Una especificación formal de tipos estructurados de datos (Parte segunda).

Por: F. Orejas. Facultad de Matemáticas.
Universidad Complutense. Madrid.

3. Equivalencia debil de tipos estructurados de datos

En la primera parte de este trabajo se dió una definición formal del tipo estructurado array [1..n] de enteros. A continuación se verá otra definición del mismo tipo estructurado, análoga a la que usualmente se utiliza al establecer la semantica (denotacional) standard de los lenguages de programación que incorporan este tipo estructurado.

Ejemplo 6':

El tipo estructurado de datos array [1..n] de enteros puede ser formalmente definido como el tipo $\Sigma(Ar'_o, \{M'_{:=}\})$, siendo Ar'_o la estructura de datos:

$$Ar'_o = (\{O_{enteros}, O_{posiciones}\}, \{f_{dirección}, f_{valor}\})$$
$$D = \{enteros\}, \quad A = \{dirección, valor\}.$$
$$O_{enteros} = \mathbb{Z} \quad O_{posiciones} = \{p_o, p_1, \ldots, p_n, \ldots\}$$
$$f_{dirección} : O_{enteros} \longrightarrow O_{posiciones}$$
$$f_{dirección}(a) = \text{indefinido} \quad \forall a \in O_{enteros}$$

$$f_{valor} : O_{posiciones} \longrightarrow O_{enteros}$$
$$f_{valor}(p) = \text{indefinido} \quad \forall p \in O_{posiciones}$$

y siendo $M'_{:=}$ la modificación:

$$M'_{:=} : |Ar'_o| \times O_{enteros} \times O_{enteros} \longrightarrow |Ar'_o|$$

La estructura $M'_{:=}(Ar, a, b)$ sería, intuitivamente, la que se obtendria despues de realizar la operación $Ar[a] := b$; es decir:

$$M'_{:=}(Ar,a,b) = \text{indef.} \quad \text{si } a > n \text{ ó } a < 1$$

en otro caso $M'_{:=}(Ar,a,b)$ será una nueva estructura, Ar', cuyas funciones de acceso estarán definidas de la siguiente forma:

sea $p_a = f^{Ar}_{dirección}(a)$, si $f^{Ar}_{dirección}(a)$ está definida, sino sea p_a una posición tal que $\forall c \in O_{enteros}$ $f^{Ar}_{dirección}(c) \neq p_a$. Entonces:

$$f^{Ar'}_{dirección}(c) = \begin{cases} f^{Ar}_{dirección}(c) & \text{si } c \neq a \\ p_a & \text{si } c = a \end{cases}$$

$$f^{Ar'}_{valor}(p) = \begin{cases} f^{Ar}_{valor}(p) & \text{si } p \neq p_a \\ b & \text{si } p = p_a \end{cases}$$

Es evidente que tanto la especificación del ejemplo 6 como la del 6' definen, desde un punto de vista intuitivo, el mismo tipo estructurado. Será pues necesario establecer alguna clase de relación de equivalencia entre tipos estructurados de datos, mas amplia que la ya estudiada, capaz de considerar idénticos tipos estructurados que, aun estando contenidos en tipos generalizados diferentes, definan de alguna forma operaciones de acceso y de modificación similares.

Notación: Sea J un conjunto de indices de funciones y sea $x \in \mathcal{H}(J)$, $x = j_1 \ldots j_n$, se denotará por f_x a $f_{j_1} \ldots . f_{j_n}$ y por $a(1,x)$ y $a(2,x)$ a $a(1,j_n)$ y $a(2,j_1)$ respectivamente.

Definición

Dados dos conjuntos de indices J_1 y J_2, una función parcial $\Psi: \mathcal{H}(J_1) \longrightarrow \mathcal{H}(J_2)$ es un homomorfismo sii $\forall x,y \in \mathcal{H}(J_1)$ si $\Psi(x)$ y $\Psi(y)$ están definidos y $xy \in \mathcal{H}(J_1)$ entonces $\Psi(xy)$ está definido y $\Psi(xy) = \Psi(x)\Psi(y)$.

Se denomina nucleo de Ψ al conjunto:

$\{x \in Dom(\Psi) / no(\exists y,z \in Dom(\Psi)\ [yz=x])\}$.

Se dice que   es finitamente caracterizable si su nucleo es finito.

