Context Constrained Computation

via a linear-like lambda calculus

Bob Atkey[1] James Wood[1]

[1]University of Strathclyde

SPLS, October 2018

Motivation

- Constrain how variables are used

Motivation

- Constrain how variables are used
- Derive more free theorems about constrained programs

Motivation

- Constrain how variables are used
- Derive more free theorems about constrained programs
- Generalise the “how many” of linear typing

Motivation

- Constrain how variables are used
- Derive more free theorems about constrained programs
- Generalise the "how many" of linear typing
  - At what security level? – information flow

Motivation

- Constrain how variables are used
- Derive more free theorems about constrained programs
- Generalise the "how many" of linear typing
  - At what security level? – information flow
  - How far away? – sensitivity analysis

Motivation

- Constrain how variables are used
- Derive more free theorems about constrained programs
- Generalise the "how many" of linear typing
  - At what security level? – information flow
  - How far away? – sensitivity analysis
  - In which direction? – monotonicity

Motivation

- Constrain how variables are used
- Derive more free theorems about constrained programs
- Generalise the “how many” of linear typing
  - At what security level? – information flow
  - How far away? – sensitivity analysis
  - In which direction? – monotonicity
- Formalised in Agda – the free theorems of the object language are available to Agda programs

Permutations

- Have:

Permutations

- Have:
  - A linearly used type KEY

Permutations

- Have:
  - A linearly used type KEY
  - An operation
    $\mathit{compareAndSwap} : \mathrm{KEY} \otimes \mathrm{KEY} \multimap \mathrm{KEY} \otimes \mathrm{KEY}$

Permutations

- Have:
  - A linearly used type KEY
  - An operation
    $\mathit{compareAndSwap} : \mathrm{KEY} \otimes \mathrm{KEY} \multimap \mathrm{KEY} \otimes \mathrm{KEY}$
- Programmer writes $\mathit{sort} : \mathrm{List}\ \mathrm{KEY} \multimap \mathrm{List}\ \mathrm{KEY}$

Permutations

- Have:
  - A linearly used type KEY
  - An operation
    $\mathit{compareAndSwap} : \mathrm{KEY} \otimes \mathrm{KEY} \multimap \mathrm{KEY} \otimes \mathrm{KEY}$
- Programmer writes *sort* $: \mathrm{List}\ \mathrm{KEY} \multimap \mathrm{List}\ \mathrm{KEY}$

Free theorem

sort is a permutation

Monotonicity

- Have:

Monotonicity

- Have:
  - A type of integers $\mathbb{Z}$

Monotonicity

- Have:
  - A type of integers $\mathbb{Z}$
  - $+ : \mathbb{Z} \otimes \mathbb{Z} \multimap \mathbb{Z}$

Monotonicity

- Have:
  - A type of integers $\mathbb{Z}$
  - $+ : \mathbb{Z} \otimes \mathbb{Z} \multimap \mathbb{Z}$
  - *neg* $: !_{\downarrow}\mathbb{Z} \multimap \mathbb{Z}$

Monotonicity

- Have:
  - A type of integers $\mathbb{Z}$
  - $+ : \mathbb{Z} \otimes \mathbb{Z} \multimap \mathbb{Z}$
  - *neg* : $!_{\downarrow}\mathbb{Z} \multimap \mathbb{Z}$
  - *zero* : $!_{0}\mathbb{Z} \multimap \mathbb{Z}$

Monotonicity

- Have:
  - A type of integers $\mathbb{Z}$
  - $+ : \mathbb{Z} \otimes \mathbb{Z} \multimap \mathbb{Z}$
  - $neg : !_{\downarrow}\mathbb{Z} \multimap \mathbb{Z}$
  - $zero : !_0\mathbb{Z} \multimap \mathbb{Z}$
  - $\ldots$
- Programmer writes a function $f : \mathbb{Z} \multimap \mathbb{Z}$

Monotonicity

- Have:
  - A type of integers $\mathbb{Z}$
  - $+ : \mathbb{Z} \otimes \mathbb{Z} \multimap \mathbb{Z}$
  - $\mathit{neg} : !_{\downarrow}\mathbb{Z} \multimap \mathbb{Z}$
  - $\mathit{zero} : !_{0}\mathbb{Z} \multimap \mathbb{Z}$
  - $\ldots$
- Programmer writes a function $f : \mathbb{Z} \multimap \mathbb{Z}$

Free theorem

f is monotonic

Fundamental lemma

- Relational parametricity argument

Fundamental lemma

- Relational parametricity argument
- $[\![t]\!] : [\![\Gamma]\!] \to [\![S]\!]$

Fundamental lemma

- Relational parametricity argument
- $[\![t]\!] : [\![\Gamma]\!] \to [\![S]\!]$
- $\forall w.\ \forall(\gamma, \gamma') \in [\![\Gamma]\!]^R\ w.\ ([\![t]\!]\ \gamma, [\![t]\!]\ \gamma') \in [\![S]\!]^R\ w$

