Recursión en Cálculo-λ puro

José de Jesús Lavalle Martínez

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Fundamentos de Lenguajes de Programación CCOS 255

Contenido

Motivación

Ejemplo 1

Considere la función sum' que toma una lista de números naturales como entrada y devuelve la suma de los elementos de la lista.

$$\text{sum'} = \lambda l.\text{ite } (\text{empty } l)(0)(\text{plus } (\text{head } l)(\text{sum'}(\text{tail } l))).$$

$$\begin{aligned}
\text{sum'}[1,2] &= \lambda l.\text{ite } (\text{empty } l)(0)(\text{plus } (\text{head } l)(\text{sum'}(\text{tail } l)))[1,2] \\
&\to_\beta \text{ite } (\text{empty } [1,2])(0)(\text{plus } (\text{head } [1,2])(\text{sum'}(\text{tail } [1,2]))) \\
&\to_\beta^* (\text{plus } (\text{head } [1,2])(\text{sum'}(\text{tail } [1,2]))) \\
&\to_\beta^* (\text{plus } (1)(\text{sum'}(\text{tail } [1,2]))) \\
&\to_\beta^* (\text{plus } (1)(\text{sum'}([2])))
\end{aligned}$$

Definición punto fijo

Definición 2

Un **punto fijo** de una función f es un elemento de su dominio que cumple lo siguiente:

$$f(x) = x.$$

Ejemplos puntos fijos

Ejemplo 3

- 5 es un punto fijo de la función constante $x \mapsto 5$.

Ejemplos puntos fijos

Ejemplo 3

- 5 es un punto fijo de la función constante $x \mapsto 5$.
- 0 y 1 son puntos fijos de la función $x \mapsto x^2$.

Ejemplos puntos fijos

Ejemplo 3

- 5 es un punto fijo de la función constante $x \mapsto 5$.
- 0 y 1 son puntos fijos de la función $x \mapsto x^2$.
- La función identidad $x \mapsto x$ tiene como puntos fijos a todos los elementos de su dominio.

Observación 4

Cualquier término-λ se puede ver como una función cuyo dominio es el conjunto de términos-λ. Un punto fijo de un término-λ F es así un término-λ M para el cual se tiene lo siguiente:

$$FM =_\beta M.$$

Ejemplo 5

El combinador $I = \lambda x.x$ tiene infinitos puntos fijos: todos los términos-λ.

Todo término-λ tiene un punto fijo I

Teorema 6

Todo término-λ tiene un punto fijo. Aún más, existe un método para construir un punto fijo para un término-λ F dado.

Esto se logra usando el llamado **combinador de punto fijo**. Este combinador de punto fijo lo creó Curry y es el término-λ Y definido como sigue:

$$Y = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx)).$$

Todo término-λ tiene un punto fijo II

Tome un término-λ arbitrario F, entonces tenemos:

$$\begin{aligned}
YF &= \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))F \\
&\rightarrow_\beta (\lambda x.F(xx))(\lambda x.F(xx)) \\
&\rightarrow_\beta F((\lambda x.F(xx))(\lambda x.F(xx))) \\
&\leftarrow_\beta F((\lambda f.(\lambda x.f(xx))(\lambda x.f(xx)))F) \\
&= F(YF)
\end{aligned}$$

Todo término-λ tiene un punto fijo II

Tome un término-λ arbitrario F, entonces tenemos:

$$\begin{aligned} {\color{red}YF} &= \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))F \\ &\rightarrow_\beta (\lambda x.F(xx))(\lambda x.F(xx)) \\ &\rightarrow_\beta F((\lambda x.F(xx))(\lambda x.F(xx))) \\ &\leftarrow_\beta F((\lambda f.(\lambda x.f(xx))(\lambda x.f(xx)))F) \\ &= {\color{green}F(YF)} \end{aligned}$$

Así, $F(YF) =_\beta YF$, esto es, YF es un punto fijo de F. Note que no tenemos reducción-β es decir $F(YF) \not\twoheadrightarrow^*_\beta YF$, sólo tenemos conversión-β.

