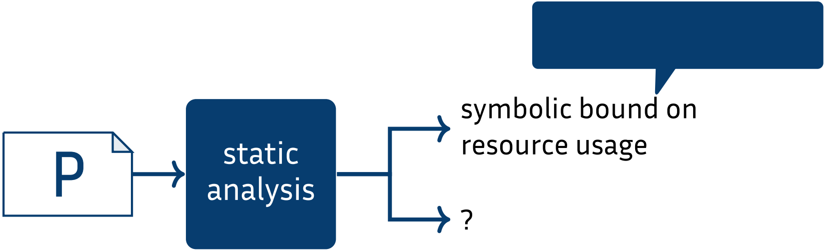
On Continuation-Passing Transformations and Expected Cost Analysis

Martin Avanzini and Gilles Barthe and Ugo Dal Lago

Joint PPS – PIHOC – DIAPASoN Workshop 2020

Static Resource Analysis



Motivations

- ★ integral part of software verification
- ★ embedded-systems
- ★ detect side-channel attacks
- ★ help programmers and compilers …

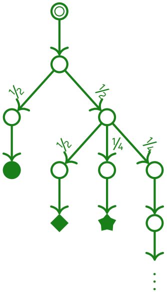
Solutions

- ★ recurrence relations
- ★ type systems
- ★ term rewriting
- ★ …

Conventional vs Probabilistic Programs

Sequential

Probabilistic



- ★ randomised algorithms
- ★ ~~statistical inference~~

Semantics

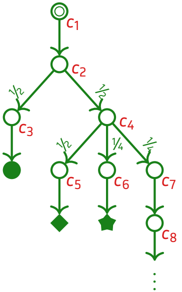
$[\![P]\!]\,(\circledcirc) = \bullet$

$[\![P]\!]\,(\circledcirc) = \{\bullet^{1/2}, \blacklozenge^{1/4}, \text{⬟}^{1/8}, \dots\}$

Conventional vs Probabilistic Programs

Sequential

Probabilistic



- ★ randomised algorithms
- ★ ~~statistical inference~~

Semantics

$\llbracket P \rrbracket\,(\circledcirc) = \text{●}$

$\llbracket P \rrbracket\,(\circledcirc) = \{\text{●}^{1/2}, \text{◆}^{1/4}, \text{⬟}^{1/8}, \dots\}$

Cost Analysis

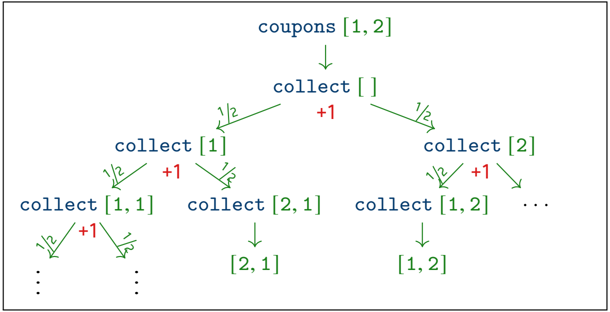
- ★ assign cost c_i to each operation
- ★ total cost of computation given by sum of all operation costs

$cost(\circledcirc) \overset{?}{\leq} b(\circledcirc)$

$E[cost(\circledcirc)] \overset{?}{\leq} b(\circledcirc)$

Example: Coupon Collector

```
coupons : [Coupons] → [Coupons]
let coupons cs =
  letrec collect os =
    if cs ⊆ os
    then os
    else collect (draw(cs)✓ :: os)
  in collect [ ]
```



Recursion tree of coupons [1, 2].

- ★ potentially non-terminating

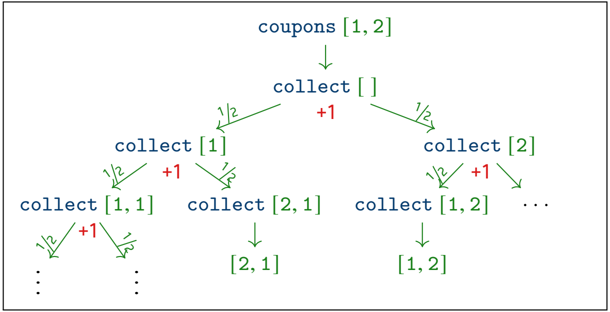
$$\texttt{coupons}\ [1,2] \rightsquigarrow \texttt{collect}\ [] \rightsquigarrow^{1/2} \texttt{collect}\ [1] \rightsquigarrow^{1/2} \texttt{collect}\ [1,1] \rightsquigarrow^{1/2} \texttt{collect}\ [1,1,1] \rightsquigarrow^{1/2} \cdots$$

- ★ expected cost finite

$$E[cost(\texttt{coupons}\ l)] \le |l| \cdot \sum_{i=1}^{|l|} 1/i \in O\left(|l| \cdot \log(|l|)\right)$$

Example: Coupon Collector

```
coupons : [Coupons] → [Coupons]
let coupons cs =
 letrec collect os =
  if cs ⊆ os
  then os
  else collect (draw(cs)✓ :: os)
 in collect [ ]
```



Recursion tree of coupons [1, 2].

- ★ potentially non-terminating

$\mathtt{coupons}\ [1,2] \rightsquigarrow \mathtt{collect}$, 1] $\rightsquigarrow^{1/2}$ . . .

How to formally derive such bounds?