Definición

Sean $E = (\{O_i\}_{i\in I}, \{f_j\}_{j \in J})$ y $E' = (\{O_{i'}\}_{i' \in I'}, \{f_{j'}\}_{j' \in J'})$ dos estructuras de datos, sea $\varphi : D^E \longrightarrow D^{E'}$ una funcion parcial y sea $\Psi: \mathcal{H}(A^E) \longrightarrow \mathcal{H}(A^{E'})$ un homomorfismo finitamente caracterizable, se dice que (φ, Ψ) es una representación directa de E en E' sii: 1) $\forall i \in D^E [\varphi(i)$ esta definido $\Rightarrow O_i = O_{\varphi(i)}]$

2) $\forall x \in \mathcal{H}_D(A^E)$ se verifica que $\Psi(x)$ está definido $\wedge$ $\varphi(a(2,x)) =$
$= a(2,\Psi(x)) \wedge \varphi(a(1,x)) = a(1,\Psi(x)) \wedge f^E_x = f^{E'}_{\Psi(x)}$

Ejemplo 7

Sean Ar_o y Ar'_o las estructuras iniciales de los tipos estructurados definidos en los ejemplos 6 y 6' respectivamente, y sean φ_1 y φ_2 dos funciones parciales:

$$\varphi_1 : D^{Ar_o} \longrightarrow D^{Ar'_o} \qquad \varphi_2 : D^{Ar'_o} \longrightarrow D^{Ar_o}$$

$$\varphi_1(\text{enteros}) = \text{enteros} \qquad \varphi_2(\text{enteros}) = \text{enteros}$$

y sean Ψ_1 y Ψ_2 dos homomorfismos finitamente caracterizables:

$$\Psi_1 : \mathcal{H}(A^{Ar_o}) \longrightarrow \mathcal{H}(A^{Ar'_o})$$
$$\Psi_2 : \mathcal{H}(A^{Ar'_o}) \longrightarrow \mathcal{H}(A^{Ar_o})$$

definidos en su nucleo:

$\Psi_1(\text{valor}) = \text{valor}\cdot\text{direccion}$

$\Psi_2(\text{valor}\cdot\text{dirección}) = \text{valor}$

Entonces se verifica que Ar_o es directamente representable en Ar'_o mediante (φ_1,Ψ_1) y Ar'_o es directamente representable en Ar_o mediante (φ_2,Ψ_2). En efecto:

1) $O^{Ar_o}_{\text{enteros}} = \mathbb{Z}$ y $O^{Ar'_o}_{\varphi_1(\text{enteros})} = O^{Ar'_o}_{\text{enteros}} = \mathbb{Z}$

2) $x \in \mathcal{H}_D(A^{Ar_o}) \Rightarrow x = \text{valor}\cdot \ldots\ldots \cdot\text{valor}$. Por tanto bastará ver que para $x = \text{valor}$ se cumple la condición, esto es:

- $\varphi_1(a(1,\text{valor})) = \varphi_1(\text{enteros}) = \text{enteros} = a(1,\text{valor}\cdot\text{dirección}) = a(1,\Psi_1(\text{valor}))$.
- $\varphi_1(a(2,\text{valor})) = \varphi_1(\text{enteros}) = \text{enteros} = a(2,\text{valor}\cdot\text{dirección}) = a(2,\Psi_1(\text{valor}))$.
- $\forall\, b \in O_{\text{enteros}} \quad f^{Ar_o}_{\text{valor}}(b) = \text{indefinido} = f^{Ar'_o}_{\text{valor}} f^{Ar'_o}_{\text{dirección}}(b) = f^{Ar'_o}_{\Psi_1(\text{valor})}(b)$.

Analogamente se vería para (φ_2,Ψ_2).

Es decir, se podrán considerar equivalentes todo par de estructuras que verifiquen que cada una es directamente representable en la otra.

Definición

Sean E y E' dos estructuras de datos y sean $\{M_k\}_{k\in K}$ y $\{M'_k\}_{k\in K}$ dos conjuntos de modificaciones sobre los tipos generalizados de E y de E', respectivamente, que verifican que: $\forall k\in K\,[m^E(k) = m^{E'}(k)$ $\forall p\,[1\leq p\leq m(k) \Rightarrow O^E_{1(k,p)} = O^{E'}_{1(k,p)}]]$ (*). Se define, entonces, el conjunto $\mathcal{R}(E,E';\{M_k\}_{k\in K},\{M'_k\}_{k\in K})$ como:

$$\mathcal{R}(E,E';\{M_k\}_{k\in K},\{M'_k\}_{k\in K}) = \bigcup_{n\in\mathbb{N}} \mathcal{R}_n(E,E';\{M_k\}_{k\in K},\{M'_k\}_{k\in K})$$

siendo:

