Bounded exponentials

- Girard ‘87
- Wadler ‘91
- Girard, Scedrov & Scott ‘92
- Wraight & Baker-Finch ‘93
- Pitt ‘94
- Hofmann & Scott ‘04
- Schöpp ‘07
- Kamide ‘08
- Dal Lago & Hofmann ‘09
- Dal Lago & Schöpp ‘10
- Reed and Pierce ‘10
- Carraro & al ‘10
- Dal Lago & Gaboardi ‘11
- Dal Lago & Petit ‘12 and ‘13,
- Gaboardi & al ‘13
- Dal Lago & Pellitta ‘13
Bounded exponentials

- Girard ‘87
- Wadler ‘91
- Girard, Scedrov & Scott ‘92
- Wraight & Baker-Finch ‘93
- Pitt ‘94
- Hofmann & Scott ‘04
- Schöpp ‘07
- Kamide ‘08
- Dal Lago & Hofmann ‘09
- Dal Lago & Schöpp ‘10
- Reed and Pierce ‘10
- Carraro & al ‘10
- Dal Lago & Gaboardi ‘11
- Dal Lago & Petit ‘12 and ‘13,
- Gaboardi & al ‘13
- Dal Lago & Pellitta ‘13

- Use types
- Polynomial computations
- Usage analysis
- Logtime computations
- Differential Privacy
- Complexity analysis
Boundedness rough idea

\[ \!_n A = (A \land 1) \bigotimes (A \land 1) \bigotimes \ldots \bigotimes (A \land 1) \quad n\text{-times} \]
Boundedness rough idea

\[ !_n A = (A \land I) \otimes (A \land I) \otimes \ldots \otimes (A \land I) \text{ n-times} \]

but this does not include all the analyses above

\[ !_n A = n \ast A \]
Indexes...
Types and effects systems

This reminds the idea of effect systems:

\[
\vdash M : (\tau, e)
\]
Types and effects systems

This reminds the idea of effect systems:

\[ \vdash M : (\tau, e) \]

- Control flow analysis
- Binding time analysis
- Causality analysis
- Side effects - regions
- File handling
- Capability checking
- Memory access
How far does this correspondence go?

Effect: description of the intensional aspects of the dynamic behaviour [Nielson&Nielson’99]
How far does this correspondence go?

**Effect:** description of the intensional aspects of the dynamic behaviour [Nielson&Nielson’99]

This is exactly what also bounded types can do!
Correspondence more formally?

Monad

effect

Comonad

??
Correspondence more formally?

- monad
- effect
- comonad
- coeffect
What is a coeffect?

Coeffect: description of the intensional aspects of the dynamic behaviour!

effect
how a program influences its environment

coeffect
requirement of a program wrt its environment
Related works: comonadic notions of computations

- Comonadic notions of computations
  [Uustalu&Vene’08]
- Coeffects: unified static analysis of context-dependences
  [Petricek&Orchard&Mycroft’13]
Our work

• a parametric version of bounded types (almost identical to Ghica&Smith and a quantitative version of Petricek&al)

• a formal treatment of primitive coeffects (basic primitives, instrumented semantics)

• a formal soundness result (inspired by quantitative semantics)
Other analysis

• usage analysis
• complexity analysis
• sensitivity analysis
• liveness analysis
• Implicit parameters
• probabilistic analysis
• signal processing
• Control Flow Analysis?
• Non Interference?
• Model Checking?
• ....
Outline

• Motivations
• Bounded types for sensitivity analysis
• Adding primitives?
• Primitives as co-effects
• Quantitative soundness
A type system for k-sensitivity
A type system for k-sensitivity

\[
\begin{align*}
x : !^1 A & \vdash x : A \\
\Gamma & \vdash r : \text{Real} \\
\Gamma, x : !^p A & \vdash e : B \\
\Gamma \vdash \lambda x.e : !^p A \rightarrow \sigma B \\
\Gamma \vdash e_1 : !^p A \rightarrow \sigma B & \quad \Delta \vdash e_2 : A \\
\Gamma + p^*\Delta & \vdash e_1 \ e_2 : B \\
\Delta & \vdash e : A \\
\Delta \leq \Gamma & \\
\Gamma, \Gamma' & \vdash e : A
\end{align*}
\]
A type system for k-sensitivity