Fundamental lemma

- Relational parametricity argument
- $[\![t]\!] : [\![\Gamma]\!] \to [\![S]\!]$
- $\forall w.\ \forall(\gamma, \gamma') \in [\![\Gamma]\!]^R\ w.\ ([\![t]\!]\ \gamma, [\![t]\!]\ \gamma') \in [\![S]\!]^R\ w$

Fundamental lemma

- Relational parametricity argument
- $[\![t]\!] : [\![\Gamma]\!] \to [\![S]\!]$
- $\forall w.\ \forall(\gamma, \gamma') \in [\![\Gamma]\!]^R\ w.\ ([\![t]\!]\ \gamma, [\![t]\!]\ \gamma') \in [\![S]\!]^R\ w$

Fundamental lemma

- Relational parametricity argument
- $[\![t]\!] : [\![\Gamma]\!] \to [\![S]\!]$
- $\forall w.\ \forall(\gamma, \gamma') \in [\![\Gamma]\!]^R\ w.\ ([\![t]\!]\ \gamma, [\![t]\!]\ \gamma') \in [\![S]\!]^R\ w$

Metasyntax

- Partially ordered semiring: $(R, \leq, 0, +, 1, \cdot)$, general elements ρ, π

Metasyntax

- Partially ordered semiring: $(R, \leq, 0, +, 1, \cdot)$, general elements ρ, π
- Contexts
  - Scopes m, n (natural numbers)
  - typing contexts $\Gamma = x : S, \ldots, y : T$
  - resourcing contexts $\Delta = x^{\pi}, \ldots, y^{\rho}$

Metasyntax

- Partially ordered semiring: $(R, \leq, 0, +, 1, \cdot)$, general elements ρ, π
- Contexts
  - Scopes m, n (natural numbers)
  - typing contexts $\Gamma = x : S, \ldots, y : T$
  - resourcing contexts $\Delta = x^{\pi}, \ldots, y^{\rho}$
- Bidirectional layered typing
  - Well scoped terms $e : \mathrm{Term}\ n\ \mathrm{syn}$ and $s : \mathrm{Term}\ n\ \mathrm{chk}$
  - $t : \mathrm{Term}\ n\ d$ ranges over e and s.

Metasyntax

- Partially ordered semiring: $(R, \leq, 0, +, 1, \cdot)$, general elements ρ, π
- Contexts
  - Scopes m, n (natural numbers)
  - typing contexts $\Gamma = x : S, \ldots, y : T$
  - resourcing contexts $\Delta = x^{\pi}, \ldots, y^{\rho}$
- Bidirectional layered typing
  - Well scoped terms $e : \mathrm{Term}\ n\ \mathrm{syn}$ and $s : \mathrm{Term}\ n\ \mathrm{chk}$
  - $t : \mathrm{Term}\ n\ d$ ranges over e and s.
  - Synthesis: $et : (\Gamma \vdash e \in S)$

Metasyntax

- Partially ordered semiring: $(R, \leq, 0, +, 1, \cdot)$, general elements ρ, π
- Contexts
  - Scopes m, n (natural numbers)
  - typing contexts $\Gamma = x : S, \ldots, y : T$
  - resourcing contexts $\Delta = x^{\pi}, \ldots, y^{\rho}$
- Bidirectional layered typing
  - Well scoped terms $e : \mathrm{Term}\ n\ \mathrm{syn}$ and $s : \mathrm{Term}\ n\ \mathrm{chk}$
  - $t : \mathrm{Term}\ n\ d$ ranges over e and s.
  - Synthesis: $et : (\Gamma \vdash e \in S)$
  - Checking: $st : (\Gamma \vdash S \ni s)$

Metasyntax

- Partially ordered semiring: $(R, \leq, 0, +, 1, \cdot)$, general elements ρ, π
- Contexts
  - Scopes m, n (natural numbers)
  - typing contexts $\Gamma = x : S, \ldots, y : T$
  - resourcing contexts $\Delta = x^\pi, \ldots, y^\rho$
- Bidirectional layered typing
  - Well scoped terms $e : \mathrm{Term}\ n\ \mathrm{syn}$ and $s : \mathrm{Term}\ n\ \mathrm{chk}$
  - $t : \mathrm{Term}\ n\ d$ ranges over e and s.
  - Synthesis: $et : (\Gamma \vdash e \in S)$
  - Checking: $st : (\Gamma \vdash S \ni s)$
  - Either: $tt : (\Gamma \vdash t : S)$