$$\mathcal{R}_o(E,E';\{M_k\}_{k\in K},\{M'_k\}_{k\in K}) = \{\langle E,E'\rangle\}$$

$$\mathcal{R}_{n+1}(E,E',\{M_k\}_{k\in K},\{M'_k\}_{k\in K}) = \{\langle E_1,E_2\rangle \,/\, \exists\langle E_3,E_4\rangle\in\mathcal{R}_n\,\exists k\in K\,\exists o_1\in O_{1(k,1)} \ldots\ \exists o_{m(k)}\in O_{1(k,m(k))}[E_1 = M_k(E_3,o_1,\ldots,o_{m(k)}) \wedge E_2=M_k(E_3,o_1,\ldots,o_{m(k)})]\}$$

Definición

Sean $\mathcal{E} = \mathcal{E}(Eo,\{M_k\}_{k\in K})$ y $\mathcal{E}' = \mathcal{E}(E'_o,\{M'_k\}_{k\in K})$ dos tipos estructurados de datos que cumplen la propiedad (*) de la definición anterior. $\mathcal{E}$ y $\mathcal{E}'$ son debilmente equivalentes, $\mathcal{E}\sim\mathcal{E}'$, siiexisten dos funciones parciales φ_1 y φ_2:

$$\varphi_1 : D^{Eo} \longrightarrow D^{E'o} \qquad \varphi_2 : D^{E'o} \longrightarrow D^{Eo}$$

y dos homomorfismos, Ψ_1 y Ψ_2, finitamente caracterizables:

$$\Psi_1 : \mathcal{K}(A^{Eo}) \longrightarrow \mathcal{K}(A^{E'o}) \qquad \Psi_2 : \mathcal{K}(A^{E'o}) \longrightarrow \mathcal{K}(A^{Eo})$$

que verifican:

1) $\forall k\in K\,\forall p[\,1\leq p\leq m(k) \Rightarrow 1^{\mathcal{E}'}(k,p)=\varphi_1(1^{\mathcal{E}}(k,p)) \wedge 1^{\mathcal{E}}(k,p) = = \varphi_2(1^{\mathcal{E}'}(k,p))]$
2) $\exists E_1\in\mathcal{E}\,\forall\langle E,E'\rangle\in\mathcal{R}(E_1,E'_o,\{M_k\}_{k\in K},\{M'_k\}_{k\in K})$ E es directamente representable en E' mediante (φ_1,Ψ_1) y E' es directamente representable en E mediante (φ_2,Ψ_2)
3) $\exists E'_1\in\mathcal{E}'\,\forall\langle E,E'\rangle\in\mathcal{R}(E_o,E'_1,\{M_k\}_{k\in K},\{M'_k\}_{k\in K})$ E es directamente repre-

sentable en E´ mediante (φ_1,ψ_1) y E´es directamente representable en E mediante (φ_2,ψ_2).

Teorema

$\sim$ es una relación de equivalencia.

Antes de demostrar el teorema se van a ver unos lemas:

Lema

Sean E_1, E_2 y E_3 tres estructuras de datos tales que E_1 es directamente representable en E_2 mediante (φ_1,ψ_1) y E_2 es directamente representable en E_3 mediante (φ_2,ψ_2), entonces E_1 es directamente representable en E_3 mediante $(\varphi_2\varphi_1,\psi_2\psi_1)$

Demostración

1) $\forall i \in D^{E1}$ $\varphi_2\varphi_1(i)$ esta definido $\Rightarrow \varphi_1(i)$ esta definido y $\varphi_2(\varphi_1(i))$ está definido $\Rightarrow O_i^{E1}=O^{E2}_{\varphi_1(i)}=O^{E3}_{\varphi_2\varphi_1(i)}$

2) $\forall x \in \mathcal{H}_D(A^{E1})$ $\psi_1(x)$ está definido $\Rightarrow \psi_1(x) \in \mathcal{H}_D(A^{E2}) \wedge \psi_2\psi_1(x)$ está definido. Por otra parte: $\varphi_1(a(2,x))=a(2,\psi_1(x)) \wedge \varphi_2(a(2,\psi_1(x)))= a(2,\psi_2\psi_1(x)) \Rightarrow \varphi_2\varphi_1(a(2,x))=a(2,\psi_2\psi_1(x))$; y $\varphi_1(a(1,x))=a(1,\psi_1(x)) \wedge \varphi_2(a(1,\psi_1(x)))=a(1,\psi_2\psi_1(x)) \Rightarrow \varphi_2\varphi_1(a(1,x))=a(1,\psi_2\psi_1(x))$. Ademas: $f_x^{E1}= f^{E2}_{\psi_1(x)} = f^{E3}_{\psi_2\psi_1(x)}$.