Todo término-λ tiene un punto fijo II

Tome un término-λ arbitrario F, entonces tenemos:

$$\begin{aligned} {\color{red}YF} &= \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))F \\ &\rightarrow_\beta (\lambda x.F(xx))(\lambda x.F(xx)) \\ &\rightarrow_\beta F((\lambda x.F(xx))(\lambda x.F(xx))) \\ &\leftarrow_\beta F((\lambda f.(\lambda x.f(xx))(\lambda x.f(xx)))F) \\ &= {\color{green}F(YF)} \end{aligned}$$

Así, $F(YF) =_\beta YF$, esto es, YF es un punto fijo de F. Note que no tenemos reducción-β es decir $F(YF) \not\rightarrow^*_\beta YF$, sólo tenemos conversión-β.

También note que YF admite una secuencia infinita de la forma:

$$\begin{aligned} F((\lambda x.F(xx))(\lambda x.F(xx))) &=_\beta F(F((\lambda x.F(xx))(\lambda x.F(xx)))) \\ &=_\beta F(F(F((\lambda x.F(xx))(\lambda x.F(xx))))) \\ &\vdots \end{aligned}$$

Todo término-λ tiene un punto fijo III

El término-λ Y no es el único combinador de punto fijo en el cálculo-λ. El siguiente combinador de punto fijo fue creado por Turing:

$$T = (\lambda x.\lambda y.y(xxy))(\lambda x.\lambda y.y(xxy))$$

Todo término-λ tiene un punto fijo III

El término-λ Y no es el único combinador de punto fijo en el cálculo-λ. El siguiente combinador de punto fijo fue creado por Turing:

$$T = (\lambda x.\lambda y.y(xxy))(\lambda x.\lambda y.y(xxy))$$

Escribimos $T = tt$ con $t = (\lambda x.\lambda y.y(xxy))$. Tome un término-$\lambda$ arbitrario F. Tenemos lo siguiente:

$$\begin{aligned}
TF &= (\lambda x.\lambda y.y(xxy))(\lambda x.\lambda y.y(xxy))F \\
&\to_\beta ((\lambda y.y(tty)))F \\
&\to_\beta F(ttF) \\
&= F(TF).
\end{aligned}$$

Todo término-λ tiene un punto fijo III

El término-λ Y no es el único combinador de punto fijo en el cálculo-λ. El siguiente combinador de punto fijo fue creado por Turing:

$$T = (\lambda x.\lambda y.y(xxy))(\lambda x.\lambda y.y(xxy))$$

Escribimos $T = tt$ con $t = (\lambda x.\lambda y.y(xxy))$. Tome un término-$\lambda$ arbitrario F. Tenemos lo siguiente:

$$\begin{aligned} TF &= (\lambda x.\lambda y.y(xxy))(\lambda x.\lambda y.y(xxy))F \\ &\to_\beta ((\lambda y.y(tty)))F \\ &\to_\beta F(ttF) \\ &= F(TF). \end{aligned}$$

Así,

$$TF =_\beta F(TF),$$

por lo tanto TF es un punto fijo de F.

Observación 7

Un combinador de punto fijo se puede usar para representar en el cálculo-λ funciones definidas recursivamente.

Suponga que una función G está definida en términos de si misma, esto es, tenemos lo siguiente:

$$G =_\beta \cdots G \cdots$$

En el lado derecho reemplazamos todas las ocurrencias de G por g, al término-λ que resulta le llamamos E. Entonces, podemos reescribir la ecuación anterior de la siguiente forma:

$$G =_\beta (\lambda g.E)G.$$

Funciones recursivas II

$$G =_\beta (\lambda g.E)G$$

Es decir, buscamos un término-λ G que sea un punto fijo del término-λ $(\lambda g.E)$.

Como sabemos un combinador de punto fijo nos permite construir un punto fijo, en este caso para el término-λ $(\lambda g.E)$, así $Y(\lambda g.E)$ es un punto fijo de $(\lambda g.E)$.

