

# Indecibilidade da Inferência de Tipos com Recursão Polimórfica

Gabriela Moreira e Luiz Gustavo Eburneo

`gabrielaamoreira05@gmail.com,`  
`botucacontato@gmail.com`

Departamento de Ciência da Computação  
Centro de Ciências e Tecnológicas  
Universidade do Estado de Santa Catarina

7 de Dezembro de 2017

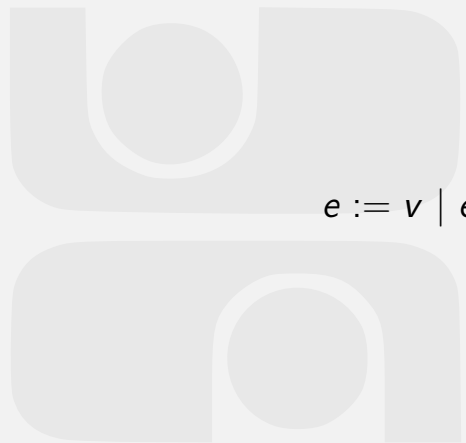
Sistemas de Tipos

Problemas

Redução

Conclusão

# Cálculo Lambda


$$e := v \mid e_1 e_2 \mid \lambda x. e$$

# Sistema de Tipos

## Sentenças (*judgments*)

$$\Gamma \vdash e : \sigma$$

## Regras

$$\frac{\Gamma_1 \vdash e_1 : \sigma_1 \quad \Gamma_2 \vdash e_2 : \sigma_2 \quad \dots \quad \Gamma_n \vdash e_n : \sigma_n}{\Gamma \vdash e : \sigma}$$

# Cálculo Lambda Simplesmente Tipado

Nesse sistema, além de expressões lambda, são definidos tipos simples. Um tipo simples  $\tau$  tem forma:

$$\tau := \alpha \mid \tau \rightarrow \tau'$$

Onde  $\alpha$  é uma variável de tipo e  $\tau \rightarrow \tau'$  é o tipo de uma função que recebe um parâmetro do tipo  $\tau$  e retorna um tipo  $\tau'$ .

**Regras:**

$$\Gamma \vdash x : \tau \text{ (VAR)} \quad \{x : \tau\} \in \Gamma$$

$$\frac{\Gamma \vdash e : \tau \rightarrow \tau' \quad \Gamma \vdash e' : \tau}{\Gamma \vdash e \ e' : \tau'} \text{ (APP)}$$

$$\frac{\Gamma, x : \tau' \vdash e : \tau}{\Gamma \vdash \lambda x. e : \tau' \rightarrow \tau} \text{ (ABS)}$$

## Exemplo para regra APP

```
chr 97 = 'a'  
chr :: Int -> Char  
97 :: Int
```

## Exemplo para regra APP

```
chr 97 = 'a'  
chr :: Int -> Char  
97 :: Int
```

```
Então  
chr 97 :: Char
```

## Operador de Ponto Fixo

Tipos recursivos podem ser definidos através do operador de ponto fixo, com o uso de expressões de tipo  $\mu\alpha.\tau$ , que denota o isomorfismo dos tipos que satisfazem a equação

$$\mu\alpha.\tau \cong \{\alpha \mapsto \mu\alpha.\tau\}\tau.$$

As regras para definir tal isomorfismo precisam ser acrescentadas ao sistema:

$$\frac{\Gamma \vdash e : \{\alpha \mapsto \mu\alpha.\tau\}\tau}{\Gamma \vdash \text{fold } e : \mu\alpha.\tau} \text{ (FOLD)} \quad \frac{\Gamma \vdash e : \mu\alpha.\tau}{\Gamma \vdash \text{unfold } e : \{\alpha \mapsto \mu\alpha.\tau\}\tau} \text{ (UNFOLD)}$$



# Tipos Parametricamente Polimórficos

Para a interpretação de tipos parametricamente polimórficos, são definidos tipos quantificados. Assim, tem-se a combinação de tipos simples ( $\tau$ ) e tipos quantificados ( $\sigma$ ):

$$\begin{aligned}\tau &::= \alpha \mid \tau \rightarrow \tau' \\ \sigma &::= \tau \mid \forall \alpha. \sigma\end{aligned}$$

Por exemplo, o tipo da função identidade:

```
id ::  $\forall \alpha. \alpha \rightarrow \alpha$   
id 'a' :: Char  
id 1 :: Int
```