Lema

Sean E, E´ y E˜ tres estructuras de datos y sean $\{M_k\}_{k\in K}$, $\{M_k'\}_{k\in K}$ y $\{M_k^{\sim}\}_{k\in K}$ tres familias de modificaciones sobre los tipos generalizados de las tres estructuras, respectivamente y tal que cumplen la propiedad (*) dos a dos. Entonces se verifica:

$$\mathcal{R}(E,E',\{M_k\}_{k\in K},\{M_k'\}_{k\in K}) = \{\langle E_1,E_1'\rangle / \exists E_1^{\sim} [\langle E_1,E_1^{\sim}\rangle \in \mathcal{R}(E,E^{\sim},\{M_k\}_{k\in K}, ,\{M_k^{\sim}\}_{k\in K}) \wedge \langle E_1^{\sim},E_1'\rangle \in \mathcal{R}(E^{\sim},E',\{M_k^{\sim}\}_{k\in K},\{M_k'\}_{k\in K})]\}$$

Lema

Sean E_1, E_2, E_3 y E_4 cuatro estructuras de datos tales que $|E_1| = |E_3|$ y $|E_2|=|E_4|$ y sean $\{M_k\}_{k\in K}$ y $\{M_k'\}_{k\in K}$ dos familias de modificaciones sobre $|E_1|$ y $|E_2|$, respectivamente, que cumplen la propiedad (*). Entonces $\langle E_3,E_4\rangle \in \mathcal{R}(E_1,E_2,\{M_k\}_{k\in K},\{M'_k\}_{k\in K}) \Rightarrow \mathcal{R}(E_3,E_4,\{M_k\}_{k\in K},\{M'_k\}_{k\in K}) \subset \mathcal{R}(E_1,E_2,\{M_k\}_{k\in K},\{M'_k\}_{k\in K})$.

La demostración de estos dos lemas es sencilla aunque algo engorrosa. Para la demostración del primero bastará ver, por inducción, que para todo n, $\mathcal{R}_n(E,E',\{M_k\}_{k\in K},\{M'_k\}_{k\in K}) = \{\langle E_1,E'_1\rangle / \exists E''_1 [\langle E_1,E''_1\rangle \in \mathcal{R}_n(E,E'',\{M_k\}_{k\in K},\{M''_k\}_{k\in K}) \wedge \langle E''_1,E'_1\rangle \in \mathcal{R}_n(E'',E',\{M''_k\}_{k\in K},\{M'_k\}_{k\in K})]\}$. Para la demostración del segundo lema, bastará ver, tambien por inducción, que si $\langle E_3,E_4\rangle \in \mathcal{R}_m(E_1,E_2,\{M_k\}_{k\in K},\{M'_k\}_{k\in K})$, entonces para todo n $\mathcal{R}_n(E_3,E_4,\{M_k\}_{k\in K},\{M'_k\}_{k\in K}) \subset \mathcal{R}_{n+m}(E_1,E_2,\{M_k\}_{k\in K},\{M'_k\}_{k\in K})$.

Demostración del teorema

La demostración de la reflexividad y de la simetria son triviales, bastará por tanto con demostrar la transitividad.

Sean $\mathcal{E} = \mathcal{E}(E_o, M_{k\,k\,K})$, $\mathcal{E}' = \mathcal{E}(E'_o, M'_{k\,k\,K})$ y $\mathcal{E}'' = \mathcal{E}(E''_o, M_{k\,k\,K})$ tales que $\mathcal{E} \sim \mathcal{E}' \wedge \mathcal{E}' \sim \mathcal{E}''$ se va a ver que $\mathcal{E} \sim \mathcal{E}''$, para ello se demostrará solo la segunda condición de la definición de $\sim$, ya que la primera es trivial y la tercera es similar a la segunda.

Por ser $\mathcal{E} \sim \mathcal{E}'$ y $\mathcal{E}' \sim \mathcal{E}''$ existen $E'_1 \in \mathcal{E}'$ y $E''_1 \in \mathcal{E}''$ y $\varphi_1, \varphi_2, \varphi_3, \varphi_4, \psi_1, \psi_2, \psi_3$ y ψ_4 tales que $\forall \langle E,E\rangle \in \mathcal{R}(E_o,E'_1,\{M_k\}_{k\in K},\{M'_k\}_{k\in K})$ E es directamente representable en E' mediante (φ_1,ψ_1) y E' es directamente representable en E mediante (φ_2,ψ_2). Ademas $\forall \langle E';E''\rangle \in \mathcal{R}(E'_o,E''_1,\{M'_k\}_{k\in K},\{M''_k\}_{k\in K})$ E' es directamente representable en E'' mediante (φ_3,ψ_3) y E'' es directamente representable en E' mediante (φ_4,ψ_4).