Por lo tanto, el término-λ G que buscamos lo podemos definir como:

$$G = Y(\lambda g.E).$$

Ejemplo funciones recursivas I

Ejemplo 8

Considere la función sum' que toma una lista de números naturales como entrada y devuelve la suma de los elementos de la lista.

$$\text{sum'} = \lambda l.\text{ite (empty } l)(0)(\text{plus (head } l)(\text{sum'(tail } l))).$$

Ejemplo 8

Considere la función sum' que toma una lista de números naturales como entrada y devuelve la suma de los elementos de la lista.

$$\text{sum'} = \lambda l.\text{ite }(\text{empty } l)(0)(\text{plus }(\text{head } l)(\text{sum'}(\text{tail } l))).$$

Solución: Tenemos que hacer lo que vimos en el caso general, buscamos todas las apariciones de sum' del lado derecho de la definición, las cambiamos digamos que por s y el término-λ resultante pasará a ser el cuerpo de λs, para obtener:

Ejemplo 8

Considere la función sum' que toma una lista de números naturales como entrada y devuelve la suma de los elementos de la lista.

$$\text{sum'} = \lambda l.\text{ite }(\text{empty } l)(0)(\text{plus }(\text{head } l)(\text{sum'}(\text{tail } l))).$$

Solución: Tenemos que hacer lo que vimos en el caso general, buscamos todas las apariciones de sum' del lado derecho de la definición, las cambiamos digamos que por s y el término-λ resultante pasará a ser el cuerpo de λs, para obtener:

$$\text{sum''} = \lambda s.\lambda l.\text{ite }(\text{empty } l)(0)(\text{plus }(\text{head } l)(s(\text{tail } l))).$$

Ejemplo funciones recursivas I

Ejemplo 8

Considere la función sum' que toma una lista de números naturales como entrada y devuelve la suma de los elementos de la lista.

$$\text{sum'} = \lambda l.\text{ite (empty } l)(0)(\text{plus (head } l)(\text{sum'(tail } l))).$$

Solución: Tenemos que hacer lo que vimos en el caso general, buscamos todas las apariciones de sum' del lado derecho de la definición, las cambiamos digamos que por s y el término-λ resultante pasará a ser el cuerpo de λs, para obtener:

$$\text{sum''} = \lambda s.\lambda l.\text{ite (empty } l)(0)(\text{plus (head } l)(s(\text{tail } l))).$$

Ahora sí, la definición recursiva de sum es:

$$\text{sum} = Y(\lambda s.\lambda l.\text{ite (empty } l)(0)(\text{plus (head } l)(s(\text{tail } l)))).$$

Ejemplo funciones recursivas II

Ejemplo 9

Considere la función map' que toma como entradas una función y una lista, regresa una lista que contiene el resultado de aplicar la función a cada elemento de la lista.

$$\text{map'} = \lambda fl.l(\lambda htz.\text{cons } (fh)(\text{map' } ft))\text{nil}.$$

Solución: Buscamos las apariciones de map' en el lado derecho de la definición y las cambiamos por m, el término-λ resultante será el cuerpo de λm, así:

$$\text{map''} = \lambda m.\lambda fl.l(\lambda htz.\text{cons } (fh)(mft))\text{nil}.$$

Así la definición recursiva de map es:

$$\text{map} = Y(\lambda m.\lambda fl.l(\lambda htz.\text{cons } (fh)(mft))\text{nil}).$$