Metasyntax

- Partially ordered semiring: $(R, \leq, 0, +, 1, \cdot)$, general elements ρ, π
- Contexts
  - Scopes m, n (natural numbers)
  - typing contexts $\Gamma = x : S, \ldots, y : T$
  - resourcing contexts $\Delta = x^{\pi}, \ldots, y^{\rho}$
- Bidirectional layered typing
  - Well scoped terms $e : \mathrm{Term}\ n\ \mathrm{syn}$ and $s : \mathrm{Term}\ n\ \mathrm{chk}$
  - $t : \mathrm{Term}\ n\ d$ ranges over e and s.
  - Synthesis: $et : (\Gamma \vdash e \in S)$
  - Checking: $st : (\Gamma \vdash S \ni s)$
  - Either: $tt : (\Gamma \vdash t : S)$
  - Resourcing: $tr : (\Delta \vdash tt)$

Metasyntax

- Partially ordered semiring: $(R, \leq, 0, +, 1, \cdot)$, general elements ρ, π
- Contexts
  - Scopes m, n (natural numbers)
  - typing contexts $\Gamma = x : S, \ldots, y : T$
  - resourcing contexts $\Delta = x^{\pi}, \ldots, y^{\rho}$
- Bidirectional layered typing
  - Well scoped terms $e : \mathrm{Term}\ n\ \mathrm{syn}$ and $s : \mathrm{Term}\ n\ \mathrm{chk}$
  - $t : \mathrm{Term}\ n\ d$ ranges over e and s.
  - Synthesis: $et : (\Gamma \vdash e \in S)$
  - Checking: $st : (\Gamma \vdash S \ni s)$
  - Either: $tt : (\Gamma \vdash t : S)$
  - Resourcing: $tr : (\Delta \vdash tt)$
  - Abbreviations $\Delta^{\Gamma} \vdash e \in S$, $\Delta^{\Gamma} \vdash S \ni s$, etc.

Products

With	Tensor
$A \mathbin{\&} B$	$A \otimes B$

Products

With	Tensor
$A \mathbin{\&} B$	$A \otimes B$

$$swap_{\&} : A \mathbin{\&} B \multimap B \mathbin{\&} A$$

$$swap_{\otimes} : A \otimes B \multimap B \otimes A$$

Products

With	Tensor
$A \mathbin{\&} B$	$A \otimes B$

$$
\begin{aligned}
\mathit{swap}_{\&} &: A \mathbin{\&} B \multimap B \mathbin{\&} A \\
\mathit{swap}_{\otimes} &: A \otimes B \multimap B \otimes A \\
\mathit{choose} &: A \mathbin{\&} A \multimap \mathrm{Bool} \multimap A
\end{aligned}
$$

Products

With	Tensor
$A \mathbin{\&} B$	$A \otimes B$

$$
\begin{aligned}
\mathit{swap}_{\&} &: A \mathbin{\&} B \multimap B \mathbin{\&} A \\
\mathit{swap}_{\otimes} &: A \otimes B \multimap B \otimes A \\
\mathit{choose} &: A \mathbin{\&} A \multimap \mathrm{Bool} \multimap A \\
\mathit{curry} &: (A \multimap B \multimap C) \multimap (A \otimes B \multimap C)
\end{aligned}
$$

With product

Negative type

With product

Negative type

Introduction

$$\frac{\Delta^{\Gamma} \vdash S_0 \ni s_0 \qquad \Delta^{\Gamma} \vdash S_1 \ni s_1}{\Delta^{\Gamma} \vdash S_0 \mathbin{\&} S_1 \ni (s_0, s_1)_{\&}}$$

With product

Negative type

Introduction

$$\frac{\Delta^{\Gamma} \vdash S_0 \ni s_0 \qquad \Delta^{\Gamma} \vdash S_1 \ni s_1}{\Delta^{\Gamma} \vdash S_0 \mathbin{\&} S_1 \ni (s_0, s_1)_{\&}}$$

With product

Negative type

Introduction

$$\frac{\Delta^{\Gamma} \vdash S_0 \ni s_0 \qquad \Delta^{\Gamma} \vdash S_1 \ni s_1}{\Delta^{\Gamma} \vdash S_0 \,\&\, S_1 \ni (s_0, s_1)_{\&}}$$

Elimination

$$\frac{\Delta^{\Gamma} \vdash e \in S_0 \,\&\, S_1 \qquad i \in \{0, 1\}}{\Delta^{\Gamma} \vdash \mathsf{proj}_i\, e \in S_i}$$

With product

Negative type

Introduction

$$\frac{\Delta^{\Gamma} \vdash S_0 \ni s_0 \qquad \Delta^{\Gamma} \vdash S_1 \ni s_1}{\Delta^{\Gamma} \vdash S_0 \,\&\, S_1 \ni (s_0, s_1)_{\&}}$$

Elimination

$$\frac{\Delta^{\Gamma} \vdash e \in S_0 \,\&\, S_1 \qquad i \in \{0, 1\}}{\Delta^{\Gamma} \vdash \mathsf{proj}_i\, e \in S_i}$$