\[
\begin{align*}
\Gamma & \vdash x : !^1 A \\
\Gamma & \vdash \lambda x.e : !^p A \rightarrow^o B \\
\Delta & \vdash e_2 : A \\
\Gamma, \Gamma' & \vdash e : A
\end{align*}
\]
A type system for k-sensitivity

ranging over Reals too

\[
\begin{align*}
\Gamma, \ x : !_p A & \vdash e : B \\
\Gamma, \ x : !_p A & \vdash \lambda x. e : !_p A \rightarrow B \\
\Gamma & \vdash e_1 : !_p A \rightarrow B \\
\Delta & \vdash e_2 : A \\
\Gamma + \text{p}^* \Delta & \vdash e_1 \ e_2 : B \\
\Delta & \vdash e : A \\
\Delta \leq \Gamma \\
\Gamma, \Gamma' & \vdash e : A
\end{align*}
\]
A type system for k-sensitivity

\[ \Gamma |- e : \mu a.A \quad \text{Γ |- e : } A[\mu a.A/a] \]
\[ \Gamma |- \text{unfold}_{\mu a.A} e : A[\mu a.A/a] \quad \Gamma |- \text{fold}_{\mu a.A} e : \mu a.A \]
\[ \Gamma |- e : A_i \]
\[ \Gamma |- \text{inj}_i e : A_1 \oplus A_2 \]
\[ \Gamma |- e : A \oplus B \quad \Delta, x: !_1 A |- e_1 : C \quad \Delta, x: !_1 B |- e_2 : C \]
\[ \Gamma + \Delta |- \text{case } e \text{ of } x.e_1 \text{ or } x.e_2 : C \]
Examples

As expected:

\[ y : !1(!1A \rightarrow \odot(!1A \rightarrow \odot B)) \vdash \lambda x. yxx : !2 A \rightarrow \odot B \]

Also:

\[ \vdash \mathrm{fold}_{\mu X. !1 \oplus X} (\mathrm{inr}_{!1 \oplus X} (\mathrm{fold}_{\mu X. !1 \oplus X} (\mathrm{inl}_{!1 \oplus X} (())))) : \mu a. !1 \oplus a \]

Adding \( \infty \) to the language of sensitivities we also have

\[ \vdash Y : !\infty(!\infty(!r A \rightarrow \odot B) \rightarrow \odot (!r A \rightarrow \odot B)) \rightarrow \odot (!r A \rightarrow \odot B) \]
Metric Preservation-Soundness

**Theorem (Metric Preservation)**
Let \( \vdash e : !c A \rightarrow o B \). Then, for \( \vdash e_1 \sim_k e_2 : A \)

\[
e e_1 \mapsto v_1 \quad \text{implies} \quad e e_2 \mapsto v_2
\]

and

\[
\vdash v_1 \sim_{c^*k} v_2 : B
\]
Metric Relation - I

\[ |- v \sim_m v : A \]

\[ |r_1 - r_2| \leq m \]

\[ |- r_1 \sim_m r_2 : R \]

\[ |- v_1 \sim_m v_2 : A_i \]

\[ |- \text{inj}_i v_1 \sim_m \text{inj}_i v_2 : A_1 \oplus A_2 \]

\[ |- v_1 \sim_m v_2 : A[\mu X.A/X] \]

\[ |- \text{fold}_{\mu X.A} v_1 \sim_m \text{fold}_{\mu X.A} v_2 : A \]
Metric Relation - 2