## Regras Sistema Damas-Milner

Assim, são acrescentadas três novas regras para quantificar, instanciar e utilizar tipos parametricamente polimórficos:

$$\frac{\Gamma \vdash e : \forall \alpha. \sigma \quad (\alpha \notin \text{ftv}(\Gamma))}{\Gamma \vdash e : \sigma} \text{ (GEN)} \quad \frac{\Gamma \vdash e : \sigma \quad (\sigma \geq \sigma')}{\Gamma \vdash e : \sigma'} \text{ (INST)}$$

$$\frac{\Gamma \vdash e : \sigma \quad \Gamma, x : \sigma \vdash e' : \tau}{\Gamma \vdash \text{let } x = e \text{ in } e' : \tau} \text{ (LET)}$$

# Sistema Milner-Mycroft

A modificação proposta por Alan Mycroft (1984) trata-se apenas de aceitar o uso de tipos quantificados com operadores de ponto fixo, o que depende de uma alteração nas regras (*FOLD*) e (*UNFOLD*):

$$\frac{\sigma \vdash e : \{\sigma \mapsto \mu\sigma.\tau\}\tau}{\sigma \vdash \text{fold } e : \mu\sigma.\tau} \text{ (FOLD')} \quad \frac{\Gamma \vdash e : \mu\sigma.\tau}{\sigma \vdash \text{unfold } e : \{\sigma \mapsto \mu\sigma.\tau\}\tau} \text{ (UNFOLD')}$$

Mas a inferência de tipos para esse sistema é um problema indecidível.

# Inferência de tipos com recursão polimórfica

Um tipo recursivamente polimórfico é aquele que ocorre em sua própria definição. Uma árvore binária totalmente balanceada pode assumir essa representação:

```
data Arvore a = Cons a (Arvore (a,a)) | Nil
```

Uma árvore de altura igual a três pode ser descrita como:

```
Cons 1 (Cons (2,3) (Cons ((4,5),(6,7)) Nil))
```

## Inferência de tipos com recursão polimórfica

Pode-se declarar uma função para calcular a altura de uma árvore:

```
altura :: Arvore a -> Int
altura Nil = 0
altura (Cons n a) = 1 + altura a
```

```
n = 1,           a = Cons (2,3) (Cons ((4,5),(6,7)) Nil)
n = (2,3),       a = Cons ((4,5),(6,7)) Nil
n = ((4,5),(6,7)), a = Nil
```

O problema em questão é a inferência do tipo de expressões como a função *altura*, onde ocorre recursão polimórfica.

# Unificação

Dois tipos  $\sigma$  e  $\sigma'$  são ditos unificáveis se e somente se existe uma substituição  $S$  da forma  $\{\alpha_1 \mapsto \tau_1, \alpha_2 \mapsto \tau_2, \dots, \alpha_n \mapsto \tau_n\}$  tal que  $S(\sigma) = S(\sigma')$ .

$$S(a, \text{Char}) = S(\text{Int}, b)$$

$$S = \{a \mapsto \text{Int}, b \mapsto \text{Char}\}$$

# Semi Unificação

Dado um conjunto de pares de tipos  $\{(M_1, N_1), (M_2, N_2), \dots, (M_n, N_n)\}$  onde  $M_i \geq N_i$  ( $M_i$  é mais geral que  $N_i$ ). O conjunto é dito semi unificável se e somente se existe um semi unificador  $S$  e um conjunto de substituições  $\{S_1, S_2, \dots, S_n\}$  tal que

$$S_i(S(M_i)) = S(N_i)$$

## Semi Unificação - SEI's

Esse problema pode ser descrito em sistemas de equações e inequações (SEI) de tipos. Seja um SEI  $\{M_{01} = N_{01}, M_{02} = N_{02}, \dots, M_{0n} = N_{0n}, M_1 \geq N_1, M_2 \geq N_2, \dots, M_m \geq N_m\}$ . Um semi unificador  $S$  e um conjunto de  $m$  substituições  $\{S_1, S_2, \dots, S_m\}$  para esse SEI são tais que:

$$\begin{aligned} S(M_{0i}) &= S(N_{0i}) \\ S(M_j) &\geq S(N_j) \\ S_j((S(M_j))) &= S(N_j) \end{aligned}$$

Para  $i$  de 1 a  $n$  e  $j$  de 1 a  $m$ .



## Semi Unificação $\leq_R$ Inferência de tipos com RP

Kfoury et al. (1993) provou que o problema da semi unificação é indecidível.

Será mostrada uma função computável que transforma uma instância do problema da semi unificação - um SEI - em uma expressão do sistema Milner-Mycroft. Essa função é de tal forma que o SEI será semi unificável se e somente se a expressão for tipável no sistema.