- ★ expected cost finite

$$E[cost(\mathtt{coupons}\ \mathtt{l})] \leq |\mathtt{l}| \cdot \sum_{i=1}^{|\mathtt{l}|} 1/i \in O\left(|\mathtt{l}| \cdot \log(|\mathtt{l}|)\right)$$

Expected Cost (Runtime) Analysis

State of the Art

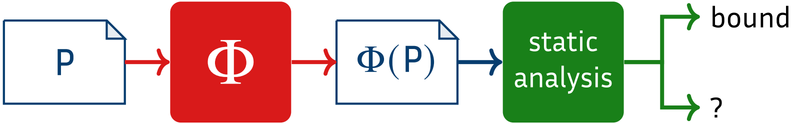
ARS
- ★ Lyapunov ranking functions (RF)
  [Bournez and Garnier'05]
  [A., Dal Lago & Yamada'19]

imperative programs
- ★ super martingale RF
  [Chakarov and Sankaranarayanan'13]
  [Ngo et al.'18]
  [Wang et al.'19]
- ★ pre-expectation calculi & upper invariants
  [Kaminski et al.'16]
  [A., Moser & Schaper'20]

functional programs
- ★ type systems
  [A., Dal Lago & Ghyselen'19]
  [Wang, Kahn & Hoffmann'20]

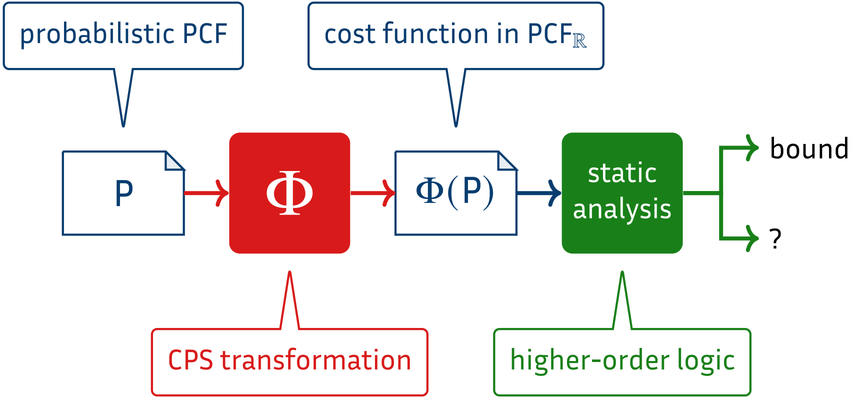
$$S \longrightarrow D \implies \eta(S) \geq \mathit{cost}(S) + E[\eta(D)]$$

Expected Cost Analysis via Transformations



- ⋆ Requirement: $cost(P) \leq cost_{\Phi}(\Phi(P))$ (or better, $cost(P) = cost_{\Phi}(\Phi(P))$)

Expected Cost Analysis via Transformations



- ⋆ Requirement: $\mathit{cost}(P) \leq \mathit{cost}_{\Phi}(\Phi(P))$ (or better, $\mathit{cost}(P) = \mathit{cost}_{\Phi}(\Phi(P))$)

From Programs to Cost Functions: Inspirations

1. Step-Counting Programs [Rosendahl'89]
   - $\mathcal{T}(E)$ is instrumentation of Lisp-program E with step-counter

$$[\![\mathcal{T}(E)]\!] : D^* \longrightarrow (D_\perp \times \mathbb{N}^\infty)$$

   - $\mathbb{N}^\infty$ ordered "vertically", thus ($n \leq_{\mathbb{N}^\infty} m$ iff $n \leq m$ or $m = \infty$)

$$[\![\texttt{letrec}\ f\,x = 1 + f\,x]\!] = [\![\lambda x.\infty]\!] \quad \text{and} \quad [\![\texttt{letrec}\ f\,x = f\,x]\!] = [\![\lambda x.0]\!]$$

$\Rightarrow$ nice for modeling non-terminating computations of finite cost

$\Rightarrow$ no negative costs

From Programs to Cost Functions: Inspirations

1. Step-Counting Programs [Rosendahl'89]
   - $\mathcal{T}(E)$ is instrumentation of Lisp-program E with step-counter

$$[\![\mathcal{T}(E)]\!] : D^* \longrightarrow (D_\bot \times \mathbb{N}^\infty)$$

   - $\mathbb{N}^\infty$ ordered "vertically", thus ($n \leq_{\mathbb{N}^\infty} m$ iff $n \leq m$ or $m = \infty$)

$$[\![\texttt{letrec}\ f\,x = 1 + f\,x]\!] = [\![\lambda x.\infty]\!] \quad \text{and} \quad [\![\texttt{letrec}\ f\,x = f\,x]\!] = [\![\lambda x.0]\!]$$

$\Longrightarrow$ nice for modeling non-terminating computations of finite cost

$\Longrightarrow$ no negative costs

2. Runtime Transformers [Kaminski et al.'16]
   - expected cost of probabilistic program P context dependent

$$\mathcal{T}(\mathcal{E}\big[P_1 \oplus_p P_2\big]) = p \cdot \mathcal{T}(\mathcal{E}[P_1]) + (1 - p) \cdot \mathcal{T}(\mathcal{E}[P_2])$$

$\Longrightarrow$ most naturally modeled in continuation passing style

$$[\![\mathcal{T}(P)]\!] : D^* \longrightarrow (D \longrightarrow \mathbb{R}^{+\infty}) \longrightarrow \mathbb{R}^{+\infty}$$

Input Language Λ_p

pure PCF
+ constants

Types $\sigma, \tau ::= \mathrm{B} \mid \sigma \rightarrow \tau$

Values $V, W ::= x \mid \lambda x.M \mid \mathtt{letrec}\, f x = M \mid \mathtt{c}(V_1, \ldots, V_n)$

Terms $M, N ::= V$

$\mid M \cdot N$

$\mid \mathtt{c}(M_1, \ldots, M_n)$

$\mid \mathtt{case}\ M\ \mathtt{of}\ \{\mathtt{c}(x_1, \ldots, x_k) \mapsto N_1 \mid y \mapsto N_2\}$

$\mid \mathtt{fn}(M_1, \ldots, M_n)$

$\mid M^{\checkmark}$ // cost annotation

$\mid \mathtt{d}(M_1, \ldots, M_n)$ // sampling primitives

Target Language $\Lambda_{\mathbb{R}}$

pure PCF
+ constants

Types $\sigma, \tau ::= \mathtt{B} \mid \mathtt{Real} \mid \sigma \to \tau$

Values $V, W ::= x \mid \lambda x.M \mid \mathtt{letrec}\, f \vec{x} = M \mid \mathtt{c}(V_1, \ldots, V_n) \mid \underline{r}\ (r \in \mathbb{R}^+)$