\( \forall \ |- \ v_1 \approx_m v_2 : A, \ |- \ e_1[v_1/x] \approx_n e_2[v_2/x] : B \)
\[ \land \ n \leq c^*m + d \]
\[ \|- \ \lambda x. e_1 \sim_d \lambda x. e_2 : !c A \rightarrow_o B \]

\[
(\forall \ v_1, e_1 \mapsto v_1 \Rightarrow \exists \ v_2, e_2 \mapsto v_2 \ \land \ |- \ v_1 \sim_d v_2 : A) \ \land \\
(\forall \ v_2, e_2 \mapsto v_2 \Rightarrow \exists \ v_1, e_1 \mapsto v_1 \ \Rightarrow \ |- \ v_1 \sim_d v_2 : A) \\
\|- \ e_1 \approx_d e_2 : A
\]
A sketch of the proof

- Almost standard proof by syntactic logical relations (binary realizability).
- An additional overhead in the management of the metric indexes and the sensitivities.
- Step indexing to take care of recursive types.
Metric Preservation-Soundness

**Theorem (Metric Preservation)**

Let \( \vdash e: !c A \rightarrow o B \). Then, for \( \vdash e_1 \approx_k e_2 : A \)

\[ e e_1 \mapsto v_1 \text{ implies } e e_2 \mapsto v_2 \]

and

\( \vdash v_1 \approx_{c^*k} v_2 : B \)
Theorem (Metric Preservation)
Let $\vdash e: !c A \rightarrow o B$. Then, for $\vdash e_1 \equiv_k e_2 : A$

$e \ e_1 \mapsto v_1$ implies $e \ e_2 \mapsto v_2$

and

$\vdash v_1 \equiv_{c^*k} v_2 : B$

Isn’t it trivial?
Simpler properties

**Property (Real functions)**
Let \( \vdash e : !_c \mathbb{R} \rightarrow \mathbb{R} \) with \( c \in \mathbb{R} \). Then, either \( e \) is a constant function or it is the identity.

**Property (Nat functions)**
Let \( \vdash e : !_c \mathbb{N} \rightarrow \mathbb{N} \) with \( c \in \mathbb{R} \). Then, either \( e \) is a constant function or it is the identity.
What is missing?

• Since $\mathbb{R}$ is a primitive data type we need primitives to manage its elements

• E.g. addition, multiplication, scaling, comparison, etc.

• A criteria to see whether a primitive is admissible or not.
Why a criteria?

Suppose that we add the following:

\[ \vdash \text{double} : ! \mathbb{R} \rightarrow o \mathbb{R} \]

Then, the metric preservation fails:
consider \( \vdash 2 \approx 3 : \mathbb{R} \), then:

\[
\begin{align*}
\text{double} & \quad \mapsto \quad 4 \\
\text{but} \quad \vdash 4 & \neq 6 : \mathbb{R}
\end{align*}
\]
Why a criteria?

• For first order functions on real numbers it can be easy to determine the right type:

\[ \begin{align*}
| & - + : !_1 R \rightarrow !_1 R \rightarrow R \\
| & - k^* : !_k R \rightarrow R \\
| & - * : !\infty R \rightarrow !\infty R \rightarrow R
\end{align*} \]

• How about higher order functions or other data types?
Let’s add a type \( \text{Set} A \) for sets of elements of type \( A \).

We extend the metric as follow:

\[
\frac{\#(s_1 \Delta s_2) \leq m}{\vdash s_1 \sim_m s_2 : \text{Set} A}
\]

E.g.: \( \{a,b\} \sim_1 \{a,b,c\} \) and \( \{a,b,d\} \sim_2 \{a,b,c\} \)

Are the following primitive admissible?

filter: \( !! (\infty A \rightarrow \text{Bool}) \rightarrow \text{Set} A \rightarrow \text{Set} A \)

map: \( !! (\infty A \rightarrow B) \rightarrow \text{Set} A \rightarrow \text{Set} B \)
Sets - DB

How about the following?

tfilter: \(!(|(\infty A \rightarrow \text{Bool}) \rightarrow N \rightarrow !|\text{Set } A \rightarrow \text{Set } A)

\[
tfilter(f, n, \{v_1, \ldots, v_m\}) = \{v_i \mid f v_i \mapsto^k \text{tt} \land k \leq n\}
\]

Similarly we can have:

tmap: \(!(|(\infty A \rightarrow B) \rightarrow N \rightarrow B \rightarrow B \rightarrow !|\text{Set } A \rightarrow \text{Set } B

OT: Another interesting example

Interestingly, the following program

\[ \text{limit } r = \frac{1}{2} r + \frac{1}{2} (\text{limit } r) \]

can be typed as:

\[ \text{|- limit : !} \ R \ --o \ R \]
Solution?