Tensor product

Positive type

Tensor product

Positive type

Introduction

$$\frac{{\Delta_0}^{\Gamma} \vdash S_0 \ni s_0 \qquad {\Delta_1}^{\Gamma} \vdash S_1 \ni s_1 \qquad \Delta \leq \Delta_0 + \Delta_1}{\Delta^{\Gamma} \vdash S_0 \otimes S_1 \ni (s_0, s_1)_{\otimes}}$$

Tensor product

Positive type

Introduction

$$\frac{{\color{red}\Delta_0}^{\Gamma} \vdash S_0 \ni s_0 \qquad \Delta_1{}^{\Gamma} \vdash S_1 \ni s_1 \qquad \Delta \leq \Delta_0 + \Delta_1}{\Delta^{\Gamma} \vdash S_0 \otimes S_1 \ni (s_0, s_1)_{\otimes}}$$

Tensor product

Positive type

Introduction

$$\frac{{\Delta_0}^{\ulcorner} \vdash S_0 \ni s_0 \qquad {\color{red}\Delta_1}^{\color{red}\ulcorner} \vdash S_1 \ni s_1 \qquad \Delta \leq \Delta_0 + \Delta_1}{\Delta^{\ulcorner} \vdash S_0 \otimes S_1 \ni (s_0, s_1)_{\otimes}}$$

Tensor product

Positive type

Introduction

$$\frac{{\Delta_0}^{\Gamma} \vdash S_0 \ni s_0 \qquad {\Delta_1}^{\Gamma} \vdash S_1 \ni s_1 \qquad \Delta \leq \Delta_0 + \Delta_1}{\Delta^{\Gamma} \vdash S_0 \otimes S_1 \ni (s_0, s_1)_{\otimes}}$$

Tensor product

Positive type

Introduction

$$\frac{{\Delta_0}^{\Gamma} \vdash S_0 \ni s_0 \qquad {\Delta_1}^{\Gamma} \vdash S_1 \ni s_1 \qquad \Delta \leq \Delta_0 + \Delta_1}{\Delta^{\Gamma} \vdash S_0 \otimes S_1 \ni (s_0, s_1)_{\otimes}}$$

Elimination

$$\frac{{\Delta_e}^{\Gamma} \vdash e \in S_0 \otimes S_1 \qquad {\Delta_s}^{\Gamma}, x \overset{1}{:} S_0, y \overset{1}{:} S_1 \vdash s \in T \qquad \Delta \leq \Delta_e + \Delta_s}{\Delta^{\Gamma} \vdash \mathrm{let}\ (x, y)_{\otimes} = e\ \mathrm{in}\ s : T \in T}$$

Tensor product

Positive type

Introduction

$$\frac{{\Delta_0}^{\Gamma} \vdash S_0 \ni s_0 \qquad {\Delta_1}^{\Gamma} \vdash S_1 \ni s_1 \qquad \Delta \leq \Delta_0 + \Delta_1}{\Delta^{\Gamma} \vdash S_0 \otimes S_1 \ni (s_0, s_1)_{\otimes}}$$

Elimination

$$\frac{{\color{red}{\Delta_e}^{\Gamma}} \vdash e \in S_0 \otimes S_1 \qquad {\Delta_s}^{\Gamma}, x \overset{1}{:} S_0, y \overset{1}{:} S_1 \vdash s \in T \qquad \Delta \leq \Delta_e + \Delta_s}{\Delta^{\Gamma} \vdash \mathrm{let}\ (x, y)_{\otimes} = e\ \mathrm{in}\ s : T \in T}$$

Tensor product

Positive type

Introduction

$$\frac{{\Delta_0}^{\Gamma} \vdash S_0 \ni s_0 \qquad {\Delta_1}^{\Gamma} \vdash S_1 \ni s_1 \qquad \Delta \leq \Delta_0 + \Delta_1}{\Delta^{\Gamma} \vdash S_0 \otimes S_1 \ni (s_0, s_1)_{\otimes}}$$

Elimination

$$\frac{{\Delta_e}^{\Gamma} \vdash e \in S_0 \otimes S_1 \qquad {\color{red}{\Delta_s}^{\Gamma}}, x \overset{1}{:} S_0, y \overset{1}{:} S_1 \vdash s \in T \qquad \Delta \leq \Delta_e + \Delta_s}{\Delta^{\Gamma} \vdash \mathrm{let}\ (x, y)_{\otimes} = e \ \mathrm{in}\ s : T \in T}$$

Tensor product

Positive type

Introduction

$$\frac{{\Delta_0}^{\Gamma} \vdash S_0 \ni s_0 \qquad {\Delta_1}^{\Gamma} \vdash S_1 \ni s_1 \qquad \Delta \le \Delta_0 + \Delta_1}{\Delta^{\Gamma} \vdash S_0 \otimes S_1 \ni (s_0, s_1)_{\otimes}}$$

