

I. Algebre

I. Astrazione Funzionale

Un programma corrisponde alla **tripla** $\{D, A, R\}$.

Un programma **definisce un nuovo operatore** sui dati perché trasforma i dati iniziali in risultati; il repertorio di operatori può quindi essere ampliato scrivendo programmi.

L'**astrazione funzionale** è la tecnica che permette di potenziare il linguaggio disponibile introducendo nuovi operatori; questo viene fatto scrivendo funzioni (sottoprogrammi).

I **costrutti linguistici** per realizzare l'astrazione funzionale permettono di definire:

- **specifica**: definisce cosa ci si aspetta dalla funzione, cioè permette di capire cosa fa tramite l'intestazione e specifica cosa si aspetta in input e cosa restituisce in output;
- **realizzazione**: implementazione del comportamento della funzione.

II. Astrazione Dati

L'**astrazione dati** permette di ampliare il numero di tipi di dati disponibili, attraverso l'introduzione di nuovi dati e nuovi operatori.

L'astrazione dati consente di estendere l'**algebra** dei dati disponibile in un linguaggio di programmazione.

Un'**algebra** è un sistema matematico costituito da:

- un **dominio**, cioè un insieme di valori;
- una **insieme di funzioni** applicabili sui valori del dominio.

Allora la **corrispondenza tra algebra e tipo astratto** si basa sul fatto che entrambi hanno un dominio di definizione ed un insieme di operazioni lecite sul dominio.

2.1 - Requisiti dell'Astrazione Dati

Non tutti i linguaggi, però, permettono di definire dati astratti; alcuni permettono di definire solo nuovi tipi di dati, che non è la stessa cosa.

Si parla di **dato astratto** se le operazioni che possono essere effettuate sui rispettivi oggetti sono isolate dai dettagli usati per realizzare il tipo.

I **requisiti per l'astrazione** dati sono quindi:

- **Requisito di Astrazione**
Si deve poter dichiarare il dato astratto come ogni altro dato, indipendentemente dalla sua realizzazione.
- **Requisito di Protezione**
Gli operatori scritti per il dato astratto devono poter essere utilizzati solo su di esso.

2.2 - Specifica e Realizzazione per l'Astrazione Dati

Anche un'astrazione dati ha una **specifica** e una **realizzazione**.

La **specifica sintattica** descrive sinteticamente:

- i dati utilizzati per definire la struttura;
- gli operatori e i loro domini di partenza e di arrivo, cioè i tipi di dati richiesti in input e quelli restituiti in output.

La **specifica semantica** definisce invece:

- un insieme ad ogni nome introdotto nella specifica sintattica;
- un valore ad ogni costante;
- una funzione ad ogni nome di operatore, specificando:
 - **pre-condizione**, che definisce quando l'operatore è utilizzabile;
 - **post-condizione**, che stabilisce come il risultato sia vincolato agli argomenti dell'operatore.

Esprimere specifiche semantiche formalmente risulta difficile, perciò solitamente si utilizza un linguaggio naturale o matematico.

La **realizzazione** descrive invece come dati e operatori vengono implementati usando dati e operatori già esistenti. Le decisioni in fase di realizzazione dipendono dal linguaggio.

2.3 - Algebre

Un'**algebra** dei dati è composta da:

1. **Insieme di dati**

Degli esempi di insiemi di dati sono interi, boolean, stringhe etc.

2. **Operatori**

Gli operatori possono essere aritmetici, logici, di confronto e di concatenazione.

3. **Nomi per indicare l'insieme di dati**

E necessario un nome per identificare l'insieme, come accade per gli interi, identificati da *integer* o per i booleani, identificati da *bool*.

4. