Ejercicio 10

Mediante el uso de razonamiento ecuacional e inducción estructural vamos a demostrar que: `indentar n (indentar m x) = indentar (n+m) x para todo n, m :: Int >= 0. x :: Doc`

Antes de continuar definimos la notación "\equiv" que hace referencia al símbolo de equivalencia ($\equiv$) Esta notación va a ser usada en múltiples ocaciones durante el razonamiento con el fin de no confundir la equivalencia con la igualdad (=)

Definiciones

Planteamos algunas definiciones de funciones y lemas de uso necesario en nuestro razonamiento ecuacional

Funciones

```
foldDoc :: b -> (String -> b -> b)  -> (Int -> b -> b) -> Doc -> b
foldDoc cVacio cTexto cLinea d = case d of
                                Vacio -> cVacio                        {FV}
                                Texto s doc -> cTexto s (rec doc)      {FT}
                                Linea ind doc -> cLinea ind (rec doc)  {FL}
                                where
                                  rec = foldDoc cVacio cTexto cLinea
{FREC}

indentar :: Int -> Doc -> Doc
indentar i = foldDoc Vacio (\s d -> Texto s d)
                           (\ind d -> Linea (ind + i) d)               {I'}
```

Para facilitar la notación vamos a definir las siguientes funciones auxiliares:

```
f :: String -> Doc -> Doc
f = \s d -> Texto s d                {F}

g :: Int ->  Int -> Doc -> Doc
g = \k i d -> Linea (k + i) d        {G}
```

De esta manera podemos reformular la función indentar:

```
indentar :: Int -> Doc -> Doc
indentar i = foldDoc Vacio f (g i)    {I}
```

Lemas

Previo a la resolución definimos 3 lemas: 1. `indentar k Vacio = Vacio para todo k :: Int >= 0 {L1}` 2. `indentar k (Texto s d) = Texto s (indentar k d) para todo k :: Int >= 0. s :: String. d :: Doc {L2}` 3. `indentar m (Linea k d) = Linea (m+k) (indentar m d) para todo m, k :: Int >= 0. d :: Doc {L3}`

Vamos a estar demostrando estos 3 lemas en un *apéndice al final del ejercicio*

Resolución

Aplicamos inducción estructural en x y definimos el *predicado unario* p de la siguiente manera:

`P(x)` $\equiv$ `indentar n (indentar m x) = indentar (m+n) x; m, n :: Int >= 0. x :: Doc`

Por el principio de inducción estructural sobre documentos, para ver que P(d) vale para todo d :: Doc basta con ver que vale:

- Caso Base: `P(Vacio)`
- Paso Inductivo A: `para todo d :: Doc. para todo s :: String. P(d) => P(Texto s d)`
  - Hipótesis Inductiva A: `P(d)`
  - Tesis Inductiva A: `P(Texto s d)`
- Paso Inductivo B: `para todo d :: Doc. para todo i :: Int. P(d) => P(Linea i d)`
  - Hipótesis Inductiva B: `P(d)`
  - Tesis Inductiva B: `P(Linea i d)`

Caso Base

```
(Vacio)  \equiv  indentar n (indentar m Vacio) = indentar (n+m) Vacio
         \equiv  indentar n Vacio = indentar (n+m) Vacio                {L1}
(*)      \equiv  Vacio = indentar (n+m) Vacio                           {L1}
         \equiv  Vacio = Vacio                                          {L1}
```

Observación (-): como `L1` vale `para todo k :: Int >= 0` entonces es una generalización frente al caso particular de `n+m`

> Queda demostrado el caso base `P(Vacio)` por definición

Partimos desde el lado izquierdo de la ecuación:

```
        indentar n (indentar m (Texto s d))
\equiv  indentar n (foldDoc Vacio f (g m))          {I}
\equiv  indentar n (f s (rec d))                    {FT}
\equiv  indentar n (f s (foldDoc Vacio f (g m) d))  {FREC}
\equiv  indentar n (f s (indentar m d))             {I}
\equiv  indentar n (Texto s (indentar m d))         {F}
\equiv  Texto s (indentar n (indentar m d))         {L2}
\equiv  Texto s (indentar (n+m) d)                  {HI}
\equiv  indentar (n+m) (Texto s d)                  {L2}
```

> Queda demostrado el paso inductivo A `P(d) => P(Texto s d) para todo s :: String. para todo d :: Doc`

Paso Inductivo B

`P(Linea i d)` $\equiv$ `indentar n (indentar m (Linea i d)) = indentar (n+m) (Linea i d) para todo n, m, i :: Int. d :: Doc`

Partimos desde el lado izquierdo de la ecuación:

```
        indentar n (indentar m (Linea i d))
\equiv  indentar n (foldDoc Vacio f (g m) (Linea i d))  {I}
\equiv  indentar n (g m i (rec d))                      {FL}
\equiv  indentar n (g m i (foldDoc Vacio f (g m) d))    {FREC}
\equiv  indentar n (g m i (indentar m d))               {I}
\equiv  indentar n (Linea (m + i) (indentar m d))       {G}
\equiv  Linea (n+m+i) (indentar (n+m) d)                {L3}
\equiv  Linea ((n+m)+i) (indentar (n+m) d)              {HI}
\equiv  indentar (n+m) (Linea i d)                      {L3}
```

> Queda demostrado el paso inductivo B `P(d) => P(Linea i d) para todo i :: Int >= 0. para todo d :: Doc`

Conclusión

Habiendo demostrado el **caso base** y los **pasos inductivos A y B**, queda demostrado por *inducción estructural* que vale `P(d) para todo d :: Doc`

```
        indentar k Vacio
\equiv  foldDoc Vacio f (g k) Vacio   {I}
\equiv  Vacio                         {FV}
```

> Demostramos que la igualdad *por definición*

Lema 2 (L2)

`indentar k (Texto s d) = Texto s (indentar k d) para todo k :: Int >= 0. s :: String. d :: Doc`

```
        indentar k (Texto s d)
\equiv  foldDoc Vacio f (g k) (Texto s d)   {I}
\equiv  f s (rec d)                         {FT}
\equiv  f s (foldDoc Vacio f (g k) d)       {FREC}
\equiv  f s (indentar k d)                  {I}
\equiv  Texto s (indentar k d)              {F}
```

> Demostramos que vale la igualdad por definición

Lema 3 (L3)

`indentar m (Linea k d) = Linea (m+k) (indentar m d) para todo m, k :: Int >= 0. d :: Doc`

```
        indentar m (Linea k d)
\equiv  foldDoc Vacio f (g m) (Linea k d)   {I}
\equiv  g m k (rec d)                       {FL}
\equiv  g m k (foldDoc Vacio f (g m) d)     {FREC}
\equiv  g m k (indentar m d)                {I}
\equiv Linea (m+k) (indentar m d)           {G}
```

> Demostramos que vale la igualdad por definición