Tarea 0

Universidad Nacional Autónoma de México
Facutlad de Ciencias
Programación Declarativa

Fong Baeza Luis Fernando Yang
313320679
fernandofong@ciencias.unam.mx

21 de Febrero 2018

1. Dada la definición recursiva de los naturales y la función suma, demostrar por inducción que:

1. $\forall n \in \mathbb{N}$ [n + 0 = n]
   Demostración por inducción sobre n:

   a) Caso base: n = 0
   Entonces siguiendo la definición de la suma hay que ver que 0 + 0 = 0 que por cómo está definida, por el lado izquierdo, si el lado izquierdo es cero, se cumple, entonces para el caso base se cumple.

   b) Hipótesis de Inducción; Supongamos que vale para n

   c) Paso Inductivo: Demostrar que se vale para S(n)
   Entonces queremos demostrar que S(n) + 0 = S(n), siguiendo la definición tenemos que S(n) + 0 = S(n + 0) pero por H.I. tenemos que n + 0 = n entonces tenemos que S(n) + 0 = S(n) $\square$

2. $\forall n \in \mathbb{N}$ [S(n) = n + 1]
   Demostración por inducción sobre n:

   a) Caso base: n = 0
   Queremos demostrar que S(0) = 0 + 1, entonces por la definición de la función suma tenemos que 0 + 1 = 1, entonces tenemos que S(0) = 1 por transitividad de la igualdad, el cual, cumple para nuestro caso base.

   b) Hipótesis de Inducción; Supongamos que vale para n.

   c) Paso Inductivo: Demostrar que S(n + 1) = n + 2
   Queremos demostrar que S(n + 1) = n + 2, entonces podemos verlo como S(n + 1) = (n + 1) + 1 pero ya sabemos por H.I. que (n + 1) = S(n) entonces S(S(n)) = S(n) + 1 que es justo lo que queríamos demostrar. $\square$

2. Dar una definición recursiva de la función $* : \mathbb{N} \to \mathbb{N}$

Entonces definimos a * como:

S(0) * n = n
S(m) * n = (n * m) + n

1. Demostrar que $\forall n \in \mathbb{N}$ [n * 0 = 0] ya que $\forall n \in \mathbb{N}$ [0 * n = 0]

   Por inducción sobre n.

   a) Caso base: n = 0
   Demostraremos que 0 * 0 = 0, pero siguiendo la definición de la función, encontramos que esto es cierto si n = 0.

   b) Hipótesis de Inducción; supongamos que vale para n.

   c) Paso inductivo: Demostrar que (n + 1) * 0 = 0
   Entonces tenemos por la demostración anterior que n + 1 = S(n) entonces queremos demostrar que S(n) * 0 = 0, entonces siguiendo la definición tenemos que S(n) * 0 = n * (0 + 0) pero por definición de suma tenemos que 0 + 0 = 0, entonces tenemos que S(n) * 0 = n * 0 y por H.I. tenemos que n * 0 = 0, entonces S(n) * 0 = 0. $\square$

2. $\forall n \in \mathbb{N}$ [n * 1 = 1]

   Inducción sobre n.

   a) Caso base: n = 1
   Vamos a ver que en efecto 1*1 = 1, entonces por la definición, como S(0) = 1, lo podemos ver como S(0) = n = 1, entonces tenemos seguido de la definición que S(0)*1 = 1.

   b) Hipótesis de inducción; supongamos que n * 1 = n.

   c) Paso Inductivo: Demostrar S(n) * 1 = S(n)
   Entonces hay que demostrar que S(n) * 1 = S(n), entonces siguiendo la definición tenemos que S(n) * 1 = (n * 1) + 1, pero por hipótesis de inducción sabemos que n * 1 = n, entonces sustituimos en la igualdad anterior y tenemos que S(n) * 1 = n + 1 y por definición n + 1 = S(n), entonces sustituyendo tenemos que S(n) * 1 = S(n). $\square$

3. Definir una función recursiva que nos diga si 2 naturales son iguales.

Podemos definirla de la siguiente forma:

f(0, 0) = True   Son iguales si llegan al mismo tiempo al 0.
f(S(n), 0) = False   Son distintos si uno llegó antes.
f(0, S(m)) = False
f(S(n), S(m)) = f(n, m)   Se calcula a ver quién llega primero o si llegan al mismo tiempo.