- A possible solution is to define a denotational model and require that each primitive lives in the right space.

- Difficult to extend with non standard data types.

- Another easy solution is to require that the primitives that we add satisfy the metric preservation!
Adding primitives

We can add a primitive $\vdash f : \!_c A \rightarrow B$ iff for every $\vdash v_1 \approx^k v_2 : A$,

$$f v_1 \mapsto w_1 \text{ and } f v_2 \mapsto w_2$$

with

$$\vdash w_1 \approx_{c^k} w_2 : B$$

In this way we preserve the metric preservation!
Generalizing this idea?

- This situation reminds what happens in Moggi’s computational lambda calculus.
- There one can define a monad representing different computations.
- In order to have interesting computations, one needs to add some primitive effect.
- By adding effects one needs to show that the semantics is preserved.
Generalizing this idea?

• We introduce a generalized calculus inspired by BLL

• There one can define a indexed comonad representing different resources.

• In order to have interesting computations, one needs to add some primitive coeffect.

• By adding coeffects one needs to show that the semantics is preserved.
Structural Semiring

\((\mathbb{R}, 0, 1, +, *, \leq, \lor, \infty)\)

- \((\mathbb{R}, 0, +)\) is a commutative monoid
- \((\mathbb{R}, 1, *)\) is a monoid
- \(*\) distributes over \(+\)
- \((\mathbb{R}, \leq, \lor)\) is a bounded sup-semilattice
- 0 is absorbing for \(*\)
- 0 is the min and \(\infty\) is the max element
**Generalized type system**

\[
\frac{x : A}{x : A \vdash x : A}
\]

\[
\frac{\Gamma, x : A \vdash e : B}{\Gamma \vdash \lambda x. e : A \rightarrow B}
\]

\[
\frac{\Gamma |\!| e_1 : A \rightarrow B}{\Gamma + \Delta \vdash e_1 \cdot e_2 : B}
\]

\[
\frac{\Gamma, x : [A]_1 \vdash e : B}{\Gamma, x : [A] \vdash e : B}
\]

\[
\frac{\Gamma |\!| e_1 : !_p A}{\Gamma + \Delta, x : [A]_p \vdash e_2 : B}
\]

\[
\frac{\Gamma |\!| e : B}{r^* [\Gamma] \vdash !e : !B}
\]

\[
\frac{\Gamma |\!| e_1 : !_p A, \Delta, x : [A]_p \vdash e_2 : B}{\Gamma + \Delta \vdash \text{let } !x = e_1 \text{ in } e_2 : B}
\]
Generalized type system

\[\Gamma \vdash 0 : \text{Nat}\]

\[\Gamma \vdash \text{s}(e) : \text{Nat}\]

\[\Gamma \vdash e : \text{Nat} \quad \Delta \vdash e_1 : C \quad \Delta, x : \text{Nat} \vdash e_2 : C\]

\[\Gamma + \Delta \vdash \text{case } e \text{ of } 0.e_1 \text{ or } \text{s}(x).e_2 : C\]

\[\Gamma \vdash e : B \quad \Gamma \leq \Gamma'\]

\[\Gamma', \Delta \vdash e : B\]

\[\Gamma, x : [A]_p \vdash e : A \quad 1 + p*q \leq q\]

\[q^*[\Gamma] \vdash \text{Fix } x.e : A\]
Adding coeffects

\[ [\Gamma] |- e : A_{\text{coeff}} \]
\[ r_{\text{coeff}}^*[\Gamma] |- \text{coeff } e : B_{\text{coeff}} \]

- We take coeffects as first class citizens
- Every coeffect \text{coeff} comes with input and output types \( A_{\text{coeff}} \) and \( B_{\text{coeff}} \) and its action \( r_{\text{coeff}} \) on the environment
- It also comes with two semantics condition that we will detail later.
A technical device

\[
\Gamma |- e : B \\
\hline
\Gamma |- \langle e \rangle : B
\]