Por pertenecer E'_1 a $\mathcal{E}'$ existirá una E''_2 tal que $\langle E'_1,E''_2\rangle \in \mathcal{R}(E'_o,E''_2,\{M'_k\}_{k\in K},\{M''_k\}_{k\in K})$. Vamos a ver que $\forall \langle E,E''\rangle \in \mathcal{R}(E_o,E''_2,\{M_k\}_{k\in K},\{M''_k\}_{k\in K})$ E es directamente representable en E'' mediante $(\varphi_3\varphi_1,\psi_3\psi_1)$ y E'' es directamente representable en E mediante $(\varphi_2\varphi_4,\psi_2\psi_4)$. En efecto, de acuerdo con el segundo lema $\langle E,E''\rangle \in \mathcal{R}(E_o,E''_2,\{M_k\}_{k\in K},\{M''_k\}_{k\in K}) \Rightarrow$

$\Rightarrow \exists E'[\langle E,E'\rangle \in \mathcal{R}(E_o,E_1',\{M_k\}_{k\in K},\{M_k'\}_{k\in K}) \wedge \langle E',E''\rangle \in \mathcal{R}(E_1',E_2'',\{M_k'\}_{k\in K},$ $\{M_k''\}_{k\in K})]$, pero de acuerdo con el tercer lema, tambien se tiene que $\langle E',E''\rangle \in \mathcal{R}(E_o',E_1'',\{M_k'\}_{k\in K},\{M_k''\}_{k\in K})$ y de acuerdo con el primer lema se tiene que E es directamente representable en E″ mediante $(\varphi_3\varphi_1,$ $\psi_3\psi_1)$ y E″ es directamente representable en E mediante $(\varphi_2\varphi_4,\psi_2\psi_4)$.

Ejemplo 8

Las dos definiciones del tipo estructurado de datos array [1..n] de enteros, de los ejemplos 6 y 6′, son debilmente equivalentes. En efecto, vamos a ver que para todo par de arrays Ar y Ar′tales que $\langle Ar,Ar'\rangle \in \mathcal{R}(Ar_o,Ar_o',\{M_{:=}\},\{M_{:=}'\})$, se verifica que Ar es directamente representable en Ar′ mediante (φ_1,ψ_1) y Ar′ es directamente representable en Ar mediante (φ_2,ψ_2), siendo φ_1,ψ_1, φ_2 y ψ_2 las del ejemplo 7.

La demostración se hara por inducción sobre los conjuntos $\mathcal{R}_n$.

caso 0 Ya se ha visto en el ejemplo 7 que Ar_o y Ar_o' verifican que cada una es directamente representable en la otra.

caso n+1 $\langle Ar,Ar'\rangle \in \mathcal{R}_{n+1} \Rightarrow \exists \langle Ar_1,Ar_1'\rangle \in \mathcal{R}_n\ \exists x,y\in \mathbb{Z}$ tal que $Ar =$ $= M_{:=}(Ar_1,x,y) \wedge Ar' = M_{:=}'(Ar_1',x,y)$. Vamos a ver que si Ar_1 es directamente representable en Ar_1' mediante (φ_1,ψ_1) entonces Ar lo es en Ar′. Para ello bastará ver que para todo z : $f^{Ar}_{valor}(z) = f^{Ar'}_{valor}\cdot$ $\cdot f^{Ar'}_{dirección}(z)$. Sea $p_x = f^{Ar'}_{dirección}(x)$, entonces:

$-z\neq x \Rightarrow f^{Ar}_{valor}(z)=f^{Ar_1}_{valor}(z) \wedge f^{Ar'}_{dirección}(z)=f^{Ar_1'}_{dirección}(z)\neq p_x$ $\Rightarrow f^{Ar'}_{valor}f^{Ar'}_{dirección}(z)=f^{Ar_1'}_{valor}f^{Ar_1'}_{dirección}(z) \Rightarrow f^{Ar}_{valor}(z)=f^{Ar'}_{valor}\cdot$ $\cdot f^{Ar'}_{dirección}(z)$.

$-z=x \Rightarrow f^{Ar}_{valor}(z)=y \wedge f^{Ar'}_{dirección}(z)=p_x \wedge f^{Ar'}_{valor}(p_x)=y \Rightarrow$ $f^{Ar}_{valor}(z)=f^{Ar'}_{valor}f^{Ar'}_{dirección}(z)$.

