Inducción Estructural

Programación funcional

Repaso: Propiedades

- Normalización y Confluencia
- Transparencia referencial
- Extensionalidad
- ¿Qué es una propiedad?
- ¿Cómo podemos verificar si se cumple una propiedad?

¿Cómo probamos lo siguiente?

```
CONMUT ≜ (\x y -> x || y) ≡ (\x y -> y || x)
```

¿Cómo probamos lo siguiente?

$\texttt{CONMUT} \triangleq \texttt{(\textbackslash x y -> x || y)} \equiv \texttt{(\textbackslash x y -> y || x)}$

Por PE debe valer para todo `n` y `m` de tipo `Bool`:

$$\underline{\texttt{(\textbackslash x y -> x || y) n m}} \equiv \underline{\texttt{(\textbackslash x y -> y || x) n m}}$$

$$\beta \Rightarrow \texttt{n || m} \equiv \texttt{m || n}$$

¿Cómo probamos lo siguiente?

$$\texttt{CONMUT} \triangleq \texttt{(\textbackslash x y -> x || y)} \equiv \texttt{(\textbackslash x y -> y || x)}$$

Por PE debe valer para todo `n` y `m` de tipo `Bool`:

$$\underline{\texttt{(\textbackslash x y -> x || y) n m}} \equiv \underline{\texttt{(\textbackslash x y -> y || x) n m}}$$

$$\beta \Rightarrow \texttt{n || m} \equiv \texttt{m || n}$$

Separo en casos:

1. `n = True, m = True` ✓
2. `n = True, m = False` ✓
3. `n = False, m = True` ✓
4. `n = False, m = False` ✓

¿Y lo siguiente?

$$\texttt{EORO} \triangleq \texttt{(\textbackslash x -> even x || odd x)} \equiv \texttt{const True}$$

¿Y lo siguiente?

$\texttt{EORO} \triangleq \texttt{(\textbackslash x -> even x || odd x)} \equiv \texttt{const True}$

donde:

```
even Z = True
even (S m) = odd m

odd Z = False
odd (S m) = even m
```

¿Y lo siguiente?

`EORO` $\triangleq$ `(\x -> even x || odd x)` $\equiv$ `const True`

donde:

```
even Z = True
even (S m) = odd m

odd Z = False
odd (S m) = even m
```

- Por PE debe valer para todo `n :: Nat`

  `(\x -> even x || odd x) n` $\equiv$ `const True n`

  $\beta \Rightarrow$ `even n || odd n` $\equiv$ `const True n`

  `const` $\Rightarrow$ `even n || odd n` $\equiv$ `True`

- ¿Separación en casos?
  1. `n = Z`
  2. `???`

Principio de inducción estructural

Sea S un conjunto inductivo, decimos que una propiedad unaria P vale para todo $x \in S$ si se cumple que:

- $P(z)$ vale para cada elemento $z \in S$ que cumple con una regla base; y
- $P(y)$ vale para cada elemento $y \in S$ construido con una regla inductiva en base a elementos $y_1, \ldots, y_n$ si valen $P(y_1), \ldots, P(y_n)$.

Principio de inducción estructural

Sea S un conjunto inductivo, decimos que una propiedad unaria P vale para todo $x \in S$ si se cumple que:

- $P(z)$ vale para cada elemento $z \in S$ que cumple con una regla base; y
- $P(y)$ vale para cada elemento $y \in S$ construido con una regla inductiva en base a elementos $y_1, \ldots, y_n$ si valen $P(y_1), \ldots, P(y_n)$.

Esto nos permite hacer una separación en casos considerando:

- los casos base z; y
- los casos inductivos y (asumiendo que P vale para $y_1, \ldots, y_n$).

Consideremos EORO unaria:

$$\texttt{EORO(n)} \triangleq \texttt{even n || odd n} \equiv \texttt{True}$$

Caso Base: EORO(Z)

$$\begin{array}{l} \underline{\texttt{even Z}}\ \texttt{|| odd Z} \equiv \texttt{True} \\ \texttt{even.1} \Rightarrow \underline{\texttt{True || odd Z}} \equiv \texttt{True} \\ \texttt{(||).1} \Rightarrow \texttt{True} \equiv \texttt{True} \end{array}$$

Caso Inductivo:

- Hipótesis inductiva: EORO(m)
- Tesis inductiva: EORO(S m)

$$\begin{array}{l} \underline{\texttt{even (S m)}}\ \texttt{|| odd (S m)} \equiv \texttt{True} \\ \texttt{even.2} \Rightarrow \texttt{odd m ||}\ \underline{\texttt{odd (S m)}} \equiv \texttt{True} \\ \texttt{odd.2} \Rightarrow \underline{\texttt{odd m || even m}} \equiv \texttt{True} \\ \texttt{CONMUT} \Rightarrow \underline{\texttt{even m || odd m}} \equiv \texttt{True} \\ \texttt{HI} \Rightarrow \texttt{True} \equiv \texttt{True} \end{array}$$

¿Por qué vale `EORO(n)` para todo `n`?

