Cálculo lambda II

Extensiones del cálculo lambda

Paradigmas de Lenguajes de Programación

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Septiembre 2017

En clases anteriores...

- Introducción a $\text{C-}\lambda^{b}$ como lenguaje representativo del paradigma funcional
- **Sintaxis**. Expresiones de tipos y términos
- **Sistema de tipado**. Contexto y juicios de tipado. Axiomas y reglas.
- **Semántica operacional**. Formas normales y valores. Interpretación.

Sintaxis

Expresiones de tipos

$$\sigma ::= Bool \mid \sigma \to \tau$$

Términos

$$M ::= x \mid true \mid false \mid if\ M\ then\ P\ else\ Q \mid \lambda x : \sigma.M \mid M\ N$$

¿Son correctas estas expresiones?

- λx:Bool.x
- $(\lambda x$:Bool.$x)\ x$
- $y\ x$
- $if\ true\ then\ x\ else\ false$
- λx:true.x
- λx:Bool.λy:Bool.$if\ x\ then\ true\ else\ y$

Los tipos nos permiten caracterizar las expresiones del lenguaje que tienen sentido

Axiomas y reglas de tipado

$$\frac{x : \sigma \in \Gamma}{\Gamma \triangleright x : \sigma} \qquad \frac{}{\Gamma \triangleright \mathit{true} : \mathit{Bool}} \qquad \frac{}{\Gamma \triangleright \mathit{false} : \mathit{Bool}}$$

$$\frac{\Gamma \cup \{x : \sigma\} \triangleright M : \tau}{\Gamma \triangleright \lambda x : \sigma.M : \sigma \to \tau} \qquad \frac{\Gamma \triangleright M : \sigma \to \tau \quad \Gamma \triangleright N : \sigma}{\Gamma \triangleright M\ N : \tau}$$

$$\frac{\Gamma \triangleright M : \mathit{Bool} \quad \Gamma \triangleright P : \sigma \quad \Gamma \triangleright Q : \sigma}{\Gamma \triangleright \mathit{if}\ M\ \mathit{then}\ P\ \mathit{else}\ Q : \sigma}$$

Semántica

La semántica operacional consiste en interpretar a los términos como estados de una máquina abstracta y definir una función de transición que indica, dado un estado, cuál es el siguiente estado

Evaluación

- La semántica nos permite interpretar las expresiones correctas (términos tipados) del lenguaje.
- El objetivo es saber como se evaluan o ejecutan los términos para conocer su *significado*.
- La semántica que usamos es "en un paso" (*small-step*).

Valores

¿Qué significan estas expresiones?

- $\lambda x{:}\text{Bool}.x$
- $(\lambda x{:}\text{Bool}.x)\ \mathit{true}$
- $(\lambda x{:}\text{Bool} \to \text{Bool}.x)\ (\lambda x{:}\text{Bool}.x)\ \mathit{false}$

¿Qué son los valores?

Los valores son las expresiones con sentido "directo". Son los posibles resultados de los programas **correctos**.

Los posibles valores en el cálculo λ presentado hasta ahora son:

$$V \ ::= \ \mathit{true} \mid \mathit{false} \mid \lambda x : \sigma.M$$

Semántica operacional (en un paso)

$$\frac{M \to M'}{M\ N \to M'\ N} \qquad \frac{N \to N'}{V\ N \to V\ N'} \qquad \frac{}{(\lambda x : \sigma.M)\ V \to M[x \leftarrow V]}$$

$$\frac{M \to M'}{\mathit{if}\ \ M\ \ \mathit{then}\ N\ \mathit{else}\ O \to \mathit{if}\ \ M'\ \ \mathit{then}\ N\ \mathit{else}\ O}$$

$$\frac{}{\mathit{if}\ \ \mathit{true}\ \ \mathit{then}\ N\ \mathit{else}\ O \to N} \qquad \frac{}{\mathit{if}\ \ \mathit{false}\ \ \mathit{then}\ N\ \mathit{else}\ O \to O}$$

Extendiendo el C-λ ...

- **Primera extensión:** los naturales
- ¿Qué se agregó?

Sintaxis para cálculo λ con pares

¿Qué hay que agregar?

- ...términos para representar el constructor y los observadores

$$M ::= \ldots \mid \, < M, N > \, \mid \pi_1(M) \mid \pi_2(M)$$

- ...y un tipo para estas nuevas expresiones

$$\sigma ::= \ldots \mid \sigma \times \tau$$

Reglas de tipado para pares

¿Qué hay que agregar?

