

FUNDAMENTOS DE COMPUTACIÓN

Repartido: Bool Parte 2 - La matemática de los booleanos

1. Las afirmaciones del Lenguaje Matemático

Desde la introducción, este curso se ha propuesto como un curso de Matemática, en particular de la Matemática de los programas.

¿Cuál es entonces la Matemática de los booleanos? ¿Qué teoremas interesantes podemos producir?

Al estudiar propiedades de los booleanos y de los programas que operan con ellos, vamos a estar sobre todo interesados en **demostrar afirmaciones** de la forma: **Para todo booleano b se cumple $P(b)$** , donde P es una propiedad concreta. En adelante, esto será escrito así ($\forall b :: \text{Bool}$) $P(b)$.

La primera observación importante es que las afirmaciones a considerar **no son parte del cálculo lambda** que estamos desarrollando. Los tipos y objetos de este lenguaje son, obviamente, mencionados en esas afirmaciones, pero ellas son formuladas en el **metalenguaje o lenguaje matemático**. Es decir, así como venimos desde el comienzo hablando en general, acerca del cálculo lambda, lo continuaremos haciendo, pero ahora formulando y demostrando propiedades de sus objetos con rigurosidad matemática.

El cuerpo de la afirmación típica a considerar, es decir $P(b)$, será una afirmación que menciona a la variable b . Posibles ejemplos de estas propiedades podrían ser:

$\text{not} (\text{not } b) = b$ o $b \ \&\& \ \text{False} = \text{False}$.

O sea, $P(b)$ afirma algo acerca del booleano genérico b . Podemos entonces separar P y b simplemente: todo lo que queda de $P(b)$ cuando omitimos b , es la propiedad P .

En los ejemplos precedentes:

- Si $P(b)$ es **$\text{not} (\text{not } b) = b$** , entonces P es **$\text{not} (\text{not } \bullet) = \bullet$**

- Si $P(b)$ es **$b \ \&\& \ \text{False} = \text{False}$** , entonces P es **$\bullet \ \&\& \ \text{False} = \text{False}$** ,

donde hemos dejado un hueco (\bullet) en el lugar del booleano genérico b .

En otras palabras, P es una expresión de nuestro lenguaje matemático que espera ser completada (saturada) con un individuo (en este caso, cualquier expresión booleana) para convertirse en una afirmación.

Este tipo de expresiones se llaman propiedades o predicados. Un ejemplo de predicado en el contexto de la aritmética podría ser la siguiente expresión: **\bullet es par**.

Cuando el hueco es rellenado por un individuo (en este caso, un entero), se obtiene una afirmación, como por ejemplo **5 es par**, que puede ser verdadera o falsa. Si es verdadera, decimos que el individuo **5** cumple la propiedad (en este caso sabemos que no se cumple).

Lo importante acerca de estas afirmaciones de la forma $P(b)$ es que debemos ser capaces de **sustituir en ellas la variable b por otras expresiones**, según nos convenga para proceder con el razonamiento.

Por suerte, ya estamos entrenados en estos menesteres, gracias al estudio de la sustitución que hemos hecho para las expresiones del cálculo lambda .

2. Demostraciones por casos

Ahora bien, cuando adjuntamos a $P(b)$ el prefijo $(\forall b :: \text{Bool})$ estamos haciendo otra afirmación: **$P(b)$ es verdadera para todos los casos posibles de $b :: \text{Bool}$** , o, en otras palabras, que todos los booleanos cumplen la propiedad P .

Recordemos que hay infinitas expresiones booleanas, por ejemplo: `True`, `not True`, `not(not True)`, `not(not(not True))`, etc... Pero por la definición dada del tipo `Bool`, sabemos que cualquier expresión booleana va a ser por definición igual a alguno de los valores `False` o `True` (y no hay otra opción!).

Esto justifica el siguiente método de demostración de propiedades de los booleanos:

Para demostrar que todos los booleanos cumplen una cierta propiedad P , alcanza claramente con demostrar:

- el caso **False**: o sea, demostrar **$P(\text{False})$**
- el caso **True**: o sea, demostrar **$P(\text{True})$** .

Aclaremos esto con un ejemplo de una propiedad a demostrar por el método recién presentado. Demostraremos el lema sobre la función **`not`**.

Copiamos aquí la definición de **`not`** dada para tenerla cerca:

`not :: Bool → Bool`

`not = λb → case b of {False → True ; True → False }`

La propiedad a demostrar se escribe como el siguiente lema:

Lema_{not}: $(\forall b :: \text{Bool}) \text{ not (not } b) = b$.

Observar que la propiedad P a demostrar es **`not (not ●) = ●`**.

Los casos a demostrar se verán en la siguiente demostración del lema.

Demostración: Por casos en $b :: \text{Bool}$.