## Milner-Mycroft Extendido

$$\frac{(\tau, \vec{\tau}) \geq (\tau', \vec{\tau})}{\Gamma \cup \{x : \tau\}, \vec{\tau} \vdash x : \tau'} \text{ (TAUT')}$$

$$\frac{\Gamma \cup \{x : \tau_x\}, (\vec{\tau}, \tau_x) \vdash e : \tau \quad \tau' = \tau_x \rightarrow \tau}{\Gamma, \vec{\tau} \vdash \lambda x. e : \tau'} \text{ (ABS')}$$

$$\frac{\Gamma, \vec{\tau} \vdash e : \tau \quad \Gamma, \vec{\tau} \vdash e' : \tau' \quad \tau = \tau' \rightarrow \tau''}{\Gamma, \vec{\tau} \vdash (ee') : \tau''} \text{ (APP')}$$

$$\frac{\Gamma, \vec{\tau} \vdash e : \tau \quad \Gamma \cup \{x : \tau_x\}, \vec{\tau} \vdash e' : \tau' \quad \tau = \tau', \tau' = \tau''}{\Gamma, \vec{\tau} \vdash \text{let } x = e \text{ in } e' : \tau''} \text{ (LET')}$$

$$\frac{\Gamma \cup \{x : \tau_x\}, \vec{\tau} \vdash e : \tau \quad \tau_x = \tau, (\tau, \vec{\tau}) \geq (\tau', \vec{\tau})}{\Gamma, \vec{\tau} \vdash \text{fix } x. e : \tau'} \text{ (FIX-P')}$$

## Pares dependentes

$$\begin{aligned}(e, e1) &= \lambda x. xee' \\ e.1 &= e(\lambda x. \lambda y. x) \\ e.2 &= e(\lambda x. \lambda y. y)\end{aligned}$$

com  $x \notin \text{ftv}(e) \cup \text{ftv}(e')$ . As regras para derivação de tipos são definidas:

$$\frac{\Gamma \vdash e : \tau \quad \Gamma \vdash e' : \tau'}{\Gamma \vdash (e, e') : (\tau, \tau')_{\tau''}} \text{ (PAIR')}$$

$$\frac{\Gamma \vdash e : (\tau, \tau')_{\tau}}{\Gamma \vdash e.1 : \tau} \text{ (PROJ')} \quad \frac{\Gamma \vdash e : (\tau, \tau')_{\tau'}}{\Gamma \vdash e.2 : \tau'}$$

Escreve-se  $e \doteq e'$  para a expressão  $\lambda y. (ye, ye')$  com  $y \notin \text{ftv}(e) \cup \text{ftv}(e')$ .

## Conversão de Equações

Dada uma equação de tipos  $M = N$ , constroem-se um contexto  $\Gamma$  e uma expressão  $\lambda \vec{x}.(m \doteq n)$  onde  $m$  e  $n$  são expressões tal que  $\Gamma \vdash m : M$ ,  $\Gamma \vdash n : N$ , e  $\vec{x} = \text{ftv}(M) \cup \text{ftv}(N)$ . Ou seja,  $M = N$  é semi unificável se e somente se  $\lambda \vec{x}.(m \doteq n)$  é tipável. Como  $M = N$  é uma equação, seu semi unificador é igual ao seu unificador, formando o teorema a seguir.

### Theorem

$M = N$  é unificável  $\iff \lambda \vec{x}.(m \doteq n)$  é tipável

## Conversão de Inequações

Sem perder generalidade, conforme Henglein (1993), é possível reduzir a prova e mostrar a conversão de um sistema de apenas duas inequações. Assim, um sistema de inequações  $M_1 \geq N_1, M_2 \geq N_2$  é convertido para uma expressão do sistema Milner-Mycroft da forma.

$$e = \text{fix} f. \lambda \vec{x}. K(m_1 \dot{=} m_2)(\lambda \vec{y}. ((f \vec{y}).1 \dot{=} n_1), \lambda \vec{y}. ((f \vec{y}).2 \dot{=} n_2)))$$

Onde  $\vec{x} = \text{ftv}(M_1) \cup \text{ftv}(M_2) \cup \text{ftv}(N_1) \cup \text{ftv}(N_2)$  e  $K = \lambda x. \lambda y. x$ .  
Assim como um contexto  $\Gamma$  tal que  
 $\Gamma \vdash m_1 : M_1, \Gamma \vdash m_2 : M_2, \Gamma \vdash n_1 : N_1$  e  $\Gamma \vdash n_2 : N_2$ .

### Theorem

$\{M_1 \geq N_1, M_2 \geq N_2\}$  é *semi unificável*  $\iff e$  é *tipável*

# Conclusão

## Semi-algoritmos

Não param respondendo corretamente para todas as instâncias do problema.

Fim =)