- Al menos una regla por cada forma nueva de sintaxis, porque cada una de ellas precisa poder ser tipada.
- Notar que, de no hacerlo, sería imposible construir términos tipables (útiles) con dicha forma.

Regla de tipado para el constructor

$$\frac{\Gamma \triangleright M : \sigma \quad \Gamma \triangleright N : \tau}{\Gamma \triangleright < M, N >: \sigma \times \tau}$$

Reglas de tipado para las proyecciones

$$\frac{\Gamma \triangleright M : \sigma \times \tau}{\Gamma \triangleright \pi_1(M) : \sigma}$$

$$\frac{\Gamma \triangleright N : \sigma \times \tau}{\Gamma \triangleright \pi_2(N) : \tau}$$

Semántica para pares

¿Qué reglas hay que agregar?

- Necesitamos reducir todos los pares *con sentido* que no sean valores.

¿Cuáles son los valores?

- Empecemos por ahí entonces...

Extensión de los valores

$$V \ ::= \ ... \mid \ < V, W >$$

Reglas de semántica para pares

Ahora sí, las reglas

$$\frac{M \to M'}{< M, N > \to < M', N >} \qquad \frac{N \to N'}{< V, N > \to < V, N' >}$$

Reglas de semántica para las proyecciones

$$\frac{M \to M'}{\pi_1(M) \to \pi_1(M')} \qquad \frac{M \to M'}{\pi_2(M) \to \pi_2(M')}$$

$$\frac{}{\pi_1(< V, W >) \to V} \qquad \frac{}{\pi_2(< V, W >) \to W}$$

Sintaxis para cálculo λ con árboles binarios

¿Qué hay que agregar?

- ...términos para representar los constructores y observadores
  $M ::= \ldots \mid Nil_\sigma \mid Bin(M, N, O) \mid root(M) \mid right(M) \mid left(M) \mid isNil(M)$

- ...y un tipo para estas nuevas expresiones

$$\sigma ::= \ldots \mid AB_\sigma$$

Reglas de tipado para árboles binarios

¿Qué hay que agregar?

- Como antes: una regla por cada forma nueva de sintaxis, porque cada una de ellas precisa poder ser tipada.

Reglas de tipado para los constructores

$$\frac{}{\Gamma \triangleright Nil_\sigma : AB_\sigma}$$

$$\frac{\Gamma \triangleright M : AB_\sigma \quad \Gamma \triangleright O : AB_\sigma \quad \Gamma \triangleright N : \sigma}{\Gamma \triangleright Bin(M, N, O) : AB_\sigma}$$

- Nil_σ es una constante diferente según el tipo σ.
  - ¡No tenemos polimorfismo!
- Para *Bin*, en cambio, el tipo queda determinado por el tipo de los subtérminos.

Reglas de tipado para los observadores

$$\frac{\Gamma \triangleright M : AB_\sigma}{\Gamma \triangleright \mathit{root}(M) : \sigma} \qquad \frac{\Gamma \triangleright M : AB_\sigma}{\Gamma \triangleright \mathit{isNil}(M) : \mathit{Bool}}$$

$$\frac{\Gamma \triangleright M : AB_\sigma}{\Gamma \triangleright \mathit{left}(M) : AB_\sigma} \qquad \frac{\Gamma \triangleright M : AB_\sigma}{\Gamma \triangleright \mathit{right}(M) : AB_\sigma}$$

Semántica para árboles binarios

- Primero, empecemos por los valores:

$$V \ ::= \ ... \mid Nil_\sigma \mid Bin(V, W, Y)$$

Reglas de semántica para los constructores

$$\frac{M \to M'}{Bin(M, N, O) \to Bin(M', N, O)}$$

$$\frac{N \to N'}{Bin(V, N, O) \to Bin(V, N', O)}$$

$$\frac{O \to O'}{Bin(V, W, O) \to Bin(V, W, O')}$$

Reglas de semántica para los observadores (1/2)

$$\frac{M \to M'}{\mathit{left}(M) \to \mathit{left}(M')} \qquad \frac{M \to M'}{\mathit{right}(M) \to \mathit{right}(M')}$$

$$\frac{M \to M'}{\mathit{root}(M) \to \mathit{root}(M')} \qquad \frac{M \to M'}{\mathit{isNil}(M) \to \mathit{isNil}(M')}$$

Reglas de semántica para los observadores (2/2)

$$\frac{}{isNil(Nil_\sigma) \to true} \qquad \frac{}{isNil(Bin(V, W, Y)) \to false}$$

$$\frac{}{left(Bin(V, W, Y)) \to V} \qquad \frac{}{right(Bin(V, W, Y)) \to Y}$$

$$\frac{}{root(Bin(V, W, Y)) \to W}$$

Otra forma de proyectar/observar

- Vamos a ver otra forma de representar proyectores u observadores más prolija y que requiere menos reglas (aunque una construcción más sofisticada).