La misma argumentación sirve para demostrar que Ar′es directamente representable en Ar mediante (φ_2,ψ_2).

Lema

$E \equiv E' \Rightarrow E$ es directamente representable en E' mediante $(id_D, id_{\mathcal{H}_D(A)})$

Demostración

1) $\forall i \in D^E \, [O_i^E = O_i^{E'}]$ en virtud de la propiedad 1) de la equivalencia de estructuras de datos.

2) $\forall x \in \mathcal{H}_D(A^E)$ se verifica que $\Psi(x)=x$ está definido, siendo $\varphi(a^E(2,x)) = a^E(2,x) = a^{E'}(2,\Psi(x))$ y $\varphi(a^E(1,x)) = a^E(1,x) = a^{E'}(1,\Psi(x))$, debido a que E y E′ tienen los mismos conjuntos de indices I y J. Ademas $f_x^E = f_x^{E'}$ por las propiedades 2) y 3) de la definición de la relación de equivalencia entre estructuras de datos.

Proposición

$\mathcal{E} \equiv \mathcal{E}' \Rightarrow \mathcal{E} \sim \mathcal{E}'$

Demostración

Sean $\varphi_1 = \varphi_2 = id_D$ y $\Psi_1 = \Psi_2 = id_{\mathcal{H}_D(A)}$. La condición 1) de la definición de $\sim$ es trivial. Veamos la condición 2):
Sea $E_o^{\sim} \in \mathcal{E}'$ tal que $E_o \equiv E_o^{\sim}$ (de acuerdo con la definición de equivalencia entre tipos estructurados existe tal $E_o^{\sim}$) vamos a ver, por inducción, que $\forall \langle E, E' \rangle \in \mathcal{R}(E_o, E_o^{\sim}, \{M_k\}_{k\in K}, \{M_k'\}_{k\in K})$ se verifica que $E \equiv E'$; con lo que de acuerdo con el Lema anterior se habría demostrado lo que se pretendía. En efecto:

$-E_o \equiv E_o^{\sim}$ por hipótesis

$-\langle E, E' \rangle \in \mathcal{E}_{n+1} \Rightarrow \exists \langle E_1, E_1' \rangle \in \mathcal{E}_n$ ($E_1 \equiv E_1'$ por hipótesis de inducción)

$\exists k \in K \; \exists o_1 \in O_{1(k,1)} \; \ldots \; \exists o_{m(k)} \in O_{1(k,m(k))} \; [E = M_k^{\mathcal{E}}(E_1, o_1, \ldots, o_{m(k)}) \wedge E' = M_k^{\mathcal{E}'}(E_1', o_1, \ldots, o_{m(k)})] \Rightarrow E \equiv E'$.

La condición 3) se demostraría en forma análoga.

Proposición

Sean $\mathcal{E} = \mathcal{E}(E_o, \{M_k\}_{k\in K})$ y $\mathcal{E}' = \mathcal{E}(E_o', \{M_k'\}_{k\in K})$ dos tipos estructurados debilmente equivalentes. Entonces existe $\mathcal{E}'' = \mathcal{E}(E_o^{\sim}, \{M_k\}_{k\in K})$

tal que $|E_o^{\sim}|=|E_o| \wedge \mathcal{T} \sim \mathcal{T}'' \wedge \exists \varphi_1, \psi_1, \varphi_2, \psi_2 \ \forall \langle E^{\sim}, E' \rangle \in \mathcal{R}(E_o^{\sim}, E_o', \{M_k\}_{k\in K}, \{M_k'\}_{k\in K})$ $E^{\sim}$ es directamente representable en E' mediante (φ_1, ψ_1) y E' es directamente representable en $E^{\sim}$ mediante (φ_2, ψ_2) .

<u>NOTA</u> Como consecuencia de esta proposición, en lo que sigue cuando se consideren tipos estructurados debilmente equivalentes, se supondrá, sin pérdida de generalidad, que la construcción de $\mathcal{R}$ se hace a partir de las estructuras iniciales de cada tipo.

La equivalencia debil nos permite, por tanto, considerar similares tipos estructurados cuyos tipos generalizados subyacentes sean distintos, pero que de alguna forma (se obtienen resultados iguales con operaciones similares ..) definen un mismo tipo de datos. Esta relación permite por tanto abstraerse de los detalles mas subjetivos de una definición, es decir, de las funciones escondidas y de los objetos que no son datos ó que aun siendo datos hayan sido incorporados a la definición de manera mas o menos arbitraria. En este sentido, las clases de equivalencia debil de tipos estructurados capturarían,de una forma muy constructiva, el concepto de tipo abstracto de datos, tal como se presenta en ADJ, Guttag, Zilles etc.

