Un Modelo de Hoja de Cálculo: Observaciones a Implementaciones

Emely Arráiz, Roger Soler, Cristina Zoltan

Universidad Simón Bolívar, Apartado 89000, Caracas , VENEZUELA

Resumen

Se presenta la especificación semántica del lenguaje subyacente en una familia de aplicaciones conocidas como "Hoja de Cálculo", con el propósito de realizar un análisis de esta familia.

Palabras Clave: Hoja de Cálculo, Especificación denotacional, Análisis de programas

1 Introducción

La Semántica Denotacional, algunas veces denominada Semántica Matemática, otras Semántica Funcional es un método desarrollado con el objetivo de describir los conceptos que aparecen en los lenguajes de programación. Este método consiste en asociar a cada constructor sintáctico del lenguaje de programación, a través de funciones de evaluación semántica, valores abstractos u objetos matemáticos. Esta asociación está dirigida por la morfología del lenguaje de programación. La validez del método está fundamentada en lo que se conoce como Teoría de Dominios, [3],[1],[2]. El propósito de este trabajo es usar el método, como lenguaje de especificación, en el proceso de análisis y prueba de aplicaciones.

2 El problema

Una Hoja de Cálculo puede ser vista como un sistema de definición de funciones:

$$y_i = E_i \tag{1}$$

donde los E_i son expresiones con variables acotadas y bien construidas en algún álgebra y y_i son los identificadores de las expresiones (funciones). Por ejemplo:

$$\begin{cases} y_1 &= x_1 * x_2 + x_3 \\ y_2 &= x_3 + 7 + x_1 \\ y_3 &= 121 \end{cases}$$

en este caso podemos estar en el álgebra de los enteros $\mathcal{Z}$, con las operaciones de multiplicación (*) y suma (+) , 7 y 121 constantes y las $x_i, i = 1, 2, 3$ representan variables acotadas. Para los y_i se tiene que son funciones definidas en $y_i : \mathcal{Z}^n \rightarrow \mathcal{Z}$. Se puede presentar (1) como:

$$z_i \quad = y_i(w_i) \tag{2}$$

Continuando el ejemplo anterior, tenemos que los w_i son identificadores de tripletas de valores en $\mathcal{Z}^3$ y los z_i son identificadores de valores en $\mathcal{Z}$ (cada z_i estará asociado al valor de la evaluación de la función y_i en w_i). Si tomamos $w_1 = w_2 = w_3 =< 3, 4, 5 >$ entonces, (2) se reescribirá como:

$$\begin{cases} z_1 & = y_1(< 3, 4, 5 >) = 3 * 4 + 5 = 17 \\ z_2 & = y_2(< 3, 4, 5 >) = 5 + 7 + 3 = 15 \\ z_3 & = y_3(< 3, 4, 5 >) = 121 \end{cases}$$

En el caso de la Hoja de Cálculo, los identificadores de funciones y_i, los identificadores de los argumentos w_i y los identificadores de los resultados de evaluación z_i son "sintácticamente" los mismos que los identificadores de las variables acotadas x_i . Es decir tenemos:

$$\begin{aligned} w_1 & \quad = w_2 = \ldots = w_n = w \\ z & \quad \equiv y \equiv w \equiv x \end{aligned} \tag{3}$$

Usando (3), la evaluación del sistema de funciones presentadas en (1) tiene la siguiente forma:

$$x_i \quad = x_i(x_1, x_2, \ldots, x_n) \tag{4}$$

De esta manera cada identificador x_i tiene asociado una función y un valor que es el resultado de evaluar la función que tiene asociada; además las variables acotadas en las funciones son las mismas x_i.

El sistema de funciones corresponde a las variables de la Hoja de Cálculo y el sistema hoja de cálculo estará constituido por un ambiente y una presentación.

El ambiente fija el tipo de cada uno de los identificadores de la hoja y la secuencia de evaluación del sistema constituido por estas.