Esquema Inductivo de los naturales

$$\mathsf{EI}_{\mathbb{N}} = \underbrace{\texttt{EORO(Z)}}_{\mathsf{CB}} \wedge (\forall \texttt{x} : \mathbb{N})(\underbrace{\texttt{EORO(x)}}_{\mathsf{HI}} \Rightarrow \underbrace{\texttt{EORO(S x)}}_{\mathsf{TI}})$$

¿Será cierto?

$$\texttt{P(xs)} \triangleq (\forall\, \texttt{ys : [a]})(\texttt{length (xs ++ ys)} \equiv \texttt{length xs + length ys})$$

¿Será cierto?

$$\texttt{P(xs)} \triangleq (\forall\, \texttt{ys} : \texttt{[a]})(\texttt{length (xs ++ ys)} \equiv \texttt{length xs + length ys})$$

Esquema Inductivo de las listas

$$\mathsf{EI}_{\texttt{[a]}} = \underbrace{\texttt{P([])}}_{\mathsf{CB}} \wedge \left(\forall \texttt{xs} : \texttt{[a]}\right)\left(\underbrace{\texttt{P(xs)}}_{\mathsf{HI}} \Rightarrow (\forall \texttt{x} : \mathsf{a})(\underbrace{\texttt{P(x:xs)}}_{\mathsf{TI}})\right)$$

¿Será cierto?

$$P(xs) \triangleq (\forall\, ys : [a])(\texttt{length}\ (xs \mathbin{++} ys) \equiv \texttt{length}\ xs + \texttt{length}\ ys)$$

Esquema Inductivo de las listas

$$\mathsf{EI}_{[a]} = \underbrace{\mathtt{P([])}}_{\mathsf{CB}} \land \left(\forall \mathtt{xs} : \mathtt{[a]}\right)\left(\underbrace{\mathtt{P(xs)}}_{\mathsf{HI}} \Rightarrow (\forall \mathtt{x} : \mathtt{a})(\underbrace{\mathtt{P(x:xs)}}_{\mathsf{TI}})\right)$$

En el pizarrón (observar que P es unaria).

Desafío de Gauss

Carl Friedrich Gauss

Demuestre `G(n)`:

```
gauss n ≡ div (n*(n+1)) 2
```

donde

```
gauss 0 = 0
gauss n = n + gauss (n-1)
```

Desafío de Gauss

Caso base: `G(0)`

$$
\begin{array}{rcl}
 & \texttt{gauss 0} \equiv & \frac{0(0+1)}{2} \\
\texttt{ARIT} \Rightarrow & \underline{\texttt{gauss 0}} \equiv & 0 \\
\texttt{gauss.1} \Rightarrow & 0 \equiv & 0
\end{array}
$$

Caso inductivo: $(\forall \texttt{n} : \mathbb{N})(\underbrace{\texttt{G(n)}}_{\text{HI}} \Rightarrow \underbrace{\texttt{G(n+1)}}_{\text{TI}})$

$$
\begin{array}{rcl}
 & \texttt{gauss (n+1)} \equiv & \frac{(n+1)(n+2)}{2} \\
\texttt{gauss.2} \Rightarrow & (n+1) + \texttt{gauss n} \equiv & \frac{(n+1)(n+2)}{2} \\
\texttt{HI} \Rightarrow & (n+1) + \frac{n(n+1)}{2} \equiv & \frac{(n+1)(n+2)}{2} \\
\texttt{ARIT} \Rightarrow & \frac{2(n+1)+n(n+1)}{2} \equiv & \frac{(n+1)(n+2)}{2} \\
\texttt{ARIT} \Rightarrow & \frac{(n+1)(n+2)}{2} \equiv & \frac{(n+1)(n+2)}{2}
\end{array}
$$