Terms $M, N ::= V$

$\mid M \cdot V$

$\mid \mathtt{c}(M_1, \ldots, M_n)$

$\mid \mathtt{case}\ V\ \mathtt{of}\ \{\mathtt{c}(x_1, \ldots, x_k) \mapsto N_1 \mid y \mapsto N_2\}$

$\mid \mathtt{fn}(M_1, \ldots, M_n)$

Target Language $\Lambda_{\mathbb{R}}$

pure PCF
+ constants

Types $\sigma, \tau ::= \mathrm{B} \mid \mathtt{Real} \mid \sigma \to \tau$

Values $V, W ::= x \mid \lambda x.M \mid \mathtt{letrec}\, f\vec{x} = M \mid \mathsf{c}(V_1, \ldots, V_n) \mid \underline{r}\ (r \in \mathbb{R}^+)$

Terms $M, N ::= V$

$$
\begin{aligned}
&\mid M \cdot V \\
&\mid \mathsf{c}(M_1, \ldots, M_n) \\
&\mid \mathtt{case}\ V\ \mathtt{of}\ \{\mathsf{c}(x_1, \ldots, x_k) \mapsto N_1 \mid y \mapsto N_2\} \\
&\mid \mathtt{fn}(M_1, \ldots, M_n)
\end{aligned}
$$

$$
\frac{\Gamma; f : \sigma_1 \to \cdots \to \sigma_k \to \mathtt{Real}; \ldots; x_i : \sigma_i, \ldots; \vdash M : \mathtt{Real}}{\Gamma \vdash \mathtt{letrec}\, f x_1 \cdots x_k = M : \mathtt{Real}}
$$

Target Language

Semantics

- ⋆ term $\Gamma \vdash M : \sigma$ interpreted as continuous function

$$[\![M]\!] : [[\![\Gamma]\!] \longrightarrow [\![\sigma]\!]] \ ,$$

where, particularly,

$$[\![\sigma_1 \to \cdots \to \sigma_k \to \texttt{Real}]\!] = [[\![\sigma_1]\!] \longrightarrow \cdots \longrightarrow [\![\sigma_k]\!] \longrightarrow \mathbb{R}^{+\infty}]$$

ordered pointwise by $\leq_{\mathbb{R}^{+\infty}}$, with bottom element $\_ \mapsto 0$ and top element $\_ \mapsto \infty$

$\Longrightarrow$ `Real`-valued primitives are monotone

Target Language
Semantics

- ⋆ term $\Gamma \vdash M : \sigma$ interpreted as continuous function

$$[\![M]\!] : [[\![\Gamma]\!] \longrightarrow [\![\sigma]\!]] \,,$$

where, particularly,

$$[\![\sigma_1 \to \cdots \to \sigma_k \to \mathtt{Real}]\!] = [[\![\sigma_1]\!] \longrightarrow \cdots \longrightarrow [\![\sigma_k]\!] \longrightarrow \mathbb{R}^{+\infty}]$$

ordered pointwise by $\leq_{\mathbb{R}^{+\infty}}$, with bottom element $\_ \mapsto 0$ and top element $\_ \mapsto \infty$

$\Rightarrow$ `Real`-valued primitives are monotone

- ⋆ recursion well-defined by monotone convergence theorem, even if “finite result” not computed in “finite time”

$$[\![\mathtt{letrec}\ \mathtt{fn}\, x = {}^{1}\!/_{x} + \mathtt{fn}\ (x+1) : \mathtt{Nat} \to \mathtt{Real}]\!] = n \mapsto \sum_{i=n}^{\infty} {}^{1}\!/_{i}$$

Expected Cost Transformer

call-by-value
one-pass CPS
[Danvy and
Nielson'03]

Mapping of Types

$$\mathtt{B}_i^{\dagger} \triangleq \mathtt{B}_i \qquad (\sigma \to \tau)^{\dagger} \triangleq \sigma^{\dagger} \to (\tau^{\dagger} \to \mathtt{Real}) \to \mathtt{Real}$$

Transformation on Values

$$x^{\dagger} \triangleq x \qquad (\lambda x.M)^{\dagger} \triangleq \lambda x k.\mathsf{ect}[M]\{k\}$$

$$\mathtt{c}(\vec{V})^{\dagger} \triangleq \mathtt{c}(\vec{V}^{\dagger}) \qquad (\mathtt{letrec}\, f x = M)^{\dagger} \triangleq \mathtt{letrec}\, f x\, k = \mathsf{ect}[M]\{k\}$$

Transformation on Terms

$$\mathsf{ect}[\cdot]\{\cdot\} : \Lambda_p(\sigma) \to \Lambda_{\mathbb{R}}(\sigma^{\dagger} \to \mathtt{Real}) \to \Lambda_{\mathbb{R}}(\mathtt{Real})$$

$$\mathsf{ect}[V]\{\kappa\} \triangleq \kappa \cdot V^{\dagger}$$

$$\mathsf{ect}[M \cdot N]\{\kappa\} \triangleq \mathsf{ect}[N]\{\lambda z.\mathsf{ect}[M]\{\lambda y.y \cdot z \cdot \kappa\}\}$$

$$\vdots$$

expected
cost

$$\mathsf{ect}[M^{\checkmark}]\{\kappa\} \triangleq \underline{1} + \mathsf{ect}[M]\{\kappa\}$$

$$\mathsf{ect}[\mathtt{d}(\vec{M})]\{\kappa\} = \mathsf{ect}[\vec{M}]\{\lambda \vec{z}.\mathsf{E}_{\mathtt{d}(\vec{z})}(\kappa)\} \ \text{where}$$

$$\mathsf{E}_{\mathtt{d}(\cdot)}(\cdot) : \mathtt{B}_1 \times \cdots \times \mathtt{B}_n \to (\mathtt{B} \to \mathtt{Real}) \to \mathtt{Real} \in \Lambda_{\mathbb{R}}^{+\infty} \ \text{for}$$