Elimination

$$\frac{{\Delta_e}^{\Gamma} \vdash e \in S_0 \otimes S_1 \qquad {\Delta_s}^{\Gamma}, x \overset{1}{:} S_0, y \overset{1}{:} S_1 \vdash s \in T \qquad \Delta \le \Delta_e + \Delta_s}{\Delta^{\Gamma} \vdash \mathrm{let}\ (x, y)_{\otimes} = e\ \mathrm{in}\ s : T \in T}$$

Tensor product

Positive type

Introduction

$$\frac{{\Delta_0}^{\Gamma} \vdash S_0 \ni s_0 \qquad {\Delta_1}^{\Gamma} \vdash S_1 \ni s_1 \qquad \Delta \leq \Delta_0 + \Delta_1}{\Delta^{\Gamma} \vdash S_0 \otimes S_1 \ni (s_0, s_1)_{\otimes}}$$

Elimination

$$\frac{{\Delta_e}^{\Gamma} \vdash e \in S_0 \otimes S_1 \qquad {\Delta_s}^{\Gamma}, x \overset{1}{:} S_0, y \overset{1}{:} S_1 \vdash s \in T \qquad \Delta \leq \Delta_e + \Delta_s}{\Delta^{\Gamma} \vdash \mathrm{let}\ (x, y)_{\otimes} = e\ \mathrm{in}\ s : T \in T}$$

Bang

Introduction

$$\frac{{\Delta_s}^{\Gamma} \vdash S \ni s \qquad \Delta \leq \rho \cdot \Delta_s}{\Delta^{\Gamma} \vdash !_\rho S \ni \mathsf{bang}\, s}$$

Bang

Introduction

$$\frac{\Delta_s{}^{\Gamma} \vdash S \ni s \qquad {\color{red}\Delta \leq \rho \cdot \Delta_s}}{\Delta^{\Gamma} \vdash\ !_{\rho} S \ni \mathsf{bang}\, s}$$

Bang

Introduction

$$\frac{\Delta_s{}^{\ulcorner} \vdash S \ni s \qquad \Delta \leq \rho \cdot \Delta_s}{\Delta^{\ulcorner} \vdash {!_\rho} S \ni \mathsf{bang}\, s}$$

Bang

Introduction

$$\frac{{\Delta_s}^{\Gamma} \vdash S \ni s \qquad \Delta \leq \rho \cdot \Delta_s}{\Delta^{\Gamma} \vdash !_{\rho} S \ni \mathsf{bang}\, s}$$

Elimination

$$\frac{{\Delta_e}^{\Gamma} \vdash e \in !_{\rho} S \qquad {\Delta_s}^{\Gamma}, x \overset{\rho}{:} S \vdash T \ni s \qquad \Delta \leq \Delta_e + \Delta_s}{\Delta^{\Gamma} \vdash \mathrm{let}\ \mathsf{bang}\, x = e \ \mathrm{in}\ s : T \in T}$$

Bang

Introduction

$$\frac{{\Delta_s}^{\Gamma} \vdash S \ni s \qquad \Delta \leq \rho \cdot \Delta_s}{\Delta^{\Gamma} \vdash !_{\rho} S \ni \mathsf{bang}\, s}$$

Elimination

$$\frac{{\Delta_e}^{\Gamma} \vdash e \in !_{\rho} S \qquad {\Delta_s}^{\Gamma}, x \overset{\rho}{:} S \vdash T \ni s \qquad \Delta \leq \Delta_e + \Delta_s}{\Delta^{\Gamma} \vdash \mathrm{let}\ \mathsf{bang}\, x = e \ \mathrm{in}\ s : T \in T}$$

Bang

Introduction

$$\frac{{\Delta_s}^{\Gamma} \vdash S \ni s \qquad \Delta \leq \rho \cdot \Delta_s}{\Delta^{\Gamma} \vdash !_{\rho} S \ni \mathsf{bang}\, s}$$

Elimination

$$\frac{{\Delta_e}^{\Gamma} \vdash e \in !_{\rho} S \qquad {\Delta_s}^{\Gamma}, x \overset{\rho}{:} S \vdash T \ni s \qquad \Delta \leq \Delta_e + \Delta_s}{\Delta^{\Gamma} \vdash \mathrm{let}\ \ \mathsf{bang}\, x = e \ \mathrm{in}\ s : T \in T}$$

Graded comonad

Bang

Introduction

$$\frac{{\Delta_s}^{\Gamma} \vdash S \ni s \qquad \Delta \leq \rho \cdot \Delta_s}{\Delta^{\Gamma} \vdash !_\rho S \ni \mathsf{bang}\, s}$$

Elimination

$$\frac{{\Delta_e}^{\Gamma} \vdash e \in !_\rho S \qquad {\Delta_s}^{\Gamma}, x \overset{\rho}{:} S \vdash T \ni s \qquad \Delta \leq \Delta_e + \Delta_s}{\Delta^{\Gamma} \vdash \mathrm{let}\ \ \mathsf{bang}\, x = e\ \mathrm{in}\ s : T \in T}$$