$$
\begin{aligned}
\text{sum'} &= \lambda l.\text{ite (empty } l)(0)(\text{plus (head } l)(\text{sum'(tail } l))). \\
\text{sum''} &= \lambda s.\lambda l.\text{ite (empty } l)(0)(\text{plus (head } l)(s(\text{tail } l))). \\
\text{sum} &= Y(\lambda s.\lambda l.\text{ite (empty } l)(0)(\text{plus (head } l)(s(\text{tail } l)))). \\
\text{sum}[1,2] &= Y(\lambda s.\lambda l.\text{ite (empty } l)(0)(\text{plus (head } l)(s(\text{tail } l))))[1,2] \\
&=_\beta \text{sum''}(Y\text{sum''})[1,2] \\
&= \lambda s.\lambda l.\text{ite (empty } l)(0)(\text{plus (head } l)(s(\text{tail } l)))(Y\text{sum''})[1,2] \\
&\to_\beta \lambda l.\text{ite (empty } l)(0)(\text{plus (head } l)((Y\text{sum''})(\text{tail } l)))[1,2] \\
&\to_\beta \text{ite (empty } [1,2])(0)(\text{plus (head } [1,2])((Y\text{sum''})(\text{tail } [1,2]))) \\
&\to_\beta^* (\text{plus (head } [1,2])((Y\text{sum''})(\text{tail } [1,2]))) \\
&\to_\beta^* (\text{plus } (1)((Y\text{sum''})(\text{tail } [1,2]))) \\
&\to_\beta^* (\text{plus } (1)((Y\text{sum''})[2])) \\
&=_\beta (\text{plus } (1)\ \text{sum''}(Y\text{sum''})[2]) \\
&= (\text{plus } (1)\lambda s.\lambda l.\text{ite (empty } l)(0)(\text{plus (head } l)(s(\text{tail } l)))(Y\text{sum''}))[2] \\
&\to_\beta (\text{plus } (1)\lambda l.\text{ite (empty } l)(0)(\text{plus (head } l)((Y\text{sum''})(\text{tail } l)))[2]) \\
&\to_\beta (\text{plus } (1)\text{ite (empty } [2])(0)(\text{plus (head } [2])((Y\text{sum''})(\text{tail } [2]))))
\end{aligned}
$$

Funcionamiento del operador de punto fijo II

$$
\begin{aligned}
&= (\text{plus } (1)\text{ite } (\text{empty } [2])(0)(\text{plus } (\text{head } [2])((Y\text{sum''})(\text{tail } [2]))))\\
&\to_\beta^* (\text{plus } (1)(\text{plus } (\text{head } [2])((Y\text{sum''})(\text{tail } [2]))))\\
&\to_\beta^* (\text{plus } (1)(\text{plus } (2)((Y\text{sum''})(\text{tail } [2]))))\\
&\to_\beta^* (\text{plus } (1)(\text{plus } (2)((Y\text{sum''})([]))))\\
&=_\beta (\text{plus } (1)(\text{plus } (2)\ \text{sum''}(Y\text{sum''})[]))\\
&= (\text{plus } (1)(\text{plus } (2)\lambda s.\lambda l.\text{ite } (\text{empty } l)(0)(\text{plus } (\text{head } l)(s(\text{tail } l)))(Y\text{sum''}))[])\\
&\to_\beta (\text{plus } (1)(\text{plus } (2)\lambda l.\text{ite } (\text{empty } l)(0)(\text{plus } (\text{head } l)((Y\text{sum''})(\text{tail } l)))[]))\\
&\to_\beta (\text{plus } (1)(\text{plus } (2)\text{ite } (\text{empty } [])(0)(\text{plus } (\text{head } [])((Y\text{sum''})(\text{tail } [])))))\\
&\to_\beta^* (\text{plus } (1)(\text{plus } (2)(0)))\\
&\to_\beta^* (\text{plus } (1)(2))\\
&\to_\beta^* (3).
\end{aligned}
$$

1. Haga cinco reducciones-β a $Y(\lambda z.z)$.
2. Para las siguientes funciones defina términos-λ, use el operador de punto fijo para dar sus definiciones recursivas, como en los ejemplos 8 y 9.
   1. $$\text{suma}(m,n) = \begin{cases} n & \text{si } m = 0, \\ \text{suma}(m-1,n)+1 & \text{si } m \geq 1. \end{cases}$$
   2. $$\text{prod}(m,n) = \begin{cases} 0 & \text{si } m = 0, \\ \text{prod}(m-1,n)+n & \text{si } m \geq 1. \end{cases}$$
   3. $$\text{fact}(n) = \begin{cases} 1 & \text{si } n = 0, \\ \text{fact}(n-1)*n & \text{si } n \geq 1. \end{cases}$$

Ejercicios II

3. Muestre el funcionamiento del operador de punto fijo para:
   1. suma 2 5,
   2. prod 2 5,
   3. fact 3.

Ejercicios III

Note que la función sum del ejemplo 8 se escribe matemáticamente así:

$$\mathsf{sum}(l) = \begin{cases} 0 & \text{si } l = \mathsf{nil}, \\ \mathsf{head}(l) + \mathsf{sum}(\mathsf{tail}(l)) & \text{si } l \neq \mathsf{nil}. \end{cases}$$

Fin del curso

Muchas gracias