$$\mathtt{d} : \mathtt{B}_1 \times \cdots \times \mathtt{B}_n \to \mathtt{B}$$

Coupon Collector Revisited

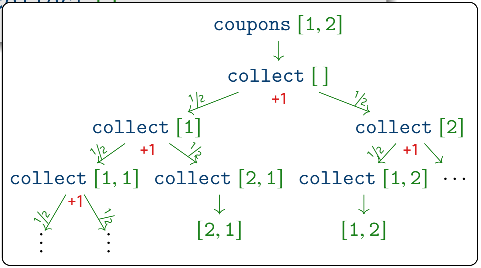
```
coupons : [Coupons] → [Coupons]
let coupons cs =
 letrec collect os =
  if cs ⊆ os
  then os
  else collect (draw(cs)$^{\checkmark}$ :: os)
 in collect [ ]
```

```
coupons : [Coupons]
    → ([Coupons] → Real) → Real
let coupons cs k =
 letrec collect os k =
  if cs ⊆ os
  then k os
  else $1 + \sum_{c \in cs} \frac{(\texttt{collect}\ (c :: os)\ k)}{|cs|}$
 in collect [ ] k

cost : [Coupons] → Real
let cost cs = coupons cs (λ_.0)
```

Coupon Collector Revisited

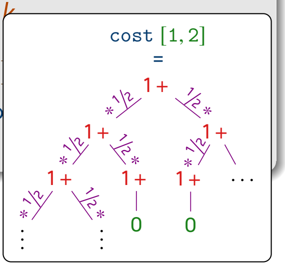
```
coupons : [Coupons] → [Coupons]
let coupons cs =
 letrec collect os =
  if cs ⊆ os
  then os
  else collect (draw(cs)✓ :: os)
 in collect [ ]
```



```
coupons : [Coupons]
      → ([Coupons] → Real) → Real
let coupons cs k =
 letrec collect os k =
  if cs ⊆ os
  then k os
  else 1 + $\sum_{c \in cs} \frac{(\texttt{collect}\ (c :: os)\ k)}{|cs|}$
 in collect [ ] k

cost : [Coupons

let cost cs = co
```

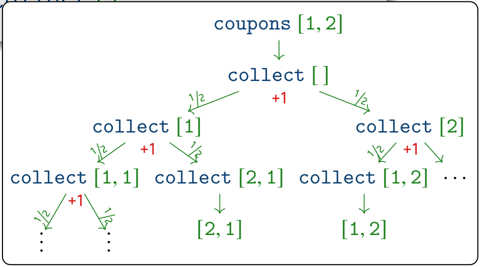


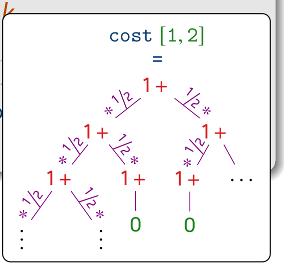
Coupon Collector Revisited

```
coupons : [Coupons] → [Coupons]
let coupons cs =
 letrec collect os =
  if cs ⊆ os
  then os
  else collect (draw(cs)✓ :: os)
 in collect [ ]
```

```
coupons : [Coupons]
     → ([Coupons] → Real) → Real
let coupons cs k =
 letrec collect os k =
  if cs ⊆ os
  then k os
  else $1 + \sum_{c \in cs} \frac{(\texttt{collect}\ (c :: os)\ k)}{|cs|}$
 in collect [ ] k

cost : [Coupons]
let cost cs = co
```





Theorem (Soundness)

For any $M \in \Lambda_p$, $E[\mathit{cost}(M)] = [\![\mathrm{ect}[M]\{\lambda\_.0\}]\!]$

A Higher-Order Logic for Reasoning About $\Lambda_{\mathbb{R}}^{+\infty}$

$$\Gamma \mid \Phi \vdash M : \sigma \mid \phi$$

- $\Gamma \vdash M : \sigma$
- Φ are assumptions on Γ
- ϕ is conclusion over distinguished variable $\mathbf{r} : \sigma$ representing M

Example

$$x : \texttt{Int}; f : \texttt{Int} \longrightarrow \texttt{Int} \mid x \leq 3; \forall z : \texttt{Int}.\, f\, z = z + 1 \vdash f\, x : \texttt{Int} \mid \mathbf{r} \leq 4$$

A Higher-Order Logic for Reasoning About $\Lambda_{\mathbb{R}}^{+\infty}$

$$\Gamma \mid \Phi \vdash M : \sigma \mid \phi$$

- $\Gamma \vdash M : \sigma$
- Φ are assumptions on Γ
- ϕ is conclusion over distinguished variable $\mathbf{r} : \sigma$ representing M

Example

$$x : \mathtt{Int}; f : \mathtt{Int} \longrightarrow \mathtt{Int} \mid x \leq 3; \forall z : \mathtt{Int}.\, f\, z = z + 1 \vdash f\, x : \mathtt{Int} \mid \mathbf{r} \leq 4$$

Theorem (Soundness)

If $\Gamma \mid \Phi \vdash M : \sigma \mid \phi$ *then for any environment* $\rho \in [\![\Gamma]\!]$,

$$[\![\Phi]\!]\,\rho \quad \Longrightarrow \quad [\![\phi]\!]\,\rho\{\mathbf{r} \mapsto [\![M]\!]\}$$

In particular $\vdash M : \sigma \mid \phi \Longrightarrow [\![\phi]\!]\,\{\mathbf{r} \mapsto [\![M]\!]\}$.