El sistema planteado dará resultados diferentes dependiendo de la secuencia de evaluación del mismo, por ejemplo:

- para la secuencia de cálculo: x_3, x_2, x_1 , $x = < 3, 4, 5 >$; se obtiene $x = < 512, 131, 121 >$.

- para la secuencia de cálculo: en paralelo $x_1 \parallel x_2 \parallel x_3$, $x = < 3, 4, 5 >$; se obtiene $x = < 17, 15, 121 >$.

Como los identificadores están asociados a valores y funciones, podemos expresar las funciones con argumentos implícitos o explícitos, obteniendose tres tipos de asociaciones para los identificadores. Si agregamos al sistema del ejemplo esta cuarta definición:

$$x_4 = x_1 + x_1() + x_1(x_1 : 6, x_2 : 7, x_3 : 8) \tag{5}$$

tenemos, para la secuencia de cálculo x_1, x_2, x_3, x_4 y $x = < 3, 4, 5, 0 >$ se obtiene $x = < 17, 15, 121, 392 >$.

3 Descripción Denotacional del Lenguaje

A continuación se propone un lenguaje para soportar algunas de las propiedades mencionadas en el párrafo anterior. Primero se presenta una descripción de la parte sintáctica del lenguaje (3.1) y luego una descripción de la semántica (3.2 y 3.3).

3.1 Dominios Sintácticos

El lenguaje está compuesto por las clases sintácticas siguientes:

- Regiones $\mathcal{REG}$ ($\Upsilon \in \mathcal{REG}$) ($\Delta \in \mathcal{VAR} \subseteq \mathcal{REG}$)
- Ordenamientos $\mathcal{ORD}$ ($\Theta \in \mathcal{ORD}$)
- Expresiones $\mathcal{EXP}$ ($E \in \mathcal{EXP}$)
- Comandos $\mathcal{CMD}$ ($\Gamma \in \mathcal{CMD}$)

Los identificadores Υ,Δ, Θ ., E y Γ representan elementos genéricos de los dominios de regiones,variables, ordenamientos, expresiones, y comandos respectivamente.

3.1.1 Dominio de Regiones ($\Upsilon \in \mathcal{REG}$)

Las regiones son subconjuntos de variables, y los elementos de este dominio (Υ) son identificadores que incluyen a los identificadores de variables (Δ) que son regiones unitarias. Los nombres de variables están pre-determinados y son infinitas. El dominio $\mathcal{VAR}$ posee un predicado de igualdad que establece cuando dos identificadores de variables son sintácticamente iguales. Este dominio tiene una organización bidimensional, es decir $\mathcal{VAR} = \mathcal{VAR}_1 \times \mathcal{VAR}_2$ y existe un orden total $(\mathcal{VAR}_1, \leq_1), (\mathcal{VAR}_2, \leq_2)$ y operaciones de sucesor y predecesor sobre $\mathcal{VAR}_1$ *y* $\mathcal{VAR}_2$

La morfología de los identificadores de variables está dada por:

$\Delta \longrightarrow < \delta_c, \delta_f >$ denotando columna fila

δ_c es el identificador de las columnas y δ_f el identificador de las filas. En el texto usaremos Δ ó $< \delta_f, \delta_c >$ cuando se requiera. La morfología de los identificadores de región no-unitaria es irrelevante.

3.1.2 Dominio de Ordenamientos ($\Theta \in \mathcal{ORD}$)

En este dominio los identificadores corresponden a las secuencias de evaluación que se utilizarán durante la ejecución de la hoja de cálculo. Las secuencias que se considerarán serán solamente las secuencias finitas ya que corresponden a computaciones finitas.