Supongase ahora que se tienen dos definiciones de un mismo tipo de datos, i.e. dos definiciones de tipos estructurados de datos debilmente equivalentes, parece conveniente dar algun criterio de decisión sobre cual de las dos definiciones es "mejor", cual utiliza menos elementos arbitrarios, por ejemplo funciones escondidas, ya que una definición sería mejor, en este sentido , debido a que una implementación del tipo estructurado necesitaría tener en cuenta menos elementos. Naturalmente la calificación de "mejor" es puramente formal, ya que cuestionar cual de las dos definiciones es mejor realmente implicaría entrar en consideraciones filosóficas entre las que estaría probablemente la de decidir cual es mas comprensible.

Definición

Sean $\mathcal{E}$ y $\mathcal{E}'$ dos tipos estructurados de datos debilmente equivalentes. $\mathcal{E}$ es una especificación mejor que $\mathcal{E}'$, $\mathcal{E} \prec \mathcal{E}'$, sii existen funciones totales:

$$\varphi: I^{\mathcal{E}} \longrightarrow I^{\mathcal{E}'},\ \psi: \mathcal{H}(J^{\mathcal{E}}) \longrightarrow \mathcal{H}(J^{\mathcal{E}'}),\ \forall i \in I\ \xi_i: O_i^{\mathcal{E}} \longrightarrow O_{\varphi(i)}^{\mathcal{E}'}$$

tales que φ es inyectiva, ψ es un homomorfismo inyectivo finitamente caracterizable, $\forall i \in D$ ξ_i es la identidad y $\forall i \in I-D$ ξ_i es inyectiva y tal que verifican:

1) $\forall j \in J[\varphi(a^{\mathcal{E}}(1,j)) = a^{\mathcal{E}'}(1,\psi(j)) \wedge \varphi(a^{\mathcal{E}}(2,j)) = a^{\mathcal{E}'}(2,\psi(j))]$.
2) $\forall k \in K\ \forall p[1 \leq p \leq m(k) \Rightarrow \varphi(1^{\mathcal{E}}(k,p)) = 1^{\mathcal{E}'}(k,p)]$.
3) $\forall \langle E,E' \rangle \in \mathcal{R}(E_o^{\mathcal{E}}, E_o^{\mathcal{E}'}, \{M_k^{\mathcal{E}}\}_{k \in K}, \{M_k^{\mathcal{E}'}\}_{k \in K})[\forall i \in D\ \forall o \in O_i\ \forall x \in \mathcal{H}(J)$ o es alcanzable en E a traves del camino x sii o es alcanzable en E′ a traves del camino $\psi(x) \wedge \forall i_1, i_2 \in D\ \forall o_1 \in O_{i_1}\ \forall o_2 \in O_{i_2}\ \forall x \in \mathcal{H}(J)$ o_1 es alcanzable en E desde o_2 a traves del camino x sii o_1 es alcanzable en E′ desde o_2 a través del camino $\psi(x)]$.

Proposición

$\prec$ es un preorden.

Proposición

$\mathcal{E} \prec \mathcal{E}' \wedge \mathcal{E}' \prec \mathcal{E} \Rightarrow$ las φ y ψ de la definición son biyecciones.

Es decir que si se considera un conjunto T de tipos estructurados debilmente equivalentes, la relación $\prec$ puede ser considerada casi un orden parcial sobre las clases de equivalencia de $\equiv$ de T. (El "casi" quiere decir que es un orden parcial salvo biyección en los conjuntos de indices).

Ejemplo 9

La especificación del tipo estructurado Array[1..n] de enteros del ejemplo 6 es mejor que la del ejemplo 6'. En efecto, tomando como $\zeta_{enteros}$ la identidad en $\mathbb{Z}$ y como φ y ψ las funciones φ_1 y ψ_1 del ejemplo 7, se tiene que se cumplen todas las condiciones de la definición de $\prec$.

Sería facil comprobar caso por caso que la definición de la relación $\prec$ caracteriza la idea del ahorro en los elementos superfluos de una especificación. Bastaría ir viendo,sucesivamente, que caracteriza el exceso de funciones escondidas, de conjuntos de objetos que no son datos, el excesivo tamaño de dichos conjuntos, etc.

