Apuntes de Programación y estructuras de datos. Implementación de TADs

Nikos Mylonakis

nicos@lsi.upc.edu

Dept. Llenguatges i Sistemes Informátics Universitat Politécnica de Catalunya

Barcelona

- Recordemos la especificación de las pilas

$$
\begin{array}{l}
\textbf{modulo } Pila\_ent \\
\textbf{usa } entero \\
\textbf{ops} \\
\quad pila\_vacia :\to pila\_ent \\
\quad empilar : entero \times pila\_ent \to pila\_ent \; (\textbf{parcial}) \\
\quad desempilar : pila\_ent \to pila\_ent \\
\quad cima : pila\_ent \to entero
\end{array}
$$

axiomas

$$desempilar(pila\_vacia) \uparrow$$

$$cima(pila\_vacia) \uparrow$$

$$desempilar(empilar(i, p)) = p$$

$$cima(empilar(i, p)) = i$$

Una posible representación de esta especificación es mediante una tabla de enteros con un apuntador al último elemento empilado.

modulo $Pila\_ent$

usa $entero$

ops

  accion $pila\_vacia(\mathbf{ent/sal}\ p : pila\_ent)$

  accion $empilar(\mathbf{ent}\ i : entero, \mathbf{ent/sal}\ p : pila\_ent)$

  accion $desempilar(\mathbf{ent/sal}\ p : pila\_ent)(\mathbf{parcial})$

  funcion $cima(p : pila\_ent)$ **retorna** $i : entero$

implementacion

tipo

  tupla

    t : **tabla** $[1..50]$**de** $entero$;

    $pos : entero$

  ftupla

ftipo

accion $pila\_vacia$(**ent**/**sal** $p : pila\_ent$)

$\{Pre : Cierto\}$

$\{Post : p.pos = 0\}$

  $p.pos := 0$

faccion

$\textbf{accion}\ empilar(\textbf{ent}\ i : entero, \textbf{ent/sal}\ p : pila\_ent)$

$\{Pre : p.pos \geq 0 \wedge p.pos < 50\}$

$\{Post : p_e.t[1..p_e.pos] = p_s.t[1..p_e.pos]$

$\quad \wedge\, p_s.pos = p_e.pos + 1 \wedge p_s.t[p_e.pos + 1] = i\}$

$p.pos := p.pos + 1;$

$p.t[p.pos] := i;$

faccion

accion $desempilar$(**ent/sal** $p : pila\_ent$)
$\{Pre : p.pos > 0 \wedge p.pos \leq 50\}$
$\{Post : p_e.t[1..p_e.pos - 1] = p_s.t[1..p_e.pos - 1]$
$\quad \wedge\, p_s.pos = p_e.pos - 1\}$
$\quad p.pos := p.pos - 1;$
faccion
funcion $cima(p : pila\_ent)$ **retorna** $i : entero$
$\{Pre : p.pos > 0 \wedge p.pos \leq 50\}$
$\{Post : i = p_e.t[p_e.pos]\}$
$\quad i := p.t[p.pos];$
retorna i
ffuncion

- Esta implementación de las pilas parece correcta pero si intentamos demostrar que la implementación de las operaciones satisface las ecuaciones nos encontramos con algunos problemas
- No podemos demostrar que para toda pila p y entero i $desempilar(empilar(i, p)) = p$ pues la posición $p.t[p.pos + 1]$ en general es distinta de $p'.t[p'.pos + 1]$ donde $p' = desempilar(empilar(i, p))$
- Este problema se soluciona interpretando la igualdad de las ecuaciones por una relación de equivalencia ($\equiv$) que se tiene que definir para cada implementación concreta

- Existe otro problema que no permite demostrar que para todo pila y entero la ecuación se satisface pues la representación permite tener números negativos en el campo pos de la pila. Como no se satisface la precondición de las operaciones para estos casos se generan errores de programación al ejecutar las operaciones de la ecuación
- Este problema se soluciona definiendo un dominio de la representación que es el conjunto de valores que representa un valor abstracto del álgebra inicial de la especificación.

- Una propiedad de este dominio es que para todo valor del dominio, después de ejecutar cualquier operación, los valores resultantes han de pertenecer a este dominio. Por este motivo al dominio se le suele llamar invariante de la representación
- Para la implementación de las pilas, la relación de equivalencia se define como $p1 \equiv p2 \Leftrightarrow p1.pos = p2.pos \wedge p1.t[1..p1.pos] = p2.t[1..p2.pos]$. Esto hace que un término que denota una pila tenga múltiples representaciones mediante tablas
- Y el invariante de representación se define como $p.pos \geq 0 \wedge p.pos \leq 50$

Por tanto la especificación y diseño de una implementación consta de

- Definición de la representación mediante tipos
- Definición de la especificación pre/post de las operaciones y su código
- Equivalencia de la representación
- Invariante de la representación

Para demostrar que una implementación es correcta hay que comprobar

- El invariante de representación es invariante. Para ello se ha de comprobar que al ejecutarse la operación de inicialización el invariante se satisface, y después de ejecutarse el resto de operaciones el invariante se preserva
- Las ecuaciones se cumplen para todo elemento del dominio sustituyendo la igualdad por equivalencia
- Como ejercicio demostrar que la implementación de las pilas de enteros dada es correcta con respecto a su especificación