Off}]\!](\mathfrak{1})(\sigma_R)} = \sup_{n>0, y>1} \frac{E(n, y)}{\min\{n, y\}} = \frac{e}{e - 1} \approx 1.58$$

Using Weakest Preweightings for Competitive Ratio

Online Optimization:

- for each optimization problem manual proofs by hand

Using Weakest Preweightings for Competitive Ratio

Online Optimization:

- for each optimization problem manual proofs by hand

Translation into Weighted Programming:

Using Weakest Preweightings for Competitive Ratio

Online Optimization:

- for each optimization problem manual proofs by hand

Translation into Weighted Programming:

- uniform framework

Using Weakest Preweightings for Competitive Ratio

Online Optimization:

- for each optimization problem manual proofs by hand

Translation into Weighted Programming:

- uniform framework
- semi-automatic verification via tools

Using Weakest Preweightings for Competitive Ratio

Online Optimization:

- for each optimization problem manual proofs by hand

Translation into Weighted Programming:

- uniform framework
- semi-automatic verification via tools
- further research in this area applies directly

Using Weakest Preweightings for Competitive Ratio

Online Optimization:

- for each optimization problem manual proofs by hand

Translation into Weighted Programming:

- uniform framework
- semi-automatic verification via tools
- further research in this area applies directly

problem-specific proofs by hand

$$D := \sup_{R \in \mathcal{R}^n} \frac{c(R, \mathsf{Alg}(R, \epsilon))}{c(R, \mathsf{Alg}'(R, \epsilon))} = \sup_{R \in \mathcal{R}^n} \frac{\mathtt{wp}[\![\mathtt{Prog}]\!](\mathfrak{I})(\sigma_R)}{\mathtt{wp}[\![\mathtt{Prog}']\!](\mathfrak{I})(\sigma_R)}$$

framework for semi-automatic verification

Using Weakest Preweightings for Competitive Ratio

Type	Function	Program	Coherence
action $a_i \in \mathcal{A}$	$a_i : \Sigma \to \Sigma$	$\texttt{wGCL}(a_i)$	$\texttt{Obs}$
cost	$c_i : \Sigma \to \mathbb{S}$ with $A(\sigma_R) \mapsto c(R, A, a_i)$	$\texttt{WExp}(c_i)$	$\texttt{Obs}$
auxiliary action	$x_i : \Sigma \to \Sigma$	$\texttt{wGCL}(x_i)$	$\texttt{Aux}$
guard	$\varphi_i : \Sigma \to \{0, 1\}$	$\texttt{BExp}(\varphi_i)$	$\texttt{Obs} \cup \texttt{Aux}$

problem-specific proofs by hand

$$D := \sup_{R \in \mathcal{R}^n} \frac{c(R, \mathsf{Alg}(R, \epsilon))}{c(R, \mathsf{Alg}'(R, \epsilon))} = \sup_{R \in \mathcal{R}^n} \frac{\texttt{wp}[\![\texttt{Prog}]\!](\mathfrak{1})(\sigma_R)}{\texttt{wp}[\![\texttt{Prog}']\!](\mathfrak{1})(\sigma_R)}$$

framework for semi-automatic verification

Classical	Observable
Arithmetic Expressions AExp $E \to x \in \texttt{Vars} \mid n \in \mathbb{N} \mid E + E \mid E \cdot E \mid E \dot{-} E$	**Observable Arithmetic Expressions** AObs $E \to E_1 \mid \ldots \mid E_k \mid q \in \mathbb{N} \mid E + E \mid E \cdot E \mid E$ with $E_1, \ldots, E_k \in \texttt{AExp}$
Boolean Expressions BExp $\varphi \to E < E \mid \varphi \wedge \varphi \mid \neg\varphi$	**Observable Boolean Expressions** BObs $\varphi \to \varphi_1 \mid \ldots \mid \varphi_\ell \mid E < E \mid \varphi \wedge \varphi \mid \neg\varphi$ with $\varphi_1, \ldots, \varphi_\ell \in \texttt{BExp}$
Weighted Expressions WExp $w \to s \in \mathbb{S} \mid w \oplus w \mid w \odot w \mid E \cdot w \mid [\varphi] \cdot w$ with $a \in \texttt{AExp}$ and $\varphi \in \texttt{BExp}$	**Observable Weighted Expressions** WObs $w \to s \in \mathbb{S} \mid w \oplus w \mid w \odot w \mid E \cdot w \mid [\varphi] \cdot w$ with $E \in \texttt{AObs}$ and $\varphi \in \texttt{BObs}$
	$\texttt{BaseObs} = \{E_1, \ldots, E_k, \varphi_1, \ldots, \varphi_\ell\}$

References I


Kevin Batz, Adrian Gallus, Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Tobias Winkler.
Weighted programming: a programming paradigm for specifying mathematical models.
Proc. ACM Program. Lang., 6(OOPSLA1):1–30, 2022.