

ISC-315-T. Proyecto de soporte de lenguaje para la manipulación de un modelo de una investigación.

Javier Falcón 20165265, Manuel Molina 20165468
y Natalia Vallejo 20165328

1 Descripción detallada del posible entorno de ejecución del intérprete

Para el concebimiento de este lenguaje, hemos decidido la realización de un *internal DSL*, tomando como anfitrión el lenguaje **R**. Esto permitirá que el usuario final pueda trabajar en un entorno de desarrollo de **R**, específicamente en su consola, la cual es un entorno ya conocido para él. Esto es posible debido a que nuestro lenguaje estará *embedded* en **R**.

2 Selección y fundamentación del paradigma de programación

Consideramos que nuestro lenguaje debe ser multiparadigma, tomando principalmente los paradigmas funcional y orientado a objetos. Esto es debido a dos cosas. Primero, si queremos tomar a **R** como nuestro lenguaje anfitrión, debería corresponderse con sus paradigmas base. La segunda razón viene en relación a que no se quiere solamente realizar cálculos con la data experimental, sino que también deben ofrecerse mecanismos de ajustes de los modelos, por lo que consideramos que estos paradigmas solucionan ambas cosas.

3 Especificación detallada de los objetivos del lenguaje propuesto

Con nuestro lenguaje buscamos otorgar mecanismos para la manipulación y ajustes de modelos de investigación, tomando en cuenta las entradas y salidas de la data, mecanismos de consolidación, ejecución del modelo a partir de la data experimental y producción de gráficos de entradas y salidas.

4 Especificación sintáctica del lenguaje propuesto

Basándonos en que el perfil del usuario describe una familiaridad con el lenguaje R, tomamos las siguientes decisiones de diseño sobre la sintáctica del lenguaje.

Caracteres o secuencia de caracteres	Qué representan
< –	Asignación
– >	Acceso a funciones que modifican el modelo
#	Comentario
Alfanuméricos, ., -	Identificadores de variables de cualquier tipo ₁
list	Especificación de una lista
r.	Comienzo de una sección de código en R
load	Se va a cargar data de un archivo
model	El comienzo de la especificación de un modelo
external	Variable externa de un modelo
var	Variable de un modelo
func	Función de un modelo
params	Parámetros de un modelo o función
modparam	Modifica un parámetro
setfunction	Añade una función a un modelo
output	Salida de una función
execute	Ejecuta las funciones especificadas en el modelo, dada la entrada y almacena en la salida
[Comienzo al acceso de un atributo del modelo
]	Fin del acceso de un atributo del modelo
“	Comienzo de una constante de cadena
”	Fin de una constante de cadena
print	Impresión en consola
(Comienzo de especificación de parámetros de una función
)	Fin de especificación de parámetros de una función
graph	Construcción de un grafo

¹ Los identificadores deben empezar en una letra, luego pueden contener cualquier secuencia de los caracteres referenciados. Las letras pueden ser tanto mayúsculas como minúsculas.

Cabe destacar que nuestro lenguaje es case sensitive, lo cual heredamos de R. Cada palabra reservada debe estar escrita completa en minúsculas. Por igual decidimos que estuviesen en inglés, no solo para que el lenguaje pueda ser utilizado internacionalmente (y, porque creemos que un usuario de este perfil debe manejar el idioma), sino porque sería contrastante cambiar a palabras reservadas en inglés cada vez que se esté haciendo referencia a alguna función de R.

5 Especificación semántica del lenguaje propuesto

- Tipado dinámico: nuestro lenguaje opta por el uso del tipado dinámico, es decir, el chequeo semántico de tipos se realiza en tiempos de ejecución.
- Los fragmentos de código de R no se incluirán en los análisis semánticos de nuestro lenguaje. De esto se encargará el intérprete de R una vez realizada la traducción.
- La data podrá ser cargada desde un fichero.
- La especificación del tipo de gráfico a generar debe concordar con los que **R** ofrece.
- Los parámetros de los modelos son modificables, con el objetivo de poder ajustarlos.
- La salida producida por la ejecución del modelo se dará en forma de lista.
- La(s) función(es) del modelo debe estar previamente implementada en **R**. Posteriormente se podrá indicar cuáles parámetros del modelo utilizará y en qué sitio guardará su salida.

6 Indicaciones generales del procedimiento de realización del intérprete para el lenguaje propuesto

Considerando que nuestro lenguaje puede desarrollarse en el entorno de ejecución de R, primero debe realizarse el proceso de interpretación propio, que engloba chequeo sintáctico y chequeo semántico. En vista del embebimiento de nuestro lenguaje, el resultado del parsing del lenguaje será un árbol de parsing mixto, con sentencias de R y sentencias de nuestro lenguaje. La realización del intérprete debería preocuparse por realizar el proceso de parsing y, a través de acciones semánticas, realizar la traducción de nuestro lenguaje a las sentencias de R equivalentes, de tal manera que pueda ser ejecutado posteriormente por el intérprete de R.

7 Ejemplo de una sesión de trabajo con el lenguaje

```

#This is a comment! Agregar comentarios + listas

f <- r.function(p1, p2) {
    #statements
}

x <- load(archivo.txt)

myModel <- model(

    external(x),
    var(a, b, c),
    func(f, y),
    params (j,k,l)
)

myModel -> modparam(j, 5)

myModel -> setFunction(
    name(f),
    params(j, k),
    output(a)
)

execute(myModel[f])

print (myModel[a])

graph("plot", x, c)

```

References

- [1] René Visser Eelco Bravenboer, Martin de Groot. *MetaBorg in Action: Examples of Domain-specific Language Embedding and Assimilation using Stratego/XT*. Department Information and Computing Sciences Utrecht University, 2006.