Graded comonad

$$extract : !_1 A \to A$$

$$extract = \lambda ba.\ \underline{\mathrm{let}\ \ \mathsf{bang}\, a = ba\ \mathrm{in}\ a : A}$$

Bang

Introduction

$$\frac{{\Delta_s}^{\Gamma} \vdash S \ni s \qquad \Delta \leq \rho \cdot \Delta_s}{\Delta^{\Gamma} \vdash !_\rho S \ni \mathsf{bang}\, s}$$

Elimination

$$\frac{{\Delta_e}^{\Gamma} \vdash e \in !_\rho S \qquad {\Delta_s}^{\Gamma}, x \overset{\rho}{:} S \vdash T \ni s \qquad \Delta \leq \Delta_e + \Delta_s}{\Delta^{\Gamma} \vdash \mathrm{let}\ \ \mathsf{bang}\, x = e \ \mathrm{in}\ s : T \in T}$$

Graded comonad

$$extract : !_1 A \to A$$

$$extract = \lambda ba.\ \underline{\mathrm{let}\ \ \mathsf{bang}\, a = ba \ \mathrm{in}\ a : A}$$

$$duplicate : !_{\pi \cdot \rho} A \to !_\pi !_\rho A$$

$$duplicate = \lambda ba.\ \underline{\mathrm{let}\ \ \mathsf{bang}\, a = ba \ \mathrm{in}\ \ \mathsf{bang}(\mathsf{bang}\, \underline{a}) : !_\pi !_\rho A}$$

Substitution

Well scoped substitution

$$m \Rightarrow n :\equiv (x \in n) \to \mathrm{Tm}\ m\ \mathrm{syn}$$

Substitution

Well scoped substitution

$$m \Rightarrow n :\equiv (x \in n) \to \mathrm{Tm}\ m\ \mathrm{syn}$$

Typed sub'n refines scoped sub'n ($\sigma : m \Rightarrow n$)

$$\Gamma_m \Rightarrow^t_\sigma \Gamma_n :\equiv ((x : T) \in \Gamma_n) \to \Gamma_m \vdash \sigma\ x \in T$$

Substitution

Well scoped substitution

$$m \Rightarrow n :\equiv (x \in n) \to \mathrm{Tm}\ m\ \mathrm{syn}$$

Typed sub'n refines scoped sub'n ($\sigma : m \Rightarrow n$)

$$\Gamma_m \Rightarrow^t_\sigma \Gamma_n :\equiv ((x : T) \in \Gamma_n) \to \Gamma_m \vdash \sigma\ x \in T$$

Resourced sub'n refines typed sub'n ($\sigma t : \Gamma_m \Rightarrow^t_\sigma \Gamma_n$)

$$\begin{aligned}\Delta_m \Rightarrow^r_{\sigma t} \Delta_n :\equiv\ & (\Delta' : n \to RCtx\, m) \\ & \times \left(\Delta_m \leq \sum_{x^\rho \in \Delta_n} \rho \cdot \Delta'_x\right) \\ & \times \left((x^\rho \in \Delta_n) \to \Delta'_x \vdash \sigma t\ x\right)\end{aligned}$$

Substitution

Well scoped substitution

$$m \Rightarrow n :\equiv (x \in n) \to \mathrm{Tm}\ m\ \mathrm{syn}$$

Typed sub'n refines scoped sub'n ($\sigma : m \Rightarrow n$)

$$\Gamma_m \Rightarrow^t_\sigma \Gamma_n :\equiv ((x : T) \in \Gamma_n) \to \Gamma_m \vdash \sigma\ x \in T$$

Resourced sub'n refines typed sub'n ($\sigma t : \Gamma_m \Rightarrow^t_\sigma \Gamma_n$)

$$\begin{aligned}\Delta_m \Rightarrow^r_{\sigma t} \Delta_n :\equiv\ & (\Delta' : \mathrm{Mat}\ R\ (m, n) \\ & \times \left(\Delta_m \leq \Delta' \Delta_n\right) \\ & \times \left((x^\rho \in \Delta_n) \to \Delta'_x \vdash \sigma t\ x\right)\end{aligned}$$

Resourced substitution example

Resourced sub'n refines typed sub'n ($\sigma t : \Gamma_m \Rightarrow^t_\sigma \Gamma_n$)

$$\begin{aligned}\Delta_m \Rightarrow^r_{\sigma t} \Delta_n :\equiv (\Delta' &: \mathrm{Mat}\ R\ (m, n)) \\ &\times \left(\Delta_m \le \Delta'\Delta_n\right) \\ &\times \left((x^\rho \in \Delta_n) \to \Delta'_x \vdash \sigma t\ x\right)\end{aligned}$$