We use \langle - \rangle as an explicit marker for what we want to observe.
Instrumented operational semantics

\[
\begin{align*}
&v \Downarrow^0 v \\
&v \Downarrow^0 v \\
\end{align*}
\]

\[
\frac{e_1 \Downarrow^p \lambda x.e'}{e''[e_2/x] \Downarrow^q v}
\]

\[
\frac{e_1 \Downarrow^p \lambda x.e'}{e''[e_2/x] \Downarrow^q v}
\]

\[
\frac{e_1 \Downarrow^p \lambda x.e'}{e''[e_2/x] \Downarrow^q v}
\]

\[
\text{let } !x=e_1 \text{ in } e_2 \Downarrow^{p+q} v
\]

\[
\frac{e \Downarrow^p v}{\text{coeff } e \Downarrow^{r^*p} \varphi(v)}
\]

\[
\frac{e \Downarrow^p v}{\langle e \rangle \Downarrow^{p+l} v}
\]
Generalizing quantitative realizability

weighted closures \(((e, \rho), r)\) and environment \((C, r)\)

quantitative pole \(\perp r\)

[set of configurations \(C \downarrow r'\) with \(r' \leq r\)]

**Saturation**: if \(C'\) in \(\perp_p\) and \(C \rightarrow C'\) then \(C\) in \(\perp_p\)

**Monotonicity**: if \(p \leq q\) then \(\perp_p \subseteq \perp_q\)
Interpretation

\[
\text{Nat} := \{ ((n, []), (0, \infty)) \mid n \in \mathbb{N} \} \\
X \rightarrow Y := \{ ((\lambda x. e, \rho), \alpha) \mid \forall (c', \beta) \in X, ((e, \rho \cdot [x/c']), \alpha + \beta) \in Y^{\bot\bot} \} \\
r * X := \{ (c, (r * p, m)) \mid (c, (p, m)) \in X \} \\
!_r X := \{ ((!e, \rho), \alpha) \mid ((e, \rho), \alpha) \in r * X \}
\]

\[
||A|| = \text{Nat}^{\bot\bot} \\
||A \rightarrow B|| = (||A|| \rightarrow_\circ ||B||)^{\bot\bot} \\
||!_r A|| = ((!_r ||A||))^{\bot\bot}
\]
Interpreting coeffects

\[
\frac{[\Gamma] \vdash e : A_{\text{coeff}}}{r_{\text{coeff}}^{*}[\Gamma] \vdash \text{coeff } e : B_{\text{coeff}}}
\]

- We have an additional requirement on coeffects for our soundness:

\((C[\cdot], p) \in \parallel B_{\text{coeff}} \parallel \vdash \) implies \((C[\text{coeff}[\cdot]], p) \in \parallel A_{\text{coeff}} \parallel \vdash\)
Adding weights

\[
\Gamma,\ x: A \vdash_{0} x : A
\]

\[
\Gamma,\ x: A \vdash_{p} e : B \\
\Delta \vdash_{q} e_{2} : A
\]

\[
\Gamma + \Delta \vdash_{p+q} e_{1} e_{2} : B
\]

\[
\Gamma,\ x: A \vdash_{p} \lambda x. e : A \rightarrow B
\]

\[
\Gamma \vdash_{p} e_{1} : A \rightarrow B
\]

\[
\Gamma + \Delta \vdash_{p+q} \text{let } !x = e_{1} \text{ in } e_{2} : B
\]

\[
\Gamma,\ x: A \vdash_{p} e : B
\]

\[
\Gamma \vdash_{p} e : B
\]

\[
\Gamma, x : [A]_{1} \vdash_{p} e : B
\]

\[
r^{*} \Gamma \vdash_{p} !e : !B
\]
Adding weights

\[ \Gamma |-p e : \text{Nat} \]

\[ \Gamma |-p s(e) : \text{Nat} \]

