ESTRUCTURAS DISCRETAS

Lógica Proposicional

Javier Enríquez Mendoza

Mauricio E. Hernández Olvera

SINÓNIMOS

A veces vamos a querer definir tipos de datos únicamente con los tipos primitivos de Haskell, sin agregarles nada.

Por ejemplo, si queremos definir los puntos en el plano en Haskell.

Anteriormente lo hicimos con tuplas de Float.

Entonces la firma de una función que recibe un punto, le hace algo y regresa otro punto, es la siguiente.

```
foo :: (Float,Float) -> (Float, Float)
```

Pero nuestros puntos siempre van a ser definidos igual (Float, Float)

SINÓNIMOS

Una forma de mejorar esto es crear un tipo de dato algebraico Punto, de la siguiente forma:

```
data Punto = Dot Float Float
```

Entonces la firma de la función ahora seria

```
foo :: Punto -> Punto
```

Claramente hubo una mejora. Pero al implementar la función tendremos que hacer pattern matching sobre el punto de entrada.

SINÓNIMOS

Tenemos entonces estas dos funciones:

```
foo :: (Float, Float) -> (Float,Float)
foo (x,y) = ( x+5, y+5 )

foo :: Punto -> Punto
foo (Dot x y) = (Dot (x+5) (y +5))
```

Parece que tenemos que elegir entre una firma verbosa o una definición poco legible.

SINÓNIMOS

Para solucionar esto, Haskell tiene sinónimos.

Un sinónimo es un tipo de dato definido únicamente con los tipos primitivos de Haskell.

Para definir un sinónimo usamos la palabra reservaba type.

Un claro ejemplo de esto es el tipo String, que esta definido como [Char].

Por ejemplo, la definición de String en Haskell es:

```
type String = [Char]
```

SINTAXIS

La forma de definir un sinónimo en Haskell es la siguiente

```
type Identificador = Definicion
```

Si seguimos con nuestro ejemplo de punto:

```
type Punto = (Float, Float)
```

Y entonces la funcion foo queda asi:

```
foo :: Punto -> Punto
foo (x,y) = (x+5, y+5)
```

ARGUMENTOS

Uno de los aspectos mas importantes de la lógica es decidir si un argumento es correcto o no.

Para responder esto, nos serviremos de la lógica matemática, que nos proporciona un conjunto de reglas operacionales que permiten derivar un nuevo hecho a partir de hechos dados.

Un argumento lógico es correcto si la verdad de sus premisas causa necesaria y obligatoriamente la verdad de su conclusión.

Esto puede mostrarse mediante las reglas lógicas de operación.

Para representar los argumentos usaremos Proposiciones.

PROPOSICIONES

De manera similar a como lo hacemos con expresiones aritméticas vamos a definir y construir las expresiones lógicas.

Empecemos por definir las expresiones elementales de la lógica.

Tenemos los valores constantes {Verdadero, Falso} conocidos como Booleanos.

A las expresiones lógicas se les conoce también como proposiciones y son enunciados a los que les podemos asociar un. valor lógico elemental (Verdadero o Falso).

Al igual que en las expresiones aritméticas vamos a tener un conjunto de variables.

Definición: Una proposición es un enunciado que puede calificarse como verdadero o falso, dependiendo del estado en que se evalúe.

ESTADO

El estado para evaluar una proposición es un conjunto de valores de las variables y constantes involucradas en la expresión.

Podemos definir un estado como un conjunto de parejas, donde cada pareja tiene un nombre de variable y su valor.

El estado nos va a servir para determinar si una proposición es verdadera o falsa.

Definicion: Un estado es una función que asigna una variable dada x a un valor v, elegidos de entre aquellos que puedan asignarse a esa variable.

OPERADORES LÓGICOS

	Sintaxis	**Significado**
Negación	$\neg P$	no P
Conjunción	$P \wedge Q$	P y Q
Disyunción	$P \vee Q$	P o Q
Implicación	$P \rightarrow Q$	P implica Q
Equivalencia	$P \leftrightarrow Q$	P si y sólo si Q

INTERPRETACIONES

Una interpretación de una proposición es una forma de evaluarla para saber que valor de verdad le corresponde según un estado dado.

Es una buena alternativa para evitar hacer la tabla de verdad de una proposición.

Cada estado posible de una proposición genera una función de interpretación sobre las formulas.

DEFINICIONES

Tautología:

Si $\mathcal{I}(P) = 1$ para toda interpretación $\mathcal{I}$, entonces P es una tautología.

Contradicción:

Si $\mathcal{I}(P) = 0$ para toda interpretación $\mathcal{I}$, entonces P es una contradicción.

Contingencia:

Si P no es tautología ni contradicción, entonces P es contingencia.

Argumento Correcto:

Un argumento es correcto, si y sólo si suponiendo que las premisas son verdaderas, entonces necesariamente la conclusión también lo es.