A Higher-Order Logic for Reasoning About $\Lambda_{\mathbb{R}}^{+\infty}$

$$\frac{\Gamma \vdash x : \sigma \qquad \Gamma \mid \Phi \vdash \phi[x/\mathbf{r}]}{\Gamma \mid \Phi \vdash x : \sigma \mid \phi}\,[\mathrm{Var}] \qquad \frac{\Gamma, x : \sigma \mid \Phi, \phi \vdash M : \tau \mid \psi}{\Gamma \mid \Phi \vdash \lambda x.M : \sigma \to \tau \mid \forall x : \sigma.\ \phi \Rightarrow \psi[\mathbf{r}\, x/\mathbf{r}]}\,[\mathrm{Abs}]$$

$$\frac{\Gamma \mid \Phi \vdash M : \sigma \to \tau \mid \forall x : \sigma.\ \phi[x/\mathbf{r}] \Rightarrow \psi[\mathbf{r}\, x/\mathbf{r}] \qquad \Gamma \mid \Phi \vdash V : \sigma \mid \phi}{\Gamma \mid \Phi \vdash M \cdot V : \tau \mid \psi}\,[\mathrm{App}]$$

$$\frac{\Gamma \mid \Phi \vdash M : \sigma \mid \phi \qquad \Gamma \mid \Phi \vdash \phi[x/\mathbf{r}] \Rightarrow \psi[x/\mathbf{r}]}{\Gamma \mid \Phi \vdash M : \sigma \mid \psi}\,[\mathrm{Sub}]$$

$$\frac{\begin{array}{c} \Gamma \mid \cdot \vdash \mathit{admissible}(\psi) \\ \Gamma, f : \vec{\sigma} \to \mathtt{Real}, \vec{x} : \vec{\sigma} \mid \Phi, \forall \vec{x} : \vec{\sigma}.\ \phi \Rightarrow \psi[f\, \vec{x}/\mathbf{r}], \phi \vdash M : \mathtt{Real} \mid \psi \end{array}}{\Gamma \mid \Phi \vdash \mathtt{letrec}\ f \vec{x} = M : \vec{\sigma} \to \mathtt{Real} \mid \forall \vec{x} : \vec{\sigma}.\ \phi \Rightarrow \psi[\mathbf{r}\, \vec{x}/\mathbf{r}]}\,[\mathrm{Letrec}]$$

$$\vdots$$

A Higher-Order Logic for Reasoning About $\Lambda_{\mathbb{R}}^{+\infty}$

$$\frac{\Gamma \vdash x : \sigma \qquad \Gamma \mid \Phi \vdash \phi[x/\mathsf{r}]}{\Gamma \mid \Phi \vdash x : \sigma \mid \phi}\,[\text{Var}]$$

$$\Gamma, x : \sigma \mid \Phi, \phi \vdash M : \tau \mid \psi$$

[Abs]

$$\Gamma \mid \Phi \vdash \lambda x$$

$$\Gamma \mid \Phi \vdash M : \sigma \to \tau \mid \forall x : \sigma.\ \phi[x/\mathsf{r}] \Rightarrow$$

$$\Gamma \mid \Phi \vdash M \cdot V :$$

$$\Gamma \mid \Phi \vdash M : \sigma \mid \phi \qquad \Gamma \mid \Phi$$

$$\Gamma \mid \Phi \vdash M : \sigma$$

$$\psi[0/\mathsf{r}] \wedge \bigwedge_{s \in S \subseteq \mathbb{R}^{+\infty}} \psi[s/\mathsf{r}] \Rightarrow \psi[\sup S/\mathsf{r}]$$

- ★ $\mathsf{r} \leq t$ but not $t \leq \mathsf{r}$
- ★ closed e.g. under $\wedge$ and $\vee$, but not $\neg$

⇒ upper but no lower-bounds

$$\frac{\begin{array}{c}\Gamma \mid \cdot \vdash \mathit{admissible}(\psi) \\ \Gamma, f : \vec{\sigma} \to \mathtt{Real}, \vec{x} : \vec{\sigma} \mid \Phi, \forall \vec{x} : \vec{\sigma}.\ \phi \Rightarrow \psi[f\,\vec{x}/\mathsf{r}], \phi \vdash M : \mathtt{Real} \mid \psi\end{array}}{\Gamma \mid \Phi \vdash \mathtt{letrec}\ f\vec{x} = M : \vec{\sigma} \to \mathtt{Real} \mid \forall \vec{x} : \vec{\sigma}.\ \phi \Rightarrow \psi[\mathsf{r}\,\vec{x}/\mathsf{r}]}\,[\text{Letrec}]$$

⋮

Analysing Coupon Collector

1. $\overbrace{cs : [\texttt{Coupons}]}^{\triangleq \Gamma} \mid \top$

$$\vdash \texttt{letrec collect}\ os\ k = M : \ldots \mid \forall os\, k.\ (os \subseteq cs \land \forall os'.k\ os' = 0) \Rightarrow \mathbf{r}\ os\ k \leq Q(os)$$

M

```
let coupons $cs$ $k$ =
 letrec collect $os$ $k$ =
  if $cs \subseteq os$
  then $k\ os$
  else $1 + \sum_{c \in cs} \frac{(\texttt{collect}\ (c :: os)\ k)}{|cs|}$
 in collect $[\,]$ $k$

let cost $cs$ = coupons $cs$ $(\lambda\_.0)$
```

Analysing Coupon Collector

1. $\overbrace{cs : [\texttt{Coupons}]}^{\triangleq \Gamma} \mid \top$

   $\vdash \texttt{letrec}\ \texttt{collect}\ os\ k = M : \ldots \mid \forall os\, k.\ (os \subseteq cs \wedge \forall os'.k\ os' = 0) \Rightarrow \mathbf{r}\ os\ k \leq Q(os)$

2. $\Gamma \mid \overbrace{\forall os\, k.\ (os \subseteq cs \wedge \forall os'.k\ os' = 0) \Rightarrow \texttt{collect}\ os\ k \leq Q(os); os \subseteq cs; \forall os'.k\ os' = 0}^{\triangleq \Phi}$