Al igual que los identificadores de región, su morfología es irrelevante. Sin embargo tendremos elementos distinguidos correspondientes a evaluaciones usuales. Podemos destacar Θ_0 como el identificador de la secuencia actual durante una evaluación; Θ_1 como el identificador de la secuencia de dependencia funcional; Θ_2 como el identificador de la secuencia considerando las variables por filas; Θ_3 como el identificador de la secuencia considerando las variables por columnas; Θ_4 como el identificador de la secuencia por orden de definición de variables.

3.1.3 Dominio de las Expresiones ($\mathcal{EXP}$)

Los elementos de éste dominio son las expresiones que se le asocian a los identificadores de variables y tendrán que ser evaluadas. Pueden ser: expresiones primitivas asociadas a un álgebra y de entrada y salida. Algunos ejemplos:

$E_P \longrightarrow$	$\nu \mid \Delta$	Valor (ver 3.2)
	$\mid E_{P1} + E_{P2} \mid E_{P1} * E_{P2} \mid$	Expresiones aritméticas
	$\mid \mathrm{FUN}(E_{P1}, E_{P2}, \ldots, E_{Pn})$	Funciones predefinidas
$E_E \longrightarrow$	$entrada(\Upsilon)$	Expresiones de entrada
$E_S \longrightarrow$	$E_P; salida(\Upsilon)$	Expresiones de salida

E_P son las expresiones que involucran variables de la Hoja de Cálculo (en sus diversas presentaciones), mientras que E_E y E_S corresponden a expresiones que involucran regiones de la hoja.

3.1.4 Dominio de los Comandos ($\mathcal{CMD}$)

Los comandos se pueden agrupar en comandos de Regiones, Ambientes y de Entrada y Salida. Los de Región permiten definir subconjuntos de variables. Los de Ambientes son aquellos que transforman el ambiente modificando la secuencia de evaluación y permitiendo la manipulación de los elementos asociados a las variables tal como el editor de expresiones.

Los comandos que se presentan dan algunas facilidades de las Hojas de Cálculo estudiadas, las extienden y generalizan. Como ejemplos, los casos siguientes:

$\Gamma_R \longrightarrow$	DefColec($\Delta_1, \ldots, \Delta_n, \Upsilon$)	Definir región con variables
	| DefPred(Δ, Υ)	Definir región con predecesores funcionales de Δ
$\Gamma_A \longrightarrow$	movAbs(Υ_1, Υ_2)	Mover región en forma absoluta
	| Calcular(o)	Evalúa la sec. en el contexto dado

Hasta este punto, se ha establecido los dominios semánticos y su morfología y se ha dado una descripción informal del significado. En los dos puntos siguientes, se expresará de manera formal el significado de las expresiones y de los comandos.

3.2 Dominios Semánticos

Es aquí donde se definen los valores u objetos matemáticos que serán asociados a los elementos sintácticos del lenguaje (el universo de discurso semántico). Los dominios semánticos que se necesitan para interpretar el lenguaje son los siguientes:

- Valores $\mathcal{VAL}$ ($\nu \in \mathcal{VAL}$)
- Tipos $\mathcal{TIP}$ ($\tau \in \mathcal{TIP}$)
- Ambientes $\mathcal{AMB}$ ($\alpha \in \mathcal{AMB}$)

3.2.1 Valores($\mathcal{VAL}$)

El dominio $\mathcal{VAL}$ está constituido por los valores ó constante de los tipos : lógico (B) , reales ($\Re$), enteros ($\mathcal{Z}$), cadenas finitas del alfabeto (Σ^*) y un valor nulo.
Es decir:

$$\nu \in \mathcal{VAL} = \mathcal{B} \oplus \Re \oplus \mathcal{Z}; \oplus \Sigma^* \oplus nula \tag{6}$$

3.2.2 Tipos ($\mathcal{TIP}$)

El dominio de tipos corresponde a los identificadores de tipos asignables a las variables. Este conjunto es fijo (tipo predefinidos) o puede ser ampliado por los usuarios. En nuestro caso lo trataremos como un conjunto extensible o no. El elemento genérico de este conjunto lo denominaremos τ.

