

Capitolo 10

Modularizzazione per tipo astratto: concettualizzazione

Nel capitolo precedente abbiamo sottolineato l'importanza della modularizzazione per tipo astratto. In questo capitolo e nel prossimo approfondiamo questa tematica riferendoci alla fase di concettualizzazione.

Abbiamo più volte ribadito che lo schema concettuale di progetto contiene da una parte l'indicazione dei moduli che hanno rilevanza concettuale in una applicazione con le loro relazioni, e dall'altra una specifica più dettagliata per ognuno dei moduli dello schema.

Il problema che affrontiamo ora riguarda i meccanismi mediante i quali si fornisce la specifica di un modulo per tipo astratto. Lo studio di questi meccanismi viene approfondito indagando sulla caratterizzazione formale dei tipi astratti di dato, cioè una caratterizzazione che ci consente di studiare le loro proprietà senza alcun riferimento agli strumenti tecnologici e realizzativi della progettazione. Seguiremo quindi un percorso che ci consentirà di trovare risposta a due importanti questioni:

1. Quali meccanismi possiamo utilizzare per specificare in modo preciso le proprietà di un tipo astratto nello schema concettuale?
2. A quale categoria di oggetti matematici si può fare riferimento per una caratterizzazione formale del concetto di tipo astratto, e come possiamo rendere la specifica di un tipo conforme a tale caratterizzazione?

Alla prima questione sono dedicati i paragrafi di cui si compone questo capitolo. Della seconda questione ci occuperemo nel prossimo capitolo. In altre parole, svolgeremo queste considerazioni in due passi. In questo capitolo introdurremo la nozione di specifica privilegiando gli aspetti intuitivi, mentre nel prossimo capitolo approfondiremo gli aspetti formali, mettendo in relazione tale nozione con un impianto algebrico e logico.

10.1 La specifica dei tipi astratti nella fase di concettualizzazione

La modularizzazione per tipo astratto in fase di concettualizzazione conduce il progettista alla individuazione dei tipi di dato che si ritengono interessanti nel problema in esame. L'interpretazione intuitiva di tipo di dato è quella che assegna a tale termine il significato di classe di oggetti su cui è possibile eseguire un insieme prefissato di operazioni. Il problema che affrontiamo in questo paragrafo riguarda la necessità di fornire, durante la fase di concettualizzazione, una specifica sufficientemente precisa di un tipo astratto di dato. Le considerazioni che abbiamo svolto sui principi di progettazione evidenziano che tale caratterizzazione deve soddisfare le seguenti condizioni:

- la descrizione del tipo sia chiara e non ambigua;
- le caratteristiche del tipo vengano specificate in modo completamente indipendente dalla rappresentazione dei valori e dalla realizzazione delle operazioni.

Ricordiamo che un tipo astratto di dato è un oggetto matematico costituito da: un insieme, detto il dominio del tipo, che raccoglie i valori del tipo, ed un insieme di funzioni, che corrispondono alle operazioni che si possono effettuare sui valori del tipo.

Un semplice esempio di tipo astratto è il tipo *Boolean*, in cui il dominio è formato dai due valori di verità (che possiamo chiamare "true" e "false") e le operazioni sono "or", "and" e "not", con il significato usuale.

Un esempio più complesso è il tipo astratto *Pila*, in cui il dominio, che chiameremo *Pila*, è formato da sequenze di elementi di un certo dominio T , e le operazioni sono:

- *PilaVuota*, che restituisce il valore corrispondente alla sequenza vuota.
- *EstVuota(p)*, che restituisce true se la pila p è il valore corrispondente a pila vuota, false altrimenti.
- *Push(p, e)*, che restituisce la pila ottenuta dalla pila p inserendo l'elemento e , che diventa il suo elemento affiorante.
- *Pop(p)* restituisce la pila ottenuta dalla pila p eliminando l'elemento affiorante.
- *Top(p)* restituisce l'elemento affiorante della pila p .