   $\vdash M : \texttt{Real} \mid \mathbf{r} \leq Q(os)$

$$\begin{array}{l}
\texttt{let}\ \texttt{coupons}\ cs\ k = \\
\quad \texttt{letrec}\ \texttt{collect}\ os\ k = \\
M \left[\begin{array}{l} \quad \texttt{if}\ cs \subseteq os \\ \quad \texttt{then}\ k\ os \\ \quad \texttt{else}\ 1 + \sum_{c \in cs} \frac{(\texttt{collect}\ (c :: os)\ k)}{|cs|} \end{array} \right. \\
\quad \texttt{in}\ \texttt{collect}\ [\,]\ k \\
\\
\texttt{let}\ \texttt{cost}\ cs = \texttt{coupons}\ cs\ (\lambda\_.0)
\end{array}$$

Analysing Coupon Collector

1. $\overbrace{cs : [\texttt{Coupons}]}^{\triangleq\Gamma} \mid \top$

   $\vdash \texttt{letrec collect}\ os\ k = M : \ldots \mid \forall os\, k.\ (os \subseteq cs \wedge \forall os'.k\ os' = 0) \Rightarrow \mathbf{r}\ os\ k \leq Q(os)$

2. $\Gamma \mid \overbrace{\forall os\, k.\ (os \subseteq cs \wedge \forall os'.k\ os' = 0) \Rightarrow \texttt{collect}\ os\ k \leq Q(os); os \subseteq cs; \forall os'.k\ os' = 0}^{\triangleq\Phi}$

   $\vdash M : \texttt{Real} \mid \mathbf{r} \leq Q(os)$

   2.1 $\Gamma \mid \Phi; cs \subseteq os \vdash k\ os : \texttt{Real} \mid \mathbf{r} \leq 0$

   2.2 $\Gamma \mid \Phi; \neg(cs \subseteq os) \vdash 1 + \sum_{c \in cs} \frac{(\texttt{collect}\ (c :: os)\ k)}{|cs|} : \texttt{Real} \mid \mathbf{r} \leq 1 + \sum_{c \in cs} \frac{Q(c :: os)}{|cs|}$

```
let coupons cs k =
  letrec collect os k =
    if cs ⊆ os
M   then k os
    else $1 + \sum_{c \in cs} \frac{(\texttt{collect}\ (c :: os)\ k)}{|cs|}$
  in collect [ ] k

let cost cs = coupons cs (λ_.0)
```

Analysing Coupon Collector

1. $\overbrace{cs : [\texttt{Coupons}]}^{\triangleq \Gamma} \mid \top$

   $\vdash \texttt{letrec collect}\ os\ k = M : \ldots \mid \forall os\, k.\ (os \subseteq cs \wedge \forall os'.k\ os' = 0) \Rightarrow \mathbf{r}\ os\ k \leq Q(os)$

2. $\Gamma \mid \overbrace{\forall os\, k.\ (os \subseteq cs \wedge \forall os'.k\ os' = 0) \Rightarrow \texttt{collect}\ os\ k \leq Q(os); os \subseteq cs; \forall os'.k\ os' = 0}^{\triangleq \Phi}$

   $\vdash M : \texttt{Real} \mid \mathbf{r} \leq Q(os)$

   2.1 $\Gamma \mid \Phi; cs \subseteq os \vdash k\ os : \texttt{Real} \mid \mathbf{r} \leq 0$

   2.2 $\Gamma \mid \Phi; \neg(cs \subseteq os) \vdash 1 + \sum_{c \in cs} \frac{(\texttt{collect}\ (c :: os)\ k)}{|cs|} : \texttt{Real} \mid \mathbf{r} \leq 1 + \sum_{c \in cs} \frac{Q(c :: os)}{|cs|}$

$\Rightarrow$ constraints 2.1 $\Phi; cs \subseteq os \vDash 0 \leq Q(os)$ and 2.2 $\Phi; \neg(cs \subseteq os) \vDash 1 + \sum_{c \in cs} \frac{Q(c :: os)}{|cs|} \leq Q(os)$

M

```
let coupons cs k =
 letrec collect os k =
  if cs ⊆ os
  then k os
  else 1 + $\sum_{c \in cs} \frac{(\texttt{collect}\ (c :: os)\ k)}{|cs|}$
 in collect [ ] k

let cost cs = coupons cs (λ_.0)
```

Analysing Coupon Collector

1. $\overbrace{cs : [\mathtt{Coupons}]}^{\triangleq\Gamma} \mid \top$
   $\vdash \mathtt{letrec}\ \mathtt{collect}\ os\ k = M : \ldots \mid \forall os\,k.\ (os \subseteq cs \wedge \forall os'.k\ os' = 0) \Rightarrow \mathbf{r}\ os\ k \leq Q(os)$

2. $\Gamma \mid \overbrace{\forall os\,k.\ (os \subseteq cs \wedge \forall os'.k\ os' = 0) \Rightarrow \mathtt{collect}\ os\ k \leq Q(os); os \subseteq cs; \forall os'.k\ os' = 0}^{\triangleq\Phi}$
   $\vdash M : \mathtt{Real} \mid \mathbf{r} \leq Q(os)$

   2.1 $\Gamma \mid \Phi; cs \subseteq os \vdash k\ os : \mathtt{Real} \mid \mathbf{r} \leq 0$

   2.2 $\Gamma \mid \Phi; \neg(cs \subseteq os) \vdash 1 + \sum_{c \in cs} \frac{(\mathtt{collect}\ (c :: os)\ k)}{|cs|} : \mathtt{Real} \mid \mathbf{r} \leq 1 + \sum_{c \in cs} \frac{Q(c :: os)}{|cs|}$

$\Rightarrow$ constraints 2.1 $\Phi; cs \subseteq os \vDash 0 \leq Q(os)$ and 2.2 $\Phi; \neg(cs \subseteq os) \vDash 1 + \sum_{c \in cs} \frac{Q(c :: os)}{|cs|} \leq Q(os)$

