Standard Meta Language

Lenguajes de Programación, Tarea Corta 7

B. Mena, A. Leandro, W. González, D. Delgado

24 de octubre, 2017

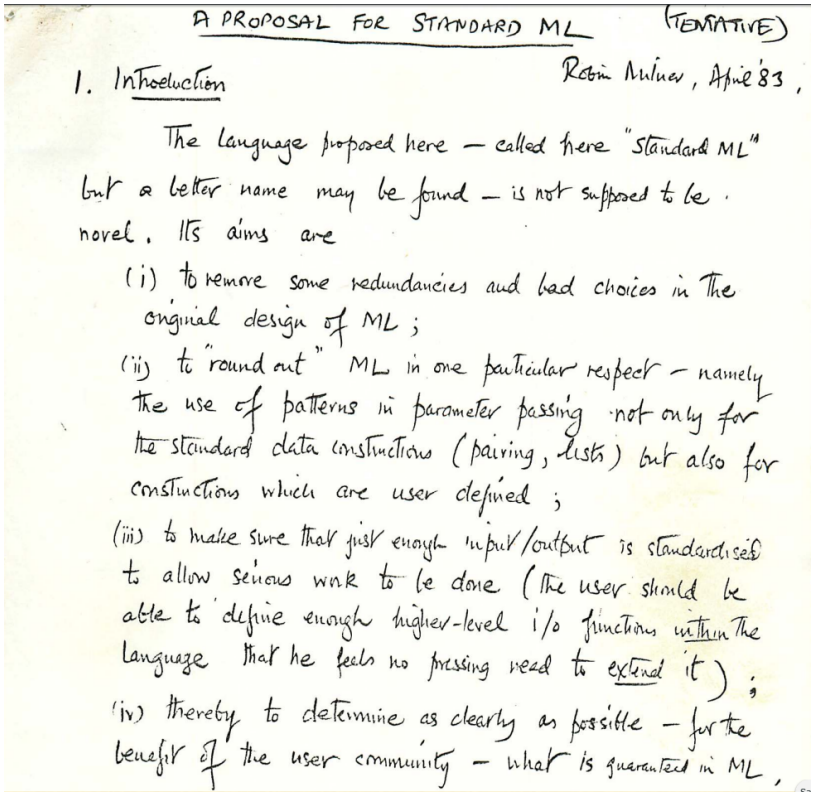
TEC | Tecnológico de Costa Rica

Tabla de contenidos

TEC | Tecnológico de Costa Rica

Datos Históricos

- Lenguaje de propósito general, modular.
- Funcional con comprobación de tipo en tiempo de compilación.
- Demostración automática de teoremas.
- Popular entre escritores de compiladores e investigadores.
- Primeras reuniones : 1983, 1984, 1985.
- Reuniones desde 1993 hasta el 2000. Sin lograr un consenso, debido a propiedades de POO.



A PROPOSAL FOR STANDARD ML (TENTATIVE)

1. Introduction — Robin Milner, April '83,

The language proposed here — called here "Standard ML" but a better name may be found — is not supposed to be novel. Its aims are

(i) to remove some redundancies and bad choices in the original design of ML;

(ii) to "round out" ML in one particular respect — namely the use of patterns in parameter passing not only for the standard data constructions (pairing, lists) but also for constructions which are user defined;

(iii) to make sure that just enough input/output is standardised to allow serious work to be done (the user should be able to define enough higher-level i/o functions within the language that he feels no pressing need to extend it);

(iv) thereby to determine as clearly as possible — for the benefit of the user community — what is guaranteed in ML,

FIGURE – Borrador de Robin Milner (sml-family.org)

TEC | Tecnológico de Costa Rica

Convenciones

Paréntesis

(x + y) ✓

(x + y) ✖

Operadores

x + y ✓

x+y ✖

Espacios

[x, y, z] ✓

[x,y,z] ✖
[x, y, z] ✖

Identación

```
if b1 then e1 else e2

if b1 then e1
else if b2 then e2
else e3

if b1 then
    e1
b2 then
    e2
else
    e3
```

Identificadores		
Tipo	Ejemplo	Uso
Simbólico	** ++	Operaciones especiales
Minúscula	x y z	Valores locales
Todo en minúscula	foo_bar	Valores locales
Mayúscula	FooBar	Constructores de tipos, de excepciones, estructuras, functors
Todo en mayúscula	FOO_BAR	Tipo de una estructura

TEC | Tecnológico de Costa Rica

Tipos de datos

- bool
- string
- char
- int
- real
- unit

string

"Hola"
"Hola" ^ "Mundo !" = "Hola Mundo !"

char

"a"

int

255 = 0xff

TEC | Tecnológico de Costa Rica

Tipos de datos

- Estándar de la revisión en 1997.
- Interfaces, operaciones básicas.
- Soporte I/O y para tipos de datos estándar.
- *Options* y listas.
- Constantes matemáticas.
- Valores positivos o negativos infinitos.
- Representación de tiempo.
- Conversiones entre valores numéricos.
- Búsquedas en cadenas de *char*.
- Operaciones sobre *strings*.