3.2.3 Ambientes($\mathcal{AMB}$)

Es una recopilación de los contextos de variables, región y evaluación definido de la forma siguiente:

$$\mathcal{AMB} \; = \; \mathcal{CTXV} \times \mathcal{CTXR} \times \mathcal{CTXE}$$

$$\alpha \; = < \varphi \, , \varphi_r \, , \varphi_o >$$

Donde en el Contexto de Variables ($\mathcal{CTXV}$) se reflejan las posibles asociaciones entre las variables y la tripleta constituida por <valor , tipo , expresión>

$$\mathcal{CTXV} = [\mathcal{VAR} \rightarrow \mathcal{VAL} \times \mathcal{TIP} \times \mathcal{EXP}]$$

$$\varphi \; \in \mathcal{CTXV} \quad \leftrightarrow \quad \varphi = < \varphi_v, \varphi_t, \varphi_e >$$

En el Contexto de Regiones($\mathcal{CTXR}$) le asociamos a una región, un conjunto de variables y su secuencia de presentación

$$\mathcal{CTXR} \; = \; [\mathcal{REG} \rightarrow \mathcal{P}(\mathcal{VAR}) \times \mathcal{VAR}^*]$$

$$\varphi_r \; \in \mathcal{CTXR} \quad \leftrightarrow \quad \varphi_r = < \varphi_{r_1}, \varphi_{r_2} >$$

Un elemento distinguido del dominio de regiones es Υ_0 que corresponde a la región definida por las variables que se visualizan en la pantalla. La función, φ_r no podrá ser sobreyectiva ya que $\mathcal{P}(\mathcal{VAR})$ tiene la potencia del continuo mientras que los identificadores son numerables.

El Contexto de Evaluación($\mathcal{CTXE}$) es la función que asocia a cada identificador $\mathcal{ORD}$ la secuencia que indica el orden en que serán evaluadas las variables y el modo representando, el modo manual (m) ó modo automático (a)

$$\varphi_o \; \in \mathcal{CTXE} \; = \; [\mathcal{ORD} \rightarrow (\mathcal{VAR}^* \times \{m, a\})]$$

3.3 Funciones de Interpretación

Ahora se definirá en forma precisa lo que sucede cuando se evalua una expresión y se ejecuta un comando. Las funciones de interpretación son:

$$\mathcal{E} \; : \; \mathcal{EXP} \longrightarrow (\mathcal{AMB} \longrightarrow (\mathcal{VAL} \; \times \; \mathcal{VAR}^*))$$

$$\mathcal{I} \; : \; \mathcal{CMD} \longrightarrow (\mathcal{AMB} \longrightarrow \mathcal{AMB})$$

es decir dada una expresión, su interpretación será una función del ambiente a los valores y a los órdenes de evaluación. Dado un comando obtenemos una función de ambientes en ambientes (i.e: la interpretación de comandos es una transformación de ambientes).

La definición de la funciones de interpretación se realiza por casos, utilizando la sintaxis abstracta (3.1). También utilizaremos la notación lambda (λ) para expresar las funciones.

Por razones de espacio solo se dá una muestra de las funciones de interpretación de expresiones y comandos. Una versión ampliada se encuentra en [4].

3.3.1 La función de interpretación ($\mathcal{E}$) de Expresiones

Si se denota como α la colección de los ambientes

$$\alpha = << \varphi_v, \varphi_e, \varphi_t >, \varphi_o, < \varphi_{r_1}, \varphi_{r_2} >>$$

siguiendo las definiciones de las expresiones (3.1.3) resulta

$$\mathcal{E}[\![\nu]\!] = \lambda\alpha \cdot < \nu, \epsilon >$$

$$\mathcal{E}[\![\Delta]\!] = \lambda\alpha \cdot < \varphi_v(\Delta), \epsilon >$$

Esta evaluación corresponde a considerar la expresión asociada a una variable como una función. Los valores asociados a los elementos que aparecen en la expresión al momento de realizar la evaluación son sus parámetros actuales.