Definición

Un tipo estructurado $\mathcal{E}$ es minimal con respecto a $\prec$ sii
$\forall \mathcal{E}' \quad \mathcal{E}' \prec \mathcal{E} \Rightarrow \mathcal{E} \prec \mathcal{E}'$

Teorema

Existen tipos $\mathcal{E}_1$ y $\mathcal{E}_2$ debilmente equivalentes, minimales e incomparables con respecto a $\prec$.

Demostración

Basta demostrar que dos especificaciones del tipo "cola de dos extremos (deque) de enteros", una de las cuales tiene una función de sucesión escondida dirigida del primer elemento al último, y teniendo la otra la función de sucesión invertida, es decir dirigida del último elemento al primero, son minimales e incomparables.

Este resultado puede considerarse en cierta medida desesperanzador, ya que nos demuestra la inexistencia de la mejor definición, lo idoneo seria demostrar la existencia y unicidad (salvo isomorfía) de especificaciones mínimas que servirían, de alguna forma para realizar implementaciones mínimas. Sin embargo, el resultado parece normal, ya que la necesidad de incluir en las especificaciones elementos arbitrarios permite precisamente que la posibilidad de elección entre estos elementos sea grande, dando lugar, cada una de las posibles opciones, a distintas especificaciones minimales.

4. Conclusiones

Se ha presentado un nuevo método para la especificación de tipos de datos. El método parece ser superior a los otros modelos constructivos existentes en el sentido de que comparte todas sus ventajas, constructibilidad, comprensibilidad y quizas el bajo nivel de abstracción (el bajo nivel de abstracción es simultaneamente una ventaja y un inconveniente), pero sin embargo no comparte todos sus inconvenientes, en particular, la sujeción a un modelo de definición previo (como pueden ser los grafos ó las listas) lo que permite mayor libertad y por tanto mayor facilidad para establecer las especificaciones.

Con respecto a las especificaciones axiomáticas (algebráicas), el nuevo método comparte las ventajas e inconvenientes de otros métodos constructivos, es decir las especificaciones las especificaciones son mas sencillas de establecer pero mas complicadas de utilizar en verificaciones y demás.

La proximidad teórica del nuevo modelo con las teorías algebráicas de tipos de datos proporciona una nueva ventaja con respecto al resto de los métodos constructivos. En efecto, como se señaló en la introducción, la presentación de este trabajo es bastante algebráica, no es pues dificil llegar a una algebraización completa, es decir a la definición de tipos de datos como álgebras ó como categorias de álgebras, con lo que una especificación $\mathcal{E}$ realizada de acuerdo con el nuevo metodo constituirá un magnífico modelo para la construcción y verificación de una especificación axiomática Σ , pues para verificar Σ bastará comprobar que Σ es inicial en la categoria de álgebras que verifican Σ , lo cual puede ser engorroso pero nunca complicado.

El nuevo método es ademas especialmente adecuado (con ligeras modificaciones) para la definición de la semántica (denotacional) de lenguajes de programación que incorporan tipos estructurados de datos.

Referencias

ADJ: GOGUEN, THATCHER, WAGNER and WRIGHT (1975), "Abstract data types as initial algebras and correctness of data representations" Proceedings of the Conference on Computer Graphics, Pattern Recognition and Data Structure.

ADJ: THATCHER, WAGNER and WRIGHT (1978), "Data type specification: parameterization and the power of specification techniques". Proceedings of the SIGACT 10th Annual Symposium on Theory of Computing

EARLEY, "Toward an understanding of data structures". CACM vol 14, 1971.

GALINIER (1978), "A software design methodology and associated tools. Partial preliminary draft. Sin publicar.

GUTTAG (1975), "The specification and application to programming of abstract data types". Univ. of Toronto. Computer Systems Research Group. Tech. Report CSRG-59.

LISKOV and ZILLES (1975), "Specification Techniques for Data Abstractions". IEEE Trans. on Software Engineering, vol. SE-1, pp.7-19

LISKOV and BERZINS (1977), "An appraisal of program specifications".

MIT, Computation Structures Group Memo 141-1.

MAJSTER (1977), "Extended directed graphs, a formalism for structured data and data structures". Acta Informatica, vol. 6

OLLONGREN (1977), "Definition of programming languages by interpreting automata". Ac. Press.

OREJAS (1978), "Una teoria de tipos de datos y su aplicación a la concepción de programas". Proceedings del CIL 79.

OREJAS (1979), "The semantics of data structures". Sin publicar.

ZILLES (1974), "Algebraic specification of data types". Progress Rep. XI, Lab. for Comp. Sc., MIT.