TEC | Tecnológico de Costa Rica

Tipos de datos

Options

Cuando es posible que algo no tenga un valor válido.

Ejemplo options

```
fun dividir x y = if y == 0
then NONE else SOME (x / y)
```

Listas

- Recursivo y polimórfico.
- Head y tail.
- Indexado.

Ejemplo listas

```
datatype 'a list = nil | : : of 'a * 'a list

val lst = [1,2,3,4] : int list
```

TEC | Tecnológico de Costa Rica

Tipos de datos

Tuplas

- Conjunto ordenado de valores.
- (1, 2) es de tipo int * int
- ('foo', false) es de tipo string * bool
- () es una 0-tupla o *unit*.
- Pueden anidarse.

Ejemplo tuplas

val alien1 = ("Groot" , 1586 , 3.14159)

val gustos = [("Gamora" , "helados") , ("Drax" , "perritos ") , ("Rocket" , "cerezas")]

val articulos = (["helado" , "perritos" , "chocolate"] , ["higado" , " alquiler "])

TEC | Tecnológico de Costa Rica

Estructuras básicas de datos

- Una función acepta un valor y normalmente devuelve otro valor.
- Tipo de retorno estático en compilación.

Ejemplo tipo implícito

```
fun factorial n = if n <1 then 1 else n * factorial (n - 1)
```

Ejemplo tipo explícito

```
fun factorial (n :int) : int = if n <1 then 1 else n * factorial (n - 1)
```

TEC | Tecnológico de Costa Rica

Funciones

- **Comprobación de tipos**, comparaciíon de argumento actual con argumento formal.
- **Funciones sin valores de retorno o argumentos**, utiliza el tipo *unit*.

TEC | Tecnológico de Costa Rica

Funciones recursivas

- Debe tener un nombre con el cual hará referencia a la misma.
- Se utiliza un binding de valor recursivo : **rec**.

Ejemplo factorial recursivo

val rec factorial : int->int = **fn** 0 => 1 | n :int => n * factorial (n - 1)

Ejemplo factorial recursivo con FUN

fun factorial 0 = 1 | factorial n :int = n * factorial (n - 1)

Ejemplo funcionaes mutuamente recursivas

fun par 0 = true | par n = impar(n-1) **and** impar 0 = false | impar n = par(n-1)

TEC | Tecnológico de Costa Rica

Funciones

if

if boolExpr **then** expr1 **else** expr2

case

case boolExpr **of** true -> expr1 | false -> expr2

Secuenciadores

- Escritas separadas por punto y coma.
- Útiles para expresiones de efectos secundarios.

Ejemplo secuencias de expresiones

- **val** x = (print "Hola\n" ; 7)

Output :
Hola
val x = 7 : int

TEC | Tecnológico de Costa Rica

Funciones

Ejemplo polimorfismo 1

```
fun swapInt(x : int, y : int) : int*int = (y,x)

fun swapReal(x : real, y : real) : real*real = (y,x)

fun swapString(x : string, y : string) : string*string = (y,x)
```

Ejemplo polimorfismo 2

```
fun swapIntReal(x : int, y : real) : real*int = (y,x)

fun swapRealInt(x : real, y : int) : int*real = (y,x)
```

Ejemplo polimorfismo 3

```
fun swap(x : 'a, y : 'b) : 'b*'a = (y,x)

val swap = fn : 'a*'b -> 'b*'a
```

- Una variable de tipo es cualquier identificador precedido por una comilla simple.
- El compilador no permite realizar ninguna operación en *x* que no sea un tipo arbitrario.

TEC | Tecnológico de Costa Rica

Características

- *Strong typing*.
- No tiene subtipos.
- Cada expresión tiene un tipo específico en compilación.
- Tipo en ejecución nunca se contrapone.
- Conversiones explícitas.
- Un programa mal tipificado ni siquiera compilará.

TEC | Tecnológico de Costa Rica

Ventajas

- Permite definir tipos de datos abstractos.
- Programadores consideran el tipado un aspecto positivo.
- Muchos errores de programación son capturados en tiempo de compilación.
- SML/NJ permite construir y manejar estructuras de datos de C.
- SML/NJ puede cargar bibliotecas de C.
- SML/NJ puede administrar memoria de estructuras de datos de C.

TEC | Tecnológico de Costa Rica

Desventajas

- Si el programador se encuentra acostumbrado a lenguajes con tipado distinto, SML puede significar un mayor tiempo en el aprendizaje.
- Los tipos, en comparación con otros lenguajes, son más complejos de comprender.

TEC | Tecnológico de Costa Rica