Define $Q(os) \triangleq |cs| \cdot H(|cs \setminus os|)$ where $H(n) = \sum_{i=1}^{n} {}^{1}/_{i}$

$$\begin{array}{l}
\mathtt{let\ coupons}\ cs\ k = \\
\quad \mathtt{letrec\ collect}\ os\ k = \\
M\left[\begin{array}{l} \quad \mathtt{if}\ cs \subseteq os \\ \quad \mathtt{then}\ k\ os \\ \quad \mathtt{else}\ 1 + \sum_{c \in cs} \frac{(\mathtt{collect}\ (c :: os)\ k)}{|cs|} \end{array}\right. \\
\quad \mathtt{in\ collect}\ [\,]\ k \\
\\
\mathtt{let\ cost}\ cs = \mathtt{coupons}\ cs\ (\lambda\_.0)
\end{array}$$

Analysing Coupon Collector

1. $\overbrace{cs : [\texttt{Coupons}]}^{\triangleq\Gamma} \mid \top$
   $\vdash \texttt{letrec collect}\ os\ k = M : \ldots \mid \forall os\,k.\ (os \subseteq cs \wedge \forall os'.k\,os' = 0) \Rightarrow \mathbf{r}\ os\ k \leq Q(os)$

2. $\Gamma \mid \overbrace{\forall os\,k.\ (os \subseteq cs \wedge \forall os'.k\,os' = 0) \Rightarrow \texttt{collect}\ os\ k \leq Q(os);\ os \subseteq cs;\ \forall os'.k\,os' = 0}^{\triangleq\Phi}$
   $\vdash M : \texttt{Real} \mid \mathbf{r} \leq Q(os)$

   2.1 $\Gamma \mid \Phi; cs \subseteq os \vdash k\ os : \texttt{Real} \mid \mathbf{r} \leq 0$

   2.2 $\Gamma \mid \Phi; \neg(cs \subseteq os) \vdash 1 + \sum_{c \in cs} \frac{(\texttt{collect}\ (c :: os)\ k)}{|cs|} : \texttt{Real} \mid \mathbf{r} \leq 1 + \sum_{c \in cs} \frac{Q(c :: os)}{|cs|}$

$\Rightarrow$ constraints 2.1 $\Phi; cs \subseteq os \vDash 0 \leq Q(os)$ and 2.2 $\Phi; \neg(cs \subseteq os) \vDash 1 + \sum_{c \in cs} \frac{Q(c :: os)}{|cs|} \leq Q(os)$

Define $Q(os) \triangleq |cs| \cdot H(|cs \setminus os|)$ where $H(n) = \sum_{i=1}^{n} {}^{1}/_{i}$

2.1 $os \subseteq cs \vDash 0 = Q(os)$

2.2 $os \subset cs \vDash 1 + \sum_{c \in cs} \frac{Q(c :: os)}{|cs|} = 1 + \sum_{c \in cs} H(|cs \setminus (c :: os)|)$

$$= 1 + |os| \cdot H(|cs \setminus os|) + |cs \setminus os| \cdot H(|cs \setminus os| - 1)$$

$$= |cs| \cdot H(|cs \setminus os|) = Q(os)$$

```
let coupons $cs$ $k$ =
  letrec collect $os$ $k$ =
      if $cs \subseteq os$
$M$   then $k\ os$
      else $1 + \sum_{c \in cs} \frac{(\texttt{collect}\ (c :: os)\ k)}{|cs|}$
  in collect [ ] $k$

let cost $cs$ = coupons $cs$ $(\lambda\_.0)$
```

Analysing Coupon Collector

1. $\overbrace{cs : [\texttt{Coupons}]}^{\triangleq\Gamma} \mid \top$

   $\vdash \texttt{letrec}\ \texttt{collect}\ os\ k = M : \ldots \mid \forall os\,k.\,(os \subseteq cs \wedge \forall os'.k\,os' = 0) \Rightarrow \mathbf{r}\ os\ k \le Q(os)$

2. $\Gamma \mid \overbrace{\forall os\,k.\,(os \subseteq cs \wedge \forall os'.k\,os' = 0) \Rightarrow \texttt{collect}\ os\ k \le Q(os); os \subseteq cs; \forall os'.k\,os' = 0}^{\triangleq\Phi}$

   $\vdash M : \texttt{Real} \mid \mathbf{r} \le Q(os)$

   2.1 $\Gamma \mid \Phi; cs \subseteq os \vdash k\ os : \texttt{Real} \mid \mathbf{r} \le 0$

   2.2 $\Gamma \mid \Phi; \neg(cs \subseteq os) \vdash 1 + \sum_{c \in cs} \frac{(\texttt{collect}\ (c :: os)\ k)}{|cs|} : \texttt{Real} \mid \mathbf{r} \le 1 + \sum_{c \in cs} \frac{Q(c :: os)}{|cs|}$

$\Rightarrow$ constraints 2.1 $\Phi; cs \subseteq os \vDash 0 \le Q(os)$ and 2.2 $\Phi; \neg(cs \subseteq os) \vDash 1 + \sum_{c \in cs} \frac{Q(c :: os)}{|cs|} \le Q(os)$

Define $Q(os) \triangleq |cs| \cdot H(|cs \setminus os|)$ where $H(n) = \sum_{i=1}^{n} {}^{1}\!/_{i}$

3. $\vdash \texttt{let}\ \texttt{cost}\ cs = \texttt{coupons}\ cs\ (\lambda\_.0) : [\texttt{Coupons}] \to \texttt{Real}$

   $\mid \forall cs.\mathbf{r} \le |cs| \cdot H(|cs|)$

```
    let coupons cs k =
     letrec collect os k =
M ⎡   if cs ⊆ os
  ⎢   then k os
  ⎣   else $1 + \sum_{c \in cs} \frac{(\texttt{collect}\ (c :: os)\ k)}{|cs|}$
     in collect [ ] k

    let cost cs = coupons cs (λ_.0)
```