- Usamos los árboles nuevamente para ejemplificar, de manera que se pueda comparar correctamente ambas formas.

Sintaxis para cálculo λ con árboles binarios bis

- Los tipos quedan igual que en el caso anterior:

$$\sigma ::= \ldots \mid AB_\sigma$$

- Y los términos,

  $M ::= \ldots \mid Nil_\sigma \mid Bin(M, N, O) \mid$
  $Case_{AB_\sigma}\ M\ of\ Nil \rightsquigarrow N\ ; Bin(m, n, o) \rightsquigarrow O$

  Aquí las minúsculas (m,n,o) representan **variables**.

Reglas de tipado para árboles binarios bis

- Para los constructores son las que ya teníamos.

$$\frac{}{\Gamma \triangleright Nil_\sigma : AB_\sigma}$$

$$\frac{\Gamma \triangleright M : AB_\sigma \quad \Gamma \triangleright O : AB_\sigma \quad \Gamma \triangleright N : \sigma}{\Gamma \triangleright Bin(M, N, O) : AB_\sigma}$$

Regla de tipado para el *Case*

$$\frac{\Gamma \triangleright M : AB_\sigma \quad \Gamma \triangleright N : \tau \qquad \Gamma \cup \{m : AB_\sigma, n : \sigma, o : AB_\sigma\} \triangleright O : \tau}{\Gamma \triangleright Case_{AB_\sigma}\ M\ of\ \ Nil \leadsto N\ ; Bin(m, n, o) \leadsto O : \tau}$$

Semántica para los árboles binarios bis

- Tenemos los mismos valores que antes:

$$V \ ::= \ ... \mid \mathit{Nil}_\sigma \mid \mathit{Bin}(V, W, Y)$$

Reglas de semántica para los constructores

- Análogas a las que ya teníamos.

$$\frac{M \to M'}{Bin(M, N, O) \to Bin(M', N, O)}$$

$$\frac{N \to N'}{Bin(V, N, O) \to Bin(V, N', O)}$$

$$\frac{O \to O'}{Bin(V, W, O) \to Bin(V, W, O')}$$

Reglas de semántica para el *Case*

$$\frac{M \to M'}{\begin{array}{c} Case_{AB_\sigma}\ M\ of\ Nil \rightsquigarrow N\ ; Bin(m,n,o) \rightsquigarrow O \\ \to \\ Case_{AB_\sigma}\ M'\ of\ Nil \rightsquigarrow N\ ; Bin(m,n,o) \rightsquigarrow O \end{array}}$$

$$\frac{}{Case_{AB_\sigma}\ Nil_\sigma\ of\ Nil \rightsquigarrow N\ ; Bin(m,n,o) \rightsquigarrow O \to N}$$

$$\frac{}{\begin{array}{c} Case_{AB_\sigma}\ Bin(V,W,Y)\ of\ Nil \rightsquigarrow N\ ; Bin(m,n,o) \rightsquigarrow O \\ \to O\{m \leftarrow V, n \leftarrow W, o \leftarrow Y\} \end{array}}$$

Ejercicio

Objetivo: Extender el lenguaje para soportar una estructura **fold** que servirá como esquema de recursión para los árboles binarios

- Tipos
  $\sigma ::= \ldots \mid AB_\sigma$

- Términos

  $M ::= \ldots \mid Nil_\sigma \mid Bin(M, N, O) \mid$
  $Fold\ M\ base = N;\ rec\ r_i\ e\ r_d = O$

2010-1c-1r

En la próxima clase...

Inferencia de tipos para C-λ.

- Mecanismo para reconstruir el tipo de una expresión cualquiera sin anotaciones de tipo.

¡Eso es todo!

$$(\lambda x : Clase.fin\ x)\ LambdaCalculo2$$