3.3.2 Funciones de interpretación ($\mathcal{I}$) de Comandos

A continuación daremos algunos de los comandos del lenguaje. Recordemos que la sintáxis de $\mathcal{I}$ es:

$$\mathcal{I} : \mathcal{CMD} \longrightarrow (\mathcal{AMB} \longrightarrow \mathcal{AMB})$$

y un ambiente α está dado por:

$$\alpha = << \varphi_v, \varphi_e, \varphi_t >, \varphi_o, < \varphi_{r_1}, \varphi_{r_2} >>$$

$$\mathcal{I}[\![\text{ DefColec}(\Delta_1, \ldots, \Delta_n, \Upsilon)]\!] = \lambda\alpha \cdot < \alpha \downarrow 1, \alpha \downarrow 2, < \varphi_{r_1}[\Upsilon|\Delta_1, \ldots, \Delta_n], \varphi_{r_2}[\Upsilon|\text{ sec}] >>$$

$$\mathcal{I}[\![\text{ calcular}(\Theta)]\!] = \lambda\alpha \cdot \text{ evaluar} < \alpha \downarrow 1, \varphi_o[\Theta_o|\text{actual}, \alpha \downarrow 3 >$$

Donde: actual es la secuencia de evaluación que obtenemos a partir de $\varphi_o(\Theta)$, haciendo substituciones "convenientemente " [1] a las variables que tenían asociadas expresiones de

[1] Los modos de sustitución serán motivo de estudio ulterior

entrada y salida (ver 3.1.3) y

$$\text{evaluar} = \lambda\alpha \cdot \text{si vacía}(\varphi_o(\Theta_o)) \text{ entonces } \alpha \text{ sino } \mathcal{I}[\![\text{ calcular}]\!]\ \alpha'$$

donde

$$\alpha' = << \varphi_v[\text{cab}\varphi_o(\Theta_o) \mid \mathcal{E}[\![\varphi_e(\text{cab}\varphi_o(\Theta_o))]\!]\alpha, \varphi_e, \varphi_t >, \varphi_o[\Theta_o \mid \text{cola}\varphi_o(\Theta_o)], \varphi_r >$$

La interpretación evalúa la secuencia de ordenamiento sin interrupciones. Los operadores de cab y cola son los usuales de cadenas. Seleccionar un orden de cálculo (Θ en nuestro caso) no ocurre en las hojas de cálculo. Normalmente es el orden por dependencia funcional (Θ_1), y no explicitado cuando ocurren circularidades.

3.4 Conclusiones

Hemos presentado una descripción formal de una familia de aplicaciones denominadas "Hojas de Cálculo". En esta descripción se establece en forma precisa, concisa y sistemática los conceptos que aparecen en dicha familia de aplicaciones, permitiendonos analizar posibles fallas o semánticas "no ortodoxas" para esos conceptos en algunas aplicaciones.

Pensamos que conceptos tales como: secuencia de evaluación, tratamiento de regiones, manejo de tipos evidencian diferencias en las distintas aplicaciones y hasta en una misma aplicación en distintas plataformas. Por último, disponer de una especificación formal es disponer de una herramienta de verificación y/o análisis.

Referencias

[1] Scott, D.; "Domains for Denotational Semantics", ICALP'82.

[2] Soler, R.;"Fundamentos de un Meta-Lenguaje para Describir Semántica", Dpto. de Computación, U.C.V.,1982.

[3] Meyer, B.;"Introduction to the Theory of Programming Lenguages", Prentice Hall Series in Computer Science, 1990.

[4] Arráiz E.,Soler R.,Zoltan C.;"Análisis de Hojas de Cálculo", a ser publicado, 1992.