

Programación I - Segundo Cuatrimestre

Unidad 1: Propositiones y Predicados

Agosto 2021

Natalia Colussi

Licenciatura en Ciencias de la Computación
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario

1. Proposiciones
2. Predicados
3. Ejemplos
4. Traducciones con Predicados
5. Evaluación con Predicados

Proposiciones

- ¿Qué es una proposición?

Definición

Es una expresión que puede evaluarse a verdadero o falso.

- Ejemplos:
 - La palabra "HOLA" tiene una longitud de 4 caracteres
 - El número 2 es mayor que 0
 - La frase "Hola Mundo!" es la concatenación de "Hola" y " Mundo!".
 - 4 es número par.
 - El año 2000 fue bisiesto.
 - Los números 17 y 21 son primos.
- ¿Para que nos sirve una proposición?
- Vemos como escribir esto en Racket...

Proposiciones

```
1 > (= (string-length "Hola") 4)
2 > ( > 2 0)
3 > (string-append "Hola" " Mundo!")
4 > ( = (modulo 4 2) 0)
5 > (and ( = (modulo 2000 4 ) 0) ; divisible por 4
6       ( or (not (= (modulo 2000 100) 0)) ; no ←
7           divisible por 100
8           (= (modulo 2000 400) 0))) ; divisible ←
9       por 400
>( and (17 ?
      (21 ? )))
```

En este último caso sería bueno tener alguna función que chequeará si el número es primo o no y de esta forma nos permitiera escribir la siguiente expresión:

```
1 >( and (es-Primo? 17)
2       (es-Primo? 21))
```

Proposiciones

- De igual modo con el año bisiesto:

```
1 > (and ( = (modulo 2000 4 ) 0) ; divisible por 4
2       ( or (not (= (modulo 2000 100) 0)) ; no ←
3           divisible por 100
              (= (modulo 2000 400) 0))) ; divisible ←
              por 400
```

- Nos sería útil contar con una función que nos permitiera determinar la verdad o falsedad para el año 2000, pero también para los años 2100 y 1900.

```
1 >(define (es-Bisiesto? año)
2 >(and ( = (modulo año 4 ) 0) ; divisible por 4
3     ( or (not (= (modulo año 100) 0)) ; no ←
4         divisible por 100
              (= (modulo año 400) 0))) ; divisible ←
              por 400
```

Predicados

- ¿Qué es un predicado?

Definición

Es una función cuyo cuerpo es una expresión booleana, donde los argumentos pueden ser de cualquier tipo, pero el resultado siempre será un valor booleano. El resultado, de la invocación al predicado, siempre será verdadero o falso.

- La signatura de un predicado será:

predicado: ? \rightarrow Boolean

- ¿Cómo escribo en Racket un predicado?

Un predicado es una FUNCIÓN! No hay problema!

Predicados

```
1 >(define (es-Bisiesto? año)
2 >      (and ( = (modulo año 4 ) 0) ; divisible ↵
3           por 4
4           ( or (not (= (modulo año 100) 0)) ; ↵
                no divisible por 100
                (= (modulo año 400) 0))) ; ↵
                divisible por 400
```

- ¿Cómo usamos un predicado?

```
1 >(es-Bisiesto? 2000) ; ve: #true
2 >(es-Bisiesto? 2100) ; ve: #false
3 >(es-Bisiesto? 1900) ; ve: #false
```

Predicados

- ¿Para qué usamos un predicado?
 - ✓ Los predicados nos permitirán tomar decisiones, en función de si se satisfacen o no.
 - ✓ Manejar comportamientos según se satisfagan o no.
 - ✓ Extender funcionalidades.
 - ✓ Prevenir errores o comportamientos inesperados de la funciones.
- Predicados en Racket. Los mismos pueden clasificarse:
 - ✓ Predicados Propios del Lenguaje (Built-in)
 - ✓ Predicados Definidos por el usuario (User-defined)
- Convenciones en Racket para identificar rápidamente un predicado.
 1. Anteponemos “es” (“is”) al nombre.
 2. Ponemos como postfijo el signo de pregunta. ?
- Vamos a realizar algunos ejemplos en Racket