Caso $b = \text{False}$: Hay que demostrar **`not (not False) = False`** {o sea, $P(\text{False})$ }

Dem. `not (not False)`
= (def. not)
`not ((λb → case b of {False → True ; True → False }) False)`
= (regla β)
`not (case False of {False → True ; True → False })`
= (case False)
`not True`
= (def. not, regla β , case True)
`False`.

Caso b = True: Hay que demostrar **not (not True) = True** {o sea, $P(True)$ }

Dem. not (not True)
= (def. not, regla β , case True)
not False
= (def. not, regla β , case False)
True.

□

Con estos dos casos acabamos de demostrar que para cualquier booleano b se cumple que $\text{not}(\text{not } b) = b$.

Cuando debamos demostrar una igualdad entre dos expresiones e1 y e2, vamos a utilizar la forma de demostración **ecuacional directa**. O sea, partimos de e1, y paso a paso vamos obteniendo expresiones intermedias mediante igualdades debidamente justificadas hasta llegar a e2. Como la igualdad es una relación **transitiva**, si tenemos una cadena de expresiones iguales, se cumple que la primera (e1) es igual a la última (e2).

Con el mismo método podemos demostrar la segunda propiedad enunciada al principio:

Lema1_{&&}: $(\forall b :: \text{Bool}) \ b \ \&\& \ \text{False} = \text{False}.$

Acá la propiedad P a demostrar es $\bullet \ \&\& \ \text{False} = \text{False}.$

Recordemos la definición de la función (&&):

$(\&\&) :: \text{Bool} \rightarrow \text{Bool} \rightarrow \text{Bool}$

$(\&\&) = \lambda b1 \rightarrow \lambda b2 \rightarrow \text{case } b1 \text{ of } \{ \text{False} \rightarrow \text{False} ; \text{True} \rightarrow b2 \}$

Procedemos a demostrar este lema por casos:

Demostración: Por casos en $b :: \text{Bool}$.

Caso b = False: $\text{False} \ \&\& \ \text{False} = ? \text{False}$

Dem. $\text{False} \ \&\& \ \text{False}$
= (def. &&)
 $(\lambda b1 \rightarrow \lambda b2 \rightarrow \text{case } b1 \text{ of } \{ \text{False} \rightarrow \text{False} ; \text{True} \rightarrow b2 \}) \ \text{False} \ \text{False}$
= (regla β x 2)
 $\text{case False of } \{ \text{False} \rightarrow \text{False} ; \text{True} \rightarrow \text{False} \}$
= (case False)
False.

Caso b = True: $\text{True} \ \&\& \ \text{False} = ? \text{False}$

Dem. $\text{True} \ \&\& \ \text{False}$
= (def. &&, regla β x 2)
 $\text{case True of } \{ \text{False} \rightarrow \text{False} ; \text{True} \rightarrow \text{False} \}$
= (case True)
False.

□

Observar que no todas las demostraciones deben hacerse por casos en el booleano. Por ejemplo, el siguiente lema se demuestra directamente, sin necesidad de analizar el valor de b :

Lema2_{&&}: $(\forall b :: \text{Bool}) \text{ False } \&\& b = \text{False}$

Esta igualdad se va a cumplir aplicando directamente la definición de $\&\&$, por lo tanto, para demostrar $(\forall b :: \text{Bool}) P(b)$ en este caso, tomaremos un $b :: \text{Bool}$ genérico y demostraremos la igualdad:

Demostración: Sea $b :: \text{Bool}$ cualquiera {se elige un b genérico}

$\text{False } \&\& b$
= (def. $\&\&$, regla β x 2)
case False of {False \rightarrow False ; True $\rightarrow b$ }
= (case False)
False.

□

Del mismo modo que demostramos los lemas que dicen cómo se comporta el $\&\&$ cuando uno de sus argumentos es False, se pueden demostrar también los siguientes resultados para cuando uno de los argumentos es True:

Lema3_{&&}: $(\forall b :: \text{Bool}) b \&\& \text{True} = b$

Lema4_{&&}: $(\forall b :: \text{Bool}) \text{True } \&\& b = b$

Dejamos estas demostraciones como *ejercicio*.

Una de ellas sale directamente y la otra requiere una demostración por casos en b . ¿Se puede ver fácilmente cuál es cuál y por qué?

Vamos ahora a demostrar un resultado más interesante, como es el caso de la conmutatividad del $\&\&$:

Conm_{&&}: $(\forall b1 :: \text{Bool})(\forall b2 :: \text{Bool}) b1 \&\& b2 = b2 \&\& b1.$

La demostración se hará por casos en alguno de los booleanos $b1$ o $b2$.

La haremos por casos en $b1$, pero también podríamos hacerlo con $b2$.