Resourced substitution example

Resourced sub'n refines typed sub'n ($\sigma t : \Gamma_m \Rightarrow^t_\sigma \Gamma_n$)

$$\begin{aligned}\Delta_m \Rightarrow^r_{\sigma t} \Delta_n :\equiv{} & (\Delta' : \mathrm{Mat}\ R\ (m, n)) \\ & \times \left(\Delta_m \leq \Delta' \Delta_n\right) \\ & \times \left((x^\rho \in \Delta_n) \to \Delta'_x \vdash \sigma t\ x\right)\end{aligned}$$

Variable rule

$$\frac{\Delta^\Gamma \leq \underline{0}, x \overset{1}{:} S, \underline{0}}{\Delta^\Gamma \vdash x \in S}$$

Resourced substitution example

Resourced sub'n refines typed sub'n ($\sigma t : \Gamma_m \Rightarrow^t_\sigma \Gamma_n$)

$$\begin{aligned}\Delta_m \Rightarrow^r_{\sigma t} \Delta_n :\equiv (\Delta' &: \mathrm{Mat}\ R\ (m, n) \\ &\times \left(\Delta_m \leq \Delta' \Delta_n\right) \\ &\times \left((x^\rho \in \Delta_n) \to \Delta'_x \vdash \sigma t\ x\right)\end{aligned}$$

Variable rule

$$\frac{\Delta^\Gamma \leq \underline{0}, x \overset{1}{:} S, \underline{0}}{\Delta^\Gamma \vdash x \in S}$$

Identity substitution

- $\mathrm{id}_\Delta :\equiv (\Delta', \mathit{prf}, \mathrm{var})$ where $\Delta'_x :\equiv \underline{0}, x^1, \underline{0}$

Resourced substitution example

Resourced sub'n refines typed sub'n $(\sigma t : \Gamma_m \Rightarrow^t_\sigma \Gamma_n)$

$$\begin{aligned}\Delta_m \Rightarrow^r_{\sigma t} \Delta_n :\equiv\ & (\Delta' : \mathrm{Mat}\ R\ (m, n) \\ & \times \left(\Delta_m \leq \Delta' \Delta_n\right) \\ & \times \left((x^\rho \in \Delta_n) \to \Delta'_x \vdash \sigma t\ x\right)\end{aligned}$$

Variable rule

$$\frac{\Delta^\Gamma \leq \underline{0}, x \overset{1}{:} S, \underline{0}}{\Delta^\Gamma \vdash x \in S}$$

Identity substitution

- $\mathrm{id}_\Delta :\equiv (\Delta', \mathit{prf}, \mathrm{var})$ where $\Delta'_x :\equiv \underline{0}, x^1, \underline{0}$
- prf is $\Delta = I\Delta = \Delta'\Delta$

Fundamental lemma, revisited

Fundamental lemma, revisited

- $[\![tt]\!] : [\![\Gamma]\!] \to [\![S]\!]$ – standard Set semantics

Fundamental lemma, revisited

- $[\![tt]\!] : [\![\Gamma]\!] \to [\![S]\!]$ – standard Set semantics
  - $[\![S \mathbin{\&} T]\!] = [\![S \otimes T]\!] = [\![S]\!] \times [\![T]\!]$

Fundamental lemma, revisited

- $\llbracket tt \rrbracket : \llbracket \Gamma \rrbracket \to \llbracket S \rrbracket$ – standard Set semantics
  - $\llbracket S \mathbin{\&} T \rrbracket = \llbracket S \otimes T \rrbracket = \llbracket S \rrbracket \times \llbracket T \rrbracket$
  - $\llbracket !_\rho S \rrbracket = \llbracket S \rrbracket$

Fundamental lemma, revisited

- $[\![tt]\!] : [\![\Gamma]\!] \to [\![S]\!]$ – standard Set semantics
  - $[\![S \mathbin{\&} T]\!] = [\![S \otimes T]\!] = [\![S]\!] \times [\![T]\!]$
  - $[\![!_\rho S]\!] = [\![S]\!]$
- $[\![S]\!]^R : \mathcal{W} \to [\![S]\!] \times [\![S]\!] \to \mathrm{Prop}$
- $[\![\Delta^\Gamma]\!]^R : \mathcal{W} \to [\![\Gamma]\!] \times [\![\Gamma]\!] \to \mathrm{Prop}$

Fundamental lemma, revisited

- $[\![tt]\!] : [\![\Gamma]\!] \to [\![S]\!]$ – standard Set semantics
  - $[\![S \mathbin{\&} T]\!] = [\![S \otimes T]\!] = [\![S]\!] \times [\![T]\!]$
  - $[\![!_\rho S]\!] = [\![S]\!]$
- $[\![S]\!]^R : \mathcal{W} \to [\![S]\!] \times [\![S]\!] \to \mathrm{Prop}$
- $[\![\Delta^\Gamma]\!]^R : \mathcal{W} \to [\![\Gamma]\!] \times [\![\Gamma]\!] \to \mathrm{Prop}$
- If tt is well resourced,
  $\forall w.\ \forall(\gamma, \gamma') \in [\![\Delta^\Gamma]\!]^R\ w.\ ([\![tt]\!]\ \gamma, [\![tt]\!]\ \gamma') \in [\![S]\!]^R\ w$