4. Sea la función misteriosa, descubrir que hace.

Si m = 0, entonces h(n, m) = 1, por cómo está definida, falta esta parte.

h(n, 2k) = h(n*n, k) para algun k $\in \mathbb{Z}$
h(n, m) = n * (h(n, m - 1))

Observando el caso base, tenemos que si m = 0, entonces regresamos uno, sin importar n, sabiendo que cualquier número elevado a la 0, es 1, entonces al menos empatan en el caso base, podemos llegar a pensar que son la misma función. Para los otros casos, sabemos que exp(n, m) = n * exp (n, m - 1) que al final de cuentas es multiplicar n sí misma, n veces. Ahora la función misteriosa h, si recibe una n impar, va a multiplicar n * h(n, (m - 1)) entonces eso haría que m - 1 fuera par, entonces va a ir calculando n * h(n*n, k) tal que 2*k = m - 1, hasta llegar a 1, que al final regresaría el resultado de elevar n a la m.

5. Dada la definición de concatenación sobre listas, demostrar por inducción sobre listas que:

1. $\forall l \in \mathbb{L}$[l ++ [ ] = l] Entonces lo haremos por inducción estructural sobre l.
   a) Caso base: l = [ ] Tenemos que l = [ ] pero queremos demostrar que [ ] ++ [ ] = [ ], entonces por la definición se sigue esto es cierto, puesto que está definida por la izquierda, pero la izquierda es la vacía, entonces se usa la definición y se llega a que [ ] ++ [ ] = [ ].
   b) Hipótesis de Inducción; supongamos que vale para una lista xs.
   c) Paso inductivo: Demostrar que (x:xs) ++ [ ] = (x:xs) En este caso ya no tenemos una lista vacía, entonces otra vez siguiendo la definición de la función ++, tenemos que (x:xs) ++ [ ] = x:(xs ++ [ ]) pero por H.I. tenemos que xs ++ [ ] = xs, entonces sustituyendo en la igualdad tenemos que (x:xs) ++ [ ] = (x:xs) probando la igualdad a la que queríamos llegar. $\square$

2. $\forall l1, l2, l3 \in \mathbb{L}$ [(l1 ++ l2) ++ l3 = l1 ++ (l2 ++ l3)] Lo haremos por inducción sobre l1
   a) Caso base: l1 = [ ] Vamos a demostrar que ([ ] ++ l2) ++ l3 = [ ] ++ (l2 ++ l3), por definición de concatenación, tenemos que [ ] ++ l2 = l2, entonces en la igualdad original tenemos que l2 ++ l3 = [ ] ++ (l2 ++ l3), pero otra vez, usando la definición, tenemos que [ ] ++ (l2 ++ l3) = l2 ++ l3, cumpliéndose la igualdad.
   b) Hipótesis de Inducción; supongamos que vale para una lista xs.
   c) Paso Inductivo: Demostrar que ((x:xs) ++ l2) ++ l3 = (x:xs) ++ (l2 ++ l3) Usando otra vez las definiciones de concatenación tenemos que para (x:xs) ++ l2 = x:(xs ++ l2) entonces al concatenarla con la lista l3, primero por definición tenemos que es x:((xs ++ l2) ++ l3) pero también por H.I. sabemos que ((xs ++ l2) ++ l3) conmuta, es decir que es igual a (xs ++ (l2 ++ l3)), entonces tenemos que x:(xs ++ (l2 ++ l3)) quien es por definición (x:xs) ++ (l2 ++ l3), entonces se cumple que ((x:xs) ++ l2) ++ l3 = (x:xs) ++ (l2 ++ l3) $\square$

6. Definir recursivamente la reversa de una lista y probar 2 propiedades.

Primero haremos la definición recursiva de reversa de una lista.
rev([ ]) = [ ]
rev(x:xs) = rev(xs) ++ [x]

1. Demuestre que $\forall l \in \mathbb{L}$ [rev(rev(l)) = l] Por Inducción Estructural sobre l.

   a) Caso base: l = [ ] Probaremos que rev(rev([ ])) = [ ], entonces rev([ ]) = [ ] por la definición pura, entonces aplicando reversa a ambos lados de la igualdad tenemos que rev(rev([ ])) = rev([ ]), pero por la igualdad anterior tenemos que se cumple que rev(rev([ ])) =[ ].

   b) Hipótesis de Inducción; suponer que vale para una lista xs.

   c) Paso Inductivo: Demostrar que vale para x:xs. Basta con probar que rev(rev(x:xs)) = x:xs, entonces, primero, la definición de reversa, nos asegura que rev(x:xs) = rev(xs) ++ [x], observemos que el lado derecho de la igualdad es una lista de igual forma, entonces podemos aplicar reversa otra vez teniendo que rev(rev(x:xs)) = rev(rev(xs)++[x]) pero por cómo está definida la función reversa, primero hacemos rev(rev xs) que ya se vió que es xs, pero lo vamos a ir pegando al final de toda la lista que es [x], entonces tenemos por H.I. que rev(rev(xs) ++ [x]) = [x]++xs que es igual a x:xs, entonces tenemos que rev(rev(x:xs)) = x:xs. $\square$