\[ \Gamma |-p e : \text{Nat} \quad \Delta |-q e_1 : \text{C} \quad \Delta, x: \text{Nat} |-r e_2 : \text{C} \]

\[ \Gamma + \Delta |-p+(q \lor r) \text{ case } e \text{ of } 0.e_1 \text{ or } s(x).e_2 : \text{C} \]

\[ \Gamma |-p e : B \quad \Gamma \leq \Gamma' \]

\[ \Gamma', \Delta |-p e : B \]

\[ [\Gamma], x: [A]_r |-p e : A \quad l+r^*q \leq q \]

\[ q^*[\Gamma] |-p \text{ Fix } x.e : A \]
Adding weights

\[
\frac{[\Gamma] \vdash_p e : A_{\text{coeff}}}{r_{\text{coeff}}\ast [\Gamma] \vdash_r^{p} \text{coeff } e : B_{\text{coeff}}}
\]

\[
[\Gamma] \vdash_p e : B
\]

\[
[\Gamma] \vdash_p^{p+1} \langle e \rangle : B
\]
Generalized soundness

**Theorem (soundness)**
Let $|-_{w} e : A$. Then, $((e, []) , w) \models A$.

**Corollary**
Let $|-_{w} e : A$. If $e \Downarrow^{P} v$ then $p \leq w$. 
Example 1 - complexity analysis

• A limited complexity analysis similar to the one in ldPCF [Dal Lago&Gaboardi’11].

• No coeffects. Semiring: \((\text{Nat}, 0, 1, +, *, \leq, \lor, \infty)\)

• The idea is to add an observation around every variable occurrence.

• The weight that one obtains is a bound on the number of variable steps in the machine.

**Lemma**

Let \( |- \_w e : A. \) \( e \xrightarrow{n} v \) then \( n \leq w^*|e| \).
Example II - signal processing

• Arctic semiring: \((\mathbb{N} \cup \{+\infty, -\infty\}, -\infty, 0, \text{max}, +, \leq, \lor, \infty)\)

• We can extend our system with the following rules

\[ \Gamma \vdash_0 s : \text{Sig} \quad \Gamma \vdash_{\mathbb{L}} p \ e : \text{Sig} \]

\[ l + \Gamma \vdash_{\mathbb{L}+p} \text{next}(e) : \text{Sig} \]

• Elements as \(s_1, s_2, \ldots\) represent streams as clocked signals

• next is a primitive to increase the clock of signals
Example II - signal processing

• Notice that interestingly here we use coeffects in a quantitative way.

\[
\Gamma |- p \, e : \text{Sig} \\
\downarrow \\
\Gamma +\Gamma |- \uparrow p \, \text{next}(e) : \text{Sig}
\]

• The soundness gives us a bound on the number of next on the argument stream.

**Lemma**
Let \(x : [\text{Sig}]_r \ |- e : A\). Then, the evaluation of \(e[s/x]\) uses at most the \(r\) first elements of \(s\).
Example III - probabilistic failure

- Probability semiring: \((\mathbb{R}_{\geq 0} \cup \{\infty\}, 0, 1, +, *, \leq, \lor, \infty)\)

- We can extend our system with the following rule

\[
\Gamma |-_{p} e : N \\
\lambda * \Gamma |-_{p} \text{coflip}(e) : N
\]

- \text{coflip}(v) \mapsto v \text{ with probability } \lambda

- \text{coflip}(v) \text{ fails with probability } (1 - \lambda)
Example III - probabilistic failure

• If we define \( \text{var}(e) \) as the number of steps (variable steps) before a failure occur or before reaching \( nf \).

• For a typable term, the soundness gives us a bound on \( \text{var}(e) \).