Observar que al hacer casos en $b1$, la propiedad P que estamos demostrando es la siguiente: $P = (\forall b2 :: \text{Bool}) \bullet \&\& b2 = b2 \&\& \bullet.$

Veamos cómo sería la demostración:

Demostración: Por casos en **b1 :: Bool**.

Caso b1 = False: $(\forall b2 :: \text{Bool}) \text{False} \ \&\& \ b2 =? \ b2 \ \&\& \ \text{False}$

Acá podemos tomar un b2 cualquiera, y sabemos que (por los Lemas 1 y 2) ambas expresiones son iguales a False.

Una forma de escribir la demostración de este caso es entonces, del siguiente modo:

Dem. Sea $b2 :: \text{Bool}$ cualquiera.
False && b2
= (Lema2_{&&})
False
= (Lema1_{&&})
b2 && False.

Observar que estamos utilizando el Lema1_{&&} “al revés”, o sea de la forma $b2 \ \&\& \ \text{False} = \text{False}$.

Esto lo podemos hacer porque la igualdad es una relación **simétrica**, o sea, si **e1 = e2** entonces también se cumple que **e2 = e1**.

Otra forma de demostrar una igualdad es utilizando las propiedades que ya mencionamos de la igualdad (transitividad y simetría), y razonar **por reducción a la misma expresión**. Esto es: para demostrar que **e1 = e2**, demostramos que **existe una tercera expresión e3** tal que **e1 = e3** y **e2 = e3**. De esta última igualdad obtenemos que $e3 = e2$ (por simetría), y finalmente, como $e1 = e3$ y $e3 = e2$, obtenemos $e1 = e2$ por transitividad.

Vamos a realizar de este modo la demostración del caso **b1 = True** de la demostración de la conmutatividad del &&. Dividimos la hoja en dos, y avanzamos en cada lado mediante igualdades directas, hasta llegar a una misma expresión:

Caso b1 = True : $(\forall b2 :: \text{Bool}) \text{True} \ \&\& \ b2 =? \ b2 \ \&\& \ \text{True}$

Dem. Sea $b2 :: \text{Bool}$ cualquiera
True && b2 | b2 && True
= (Lema4_{&&}) | = (Lema3_{&&})
b2 | b2

Ambas expresiones son iguales por ser iguales a la misma expresión b2.



Vamos ahora a dedicarnos a la disyunción lógica, o sea el conectivo o:

((I)) :: Bool → Bool → Bool

((I)) = $\lambda b1 \rightarrow \lambda b2 \rightarrow \text{case } b1 \text{ of } \{ \text{False} \rightarrow b2 ; \text{True} \rightarrow \text{True} \}$

Probaremos directamente (sin lemas auxiliares) que esta función es conmutativa, o sea:

Conm_{||}: $(\forall b1 :: \text{Bool})(\forall b2 :: \text{Bool}) \ b1 \ || \ b2 = b2 \ || \ b1.$

Demostración: Por casos en **b1::Bool**.

Caso b1 = False: $(\forall b2 :: \text{Bool}) \ \text{False} \ || \ b2 =? \ b2 \ || \ \text{False}$

Observar que la parte izquierda de la igualdad reduce a b2, pero para saber a qué es igual la parte derecha, debemos hacer al hacer casos en b2.

Por casos en **b2::Bool**

Caso b2 = False: $\text{False} \ || \ \text{False} =? \ \text{False} \ || \ \text{False}$

Esta igualdad se cumple trivialmente, ya que la relación = es reflexiva (toda expresión es igual a sí misma).

Caso b2 = True: $\text{False} \ || \ \text{True} =? \ \text{True} \ || \ \text{False}$

$\text{False} \ \ \text{True}$		$\text{True} \ \ \text{False}$
$= (\text{def. } , \beta \times 2, \text{case})$		$= (\text{def. } , \beta \times 2, \text{case})$
True		True

Ambas expresiones son iguales por ser iguales a la misma expresión.

En el caso b1 = True también deberemos considerar casos en b2:

Caso b1 = True: $(\forall b2 :: \text{Bool}) \ \text{True} \ || \ b2 =? \ b2 \ || \ \text{True}$

Por casos en **b2::Bool**

Caso b2 = False: $\text{True} \ || \ \text{False} =? \ \text{True} \ || \ \text{False}$

En el caso anterior demostramos que $\text{False} \ || \ \text{True} = \text{True} \ || \ \text{False}$.

Como la igualdad es una relación simétrica, sabemos que entonces se cumple $\text{True} \ || \ \text{False} = \text{True} \ || \ \text{False}$, que es lo que queremos demostrar.

Caso b2 = True: $\text{True} \ || \ \text{True} =? \ \text{True} \ || \ \text{True}$

Se cumple por reflexividad del =.



Este es un ejemplo de demostración por casos en un booleano (b1), que tiene otra demostración por casos en otro booleano (b2) dentro de cada caso.