Ejemplos

Ejemplos Predicados (Built-in)

- Más usados de Racket (números)

- ✓ `even?` (es par)
- ✓ `odd?` (es impar)
- ✓ `zero?` (es cero)
- ✓ `positive?` (es > 0)
- ✓ `negative?` (es < 0)

- Más usados de Racket (string)

- ✓ `string?`
- ✓ `string-upper-case?`
- ✓ `string-lower-case?`
- ✓ `string-numeric?`
- ✓ `string-alphabetic?`

Ejemplos Predicados (Built-in)

- Más predicados numéricos ...
 - ✓ `real?`
 - ✓ `number?`
 - ✓ `exact?`
 - ✓ `inexact?`
 - ✓ `integer?`
- Otros predicados sobre Datos Primitivos
 - ✓ `image?`
 - ✓ `string?`
 - ✓ `number?`
 - ✓ `boolean?`

Ejemplos

- Ejemplo 1:

Definimos un predicado que chequea si la longitud de una palabra es mayor o igual que 4.

```
1 ;es-str-4?: String -> Boolean
2 ;-----
3 ( define (es-str-4? str)
4           (>= (string-length str ) 4) )
5 ;-----
6 (es-str-4? "Hola") ; ve: #t
7 (es-str-4? "Abracadabra") ; ve: #t
8 (es-str-4? "Hoy") ; ve: #f
9 (es-str-4? "M"); ve: #f
10 (es-str-4? ""); ve: #f
```

- Vamos por algunos ejemplos más...

- Ejemplo 2:

*Definimos un predicado que determina si la longitud de una string **str** es mayor o igual que una cota **inferior** y menor estricta que una cota **superior**.*

- Ejemplo 3:

*Definimos un predicado que determina si una string **str** comienza con una vocal.*

- Ejemplo 4:

*Definimos un predicado que determina si una string **str** no comienza con una vocal.*

Ejemplos

- Ejemplos varios para trabajar en 7 min!

Definimos un predicado que determina si dada la edad de una persona ésta puede:

1. *Votar en la Provincia de Santa Fe.*
2. *Votar a nivel Nacional.*
3. *Manejar un automovil.*
4. *Consumir alcohol.*
5. *Contraer matrimonio.*
6. *Emanciparse.*
7. *Recibir una herencia.*
8. *Ser una adolescente.*
9. *Ser un niño.*
10. *Ser un adulto.*
11. *Ser un adulto mayor.*
12. *Ser un estudiante del ciclo primario.*

Traducciones con Predicados

Traducciones

- De igual modo a como hicimos las traducciones con las proposiciones, haremos las traducciones que involucren el uso de predicados.
- Traducciones podemos usar predicados existentes (built-in) y definir nuevos.
- Ejemplos:
 - ✓ "La cadena "Abracadabra" tiene una longitud mayor que 4 y comienza con una vocal"

```
1      (define CADENA "Abracadabra")
2      (and (es-str-4? CADENA)
3           (comienza-vocal? CADENA))
```

- ✓ "1900 no fue un año bisiesto"

```
1      (not (es-bisiesto? 1900))
```

- Siempre es mejor dividir la traducción en partes, analizar qué predicados existen en Racket, cuáles son los que tengo que definir, determinar conectores que unen las condiciones, etc.
- Vamos a trabajar con ejemplos en el Jamboard!

Evaluación con Predicados

Evaluación de Predicados

```
(even? (* (+ (* 2 1 3) (- 1 1)) (cos 0)))  
= {Def. × }  
  (even? (* (+ 6 (- 1 1)) (cos 0)))  
= {Def. - }  
  (even? (* (+ 6 0) (cos 0)))  
= {Def. + }  
  (even? (* 6 (cos 0)))  
= {Def. cos }  
  (even? (* 6 1))  
= {Def. × }  
  (even? 6 )  
= {Def. even? }  
  #true.
```



¿Preguntas?

- Pueden dejar las consultas en el foro del capítulo, aclaren qué son sobre cuestiones teóricas.
- Pueden consultar también sobre sus dudas en los horarios de clases teóricos síncronos.