Lemma

Let \( |-w \ e : A \). Then, \( \text{var}(e) \leq w \).
Quantitative Type Inference

Traditional Type Inference
- Constraint generation
- Constraint resolution

Odersky et Al ’01
Pottier Remy’05
Quantitative Type Inference

Traditional Type Inference
- Constraint generation
- Constraint resolution

Quantitative Type Inference
- Constraint generation
  - Symbolic
  - Quantitative
- Symbolic constraint resolution
- Quantitative constraint resolution
Quantitative Type Inference

Traditional Type Inference

- Constraint generation
- Constraint resolution

Quantitative Type Inference

- Constraint generation
  - symbolic
  - quantitative
- symbolic constraint resolution
- quantitative constraint resolution
Constraint Generation Algorithm

\[ \Gamma^* |- e \Rightarrow \sigma |- \Delta | C \]

- fresh context
- decorated context
- constraints
Costraint Generation
Algorithm

\[
\Gamma \vdash x \Rightarrow \Gamma(x) + x : \Gamma(x) \mid \emptyset
\]

\[
x : \alpha, \Gamma \vdash e \Rightarrow \beta + \Gamma_o \mid C \quad (x : \alpha \in \Gamma_o) \quad \alpha \text{ fresh}
\]

\[
\Gamma \vdash \lambda x.e \Rightarrow \alpha \rightarrow \beta \vdash x : \Gamma(x) \mid C
\]

\[
\Gamma \vdash e \Rightarrow \alpha \rightarrow \beta \vdash \Gamma_f \mid C \quad \Gamma \vdash e' \Rightarrow \alpha' \vdash \Gamma_a \mid C' \quad \Delta \equiv \text{sum.up} (\Gamma_f, \Gamma_a)
\]

\[\Gamma \vdash e e' \Rightarrow \beta \vdash \Delta \mid C \land C' \land \alpha' \leq \alpha\]

\[
\Gamma \vdash e \Rightarrow \alpha \vdash \Gamma_e \mid C \quad [\Delta] \equiv \text{up} (\Gamma_e) \quad r \text{ fresh}
\]

\[
\Gamma \vdash !e \Rightarrow !r \alpha \vdash r \star [\Delta] \mid C
\]

\[
\Gamma \vdash e \Rightarrow !r \alpha \vdash \Gamma_e \mid C
\]

\[
x : \alpha, \Gamma \vdash e' \Rightarrow \beta \vdash \Gamma_{e'} \mid C' \quad \Delta \equiv \text{up} (x : [\alpha]_{r'} \downarrow \Gamma_{e'}) \quad \Sigma \equiv \text{sum.up} (\Gamma_e, \Delta)
\]

\[
\Gamma \vdash \text{let } !x = e \text{ in } e' \Rightarrow \beta \vdash \Sigma \mid r \leq r' \land C \land C'
\]

\[
x : \alpha, \Gamma \vdash e \Rightarrow \alpha \vdash \Gamma_f \mid C \quad \Delta \equiv \text{up} (x : [\alpha]_p \downarrow \Gamma_f) \quad [\Sigma] \equiv \text{up}(\Delta) \quad q \text{ fresh}
\]

\[
\Gamma \vdash \text{fix } x.e \Rightarrow \alpha \vdash q \star [\Sigma] \mid (1 + p \star q) \leq q \land C
\]
CBN vs CBV boxes

\[
\frac{[\Gamma] |- e : A_{coeff}}{r_{coeff}^* [\Gamma] |- coeff e : B_{coeff}}
\]

• The rule for coeffects corresponds to a kind of call-by-value promotion (!-boxes).

• This changes the natural surface strategy usually associated with call-by-name or call-by-value translations.

• Different ways to combine/present them?
Data dependent analyses

- The analysis we presented so far, cannot be expressed in terms of the value of the data input. (e.g. clock signal value, a given probability value, etc...)
- Many analyses require such kind of dependency.
- Some dependency is already provided by the semantics but the type system is not yet good enough to express it.
Todo: linear dependent types

• A first step is to add lightweight linear dependent types as the one in [Gaboardi&al’13]

• These corresponds to a combination of sized types, size-dependent pattern matching, universal and existential quantifications on resources and size information

• The long term goal is to have also a more precise type system - generalized relative completeness?
A general coeffect calculus

- The comonad we consider is parametrized on the elements of the structural semiring.
- This permits us to be quantitative.
- In general, one would like also to be able to define more general comonads, e.g.

- How to generalize it?