Analysing Coupon Collector

1. $\overbrace{cs : [\texttt{Coupons}]}^{\triangleq\Gamma} \mid \top$

   $\vdash \texttt{letrec collect}\ os\ k = M : \ldots \mid \forall os\, k.\ (os \subseteq cs \wedge \forall os'.k\ os' = 0) \Rightarrow \mathbf{r}\ os\ k \le Q(os)$

2. $\Gamma \mid \overbrace{\forall os\, k.\ (os \subseteq cs \wedge \forall os'.k\ os' = 0) \Rightarrow \texttt{collect}\ os\ k \le Q(os); os \subseteq cs; \forall os'.k\ os' = 0}^{\triangleq\Phi}$

   $\vdash M : \texttt{Real} \mid \mathbf{r} \le Q(os)$

   2.1 $\Gamma \mid \Phi; cs \subseteq os \vdash k\ os : \texttt{Real} \mid \mathbf{r} \le 0$

   2.2 $\Gamma \mid \Phi; \neg(cs \subseteq os) \vdash 1 + \sum_{c \in cs} \frac{(\texttt{collect}\ (c :: os)\ k)}{|cs|} : \texttt{Real} \mid \mathbf{r} \le 1 + \sum_{c \in cs} \frac{Q(c :: os)}{|cs|}$

$\Rightarrow$ constraints 2.1 $\Phi; cs \subseteq os \vDash 0 \le Q(os)$ and 2.2 $\Phi; \neg(cs \subseteq os) \vDash 1 + \sum_{c \in cs} \frac{Q(c :: os)}{|cs|} \le Q(os)$

Define $Q(os) \triangleq |cs| \cdot H(|cs \setminus os|)$ where $H(n) = \sum_{i=1}^{n} 1/i$

3. $\vdash \texttt{let cost}\ cs = \texttt{coupons}\ cs\ (\lambda\_.0) : [\texttt{Coupons}] \to \texttt{Real}$

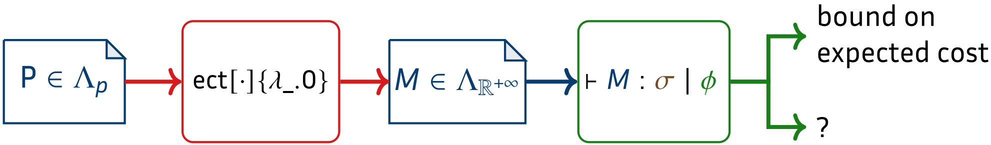
   $\mid \forall cs.\mathbf{r} \le |cs| \cdot H(|cs|)$

$\Rightarrow$ $E[cost(\texttt{coupons 1})] = [\![\texttt{cost 1}]\!] \le |1| \cdot H(|1|)$

```
let coupons cs k =
  letrec collect os k =
      if $cs \subseteq os$
M     then $k\ os$
      else $1 + \sum_{c \in cs} \frac{(\texttt{collect}\ (c :: os)\ k)}{|cs|}$
  in collect [ ] k

let cost cs = coupons cs (λ_.0)
```

Conclusion



$\mathsf{P} \in \Lambda_p$

$\mathsf{ect}[\cdot]\{\lambda\_.0\}$

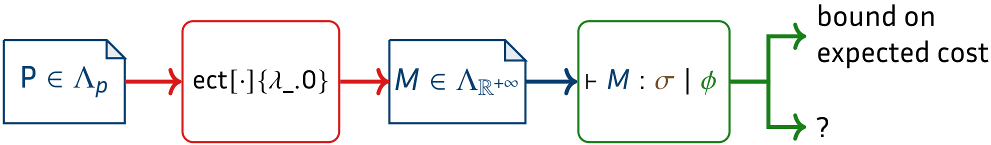
$M \in \Lambda_{\mathbb{R}^{+\infty}}$

$\vdash M : \sigma \mid \phi$

bound on expected cost

?

Conclusion



Remarks

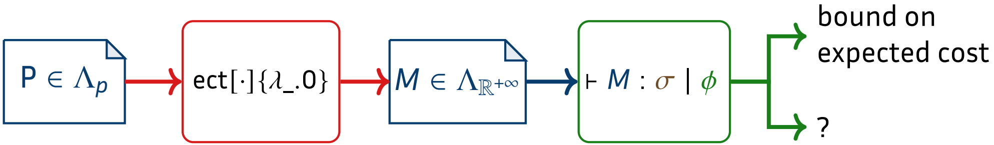
- embeds the *ert*-calculus of Kaminski et al. for imperative programs

$$\mathrm{ert}[\mathtt{C}](\llbracket\kappa\rrbracket)(\sigma) = \llbracket \mathrm{ect}[(\!|\mathtt{C}|\!)\ \sigma]\{\kappa\}\rrbracket$$

- allows reasoning about pre-expectation of $\kappa : \mathtt{B} \longrightarrow \mathtt{Real}$

$$\llbracket \mathrm{ect}[M]\{\kappa\}\rrbracket = E(cost(M)) + E(\llbracket \kappa\ M\rrbracket)$$

Conclusion



$\mathsf{P} \in \Lambda_p$ → $\mathrm{ect}[\cdot]\{\lambda\_.0\}$ → $M \in \Lambda_{\mathbb{R}^{+\infty}}$ → $\vdash M : \sigma \mid \phi$ → bound on expected cost / ?

Remarks

- ⋆ embeds the *ert*-calculus of Kaminski et al. for imperative programs

$$\mathrm{ert}[\mathtt{C}](\llbracket \kappa \rrbracket)(\sigma) = \llbracket \mathrm{ect}[\llparenthesis \mathtt{C} \rrparenthesis\ \sigma]\{\kappa\} \rrbracket$$

- ⋆ allows reasoning about pre-expectation of $\kappa : \mathtt{B} \longrightarrow \mathtt{Real}$

$$\llbracket \mathrm{ect}[M]\{\kappa\} \rrbracket = E(cost(M)) + E(\llbracket \kappa\ M \rrbracket)$$

Open Questions

1. expressiveness?
2. implementation?
3. lower bounds?

Thank you for your attention!