Fundamental lemma, revisited

- $[\![tt]\!] : [\![\Gamma]\!] \to [\![S]\!]$ – standard Set semantics
  - $[\![S \mathbin{\&} T]\!] = [\![S \otimes T]\!] = [\![S]\!] \times [\![T]\!]$
  - $[\![!_\rho S]\!] = [\![S]\!]$
- $[\![S]\!]^R : \mathcal{W} \to [\![S]\!] \times [\![S]\!] \to \mathrm{Prop}$
- $[\![\Delta^\Gamma]\!]^R : \mathcal{W} \to [\![\Gamma]\!] \times [\![\Gamma]\!] \to \mathrm{Prop}$
- If tt is well resourced,
  $\forall w.\ \forall(\gamma, \gamma') \in [\![\Delta^\Gamma]\!]^R\ w.\ ([\![tt]\!]\ \gamma, [\![tt]\!]\ \gamma') \in [\![S]\!]^R\ w$
- Consequences:
  - Worlds are bags of keys, semiring counts usages
    $\implies$ all functions are permutations
  - Worlds are trivial, semiring tracks polarity
    $\implies$ all functions are monotonic

Fundamental lemma, revisited

- $[\![tt]\!] : [\![\Gamma]\!] \to [\![S]\!]$ – standard Set semantics
  - $[\![S \mathbin{\&} T]\!] = [\![S \otimes T]\!] = [\![S]\!] \times [\![T]\!]$
  - $[\![!_\rho S]\!] = [\![S]\!]$
- $[\![S]\!]^R : \mathcal{W} \to [\![S]\!] \times [\![S]\!] \to \mathrm{Prop}$
- $[\![\Delta^\Gamma]\!]^R : \mathcal{W} \to [\![\Gamma]\!] \times [\![\Gamma]\!] \to \mathrm{Prop}$
- If tt is well resourced,
  $\forall w.\ \forall(\gamma, \gamma') \in [\![\Delta^\Gamma]\!]^R\ w.\ ([\![tt]\!]\ \gamma, [\![tt]\!]\ \gamma') \in [\![S]\!]^R\ w$
- Consequences:
  - Worlds are bags of keys, semiring counts usages
    $\implies$ all functions are permutations
  - Worlds are trivial, semiring tracks polarity
    $\implies$ all functions are monotonic
  - Worlds are distances, semiring tracks distances
    $\implies$ all functions are non-expansive

Fundamental lemma, revisited

- $[\![tt]\!] : [\![\Gamma]\!] \to [\![S]\!]$ – standard Set semantics
  - $[\![S \,\&\, T]\!] = [\![S \otimes T]\!] = [\![S]\!] \times [\![T]\!]$
  - $[\![!_\rho S]\!] = [\![S]\!]$
- $[\![S]\!]^R : \mathcal{W} \to [\![S]\!] \times [\![S]\!] \to \mathrm{Prop}$
- $[\![\Delta^\Gamma]\!]^R : \mathcal{W} \to [\![\Gamma]\!] \times [\![\Gamma]\!] \to \mathrm{Prop}$
- If tt is well resourced,
  $\forall w.\ \forall(\gamma, \gamma') \in [\![\Delta^\Gamma]\!]^R\ w.\ ([\![tt]\!]\ \gamma, [\![tt]\!]\ \gamma') \in [\![S]\!]^R\ w$
- Consequences:
  - Worlds are bags of keys, semiring counts usages
    $\implies$ all functions are permutations
  - Worlds are trivial, semiring tracks polarity
    $\implies$ all functions are monotonic
  - Worlds are distances, semiring tracks distances
    $\implies$ all functions are non-expansive
  - Worlds are sets of security levels, semiring same
    $\implies$ high security data do not interfere with low security data

Conclusion

- https://github.com/laMudri/quantitative

Conclusion

- `https://github.com/laMudri/quantitative`
- Abadi, Banerjee, Heintze 1999 – A Core Calculus of Dependency
- Reed, Pierce 2010 – Distance Makes the Types Grow Stronger
- Arntzenius 2018 – Type inference for monotonicity

Conclusion

- `https://github.com/laMudri/quantitative`
- Abadi, Banerjee, Heintze 1999 – A Core Calculus of Dependency
- Reed, Pierce 2010 – Distance Makes the Types Grow Stronger
- Arntzenius 2018 – Type inference for monotonicity
- Staged computation?
- More problems?