Quest'ultimo esempio ci chiarisce diversi aspetti che la definizione di tipo astratto presentata in precedenza tende a semplificare:

- in genere un tipo astratto è costituito da più domini, oltre al dominio del tipo (il cosiddetto dominio di interesse). Nel caso della pila, i domini in gioco sono: *Pila*, il dominio del tipo boolean, e T , il dominio degli elementi che formano le pile;
- le funzioni possono essere definite su diversi domini, ossia la loro applicazione può richiedere zero, uno o più argomenti, e restituiscono un valore di un certo dominio. Inoltre, la definizione delle funzioni richiede una descrizione della legge con cui la funzione calcola il risultato in dipendenza degli argomenti.

La seguente definizione coglie precisamente questi aspetti, ed evidenzia che, dal punto di vista matematico, un tipo astratto di dato può essere considerato come un'algebra eterogenea, cioè un'algebra che coinvolge più insiemi (cfr. capitolo 7).

Definizione 10.1.1 (Tipo astratto di dato) *Un tipo astratto di dato è un oggetto matematico costituito da due componenti:*

1. *una collezione di domini, uno dei quali, chiamato dominio di interesse, riveste particolare importanza, perché è il dominio del tipo, cioè rappresenta tutti i valori del tipo;*
2. *un insieme di funzioni, ciascuna delle quali ha come dominio di definizione il prodotto cartesiano di n domini che appartengono alla collezione di cui al punto 1, e come codominio un dominio che appartiene alla collezione stessa.*

Questa definizione chiarisce che, per potere parlare di un tipo astratto è necessario almeno munirsi di alcuni simboli che, a livello sintattico, consentano di denotare i domini e le funzioni del tipo. Questo aspetto sintattico viene colto dalla nozione di segnatura, che abbiamo già discusso nel capitolo 7.

Utilizzando il concetto di segnatura per formalizzare la nozione di tipo astratto, si introduce una nuova nozione: quando scegliamo una segnatura per un tipo astratto si evidenzia spesso una delle sorte, quella che denota il dominio di valori del tipo: tale sorta viene chiamata *sorta di interesse*.

Ora che abbiamo chiarito meglio cosa si intende per tipo astratto di dato, osserviamo che questa nozione è molto generale, e coglie in pieno il concetto di astrazione sui dati già discussa nel paragrafo 2.1.2. La nozione di tipo astratto consente di concettualizzare:

- i tipi di dato utilizzati nella progettazione di algoritmi e nella realizzazione dei corrispondenti programmi, quali array, liste, insiemi, pile, code, alberi, grafi ecc.; si usa spesso il termine *struttura di dati* per riferirsi a questi tipi di dato, per sottolineare il fatto che ad essi sono associati noti metodi di rappresentazione basati su strutture concrete quali vettori, record e puntatori, disponibili nei linguaggi di programmazione;
- classi di oggetti qualsiasi, che si ritengono importanti in una applicazione. Consideriamo, ad esempio, una applicazione in cui è rilevante la classe degli studenti iscritti ad un corso universitario. Possiamo pensare a questa classe come ad un tipo astratto che ha come dominio l'insieme degli studenti e come operazioni fondamentali, ad esempio: l'assegnazione di un numero di matricola, l'iscrizione ad un certo anno di corso, la scelta di un piano di studi, e così via.

È evidente che la possibilità di usare non solo tipi di dato consolidati e già ampiamente studiati, ma anche tipi astratti che noi come progettisti ideiamo e definiamo, evidenzia maggiormente l'importanza di indagare su come si specificano le caratteristiche di un tipo astratto nella fase di concettualizzazione. La definizione data sopra ci chiarisce che la specifica di un tipo deve fornire gli elementi per caratterizzare i domini e le funzioni del tipo. Inoltre, di ogni funzione occorre fornire altri elementi, quali i domini degli argomenti, il dominio del risultato, e anche una descrizione del suo significato.

Nel seguito, adotteremo un semplice metodo per fornire la specifica di un tipo, che verrà raffinato e formalizzato nel capitolo successivo. Secondo tale metodo, la specifica di un tipo T ha la seguente forma.

TipoAstratto T

Commento descrizione discorsiva del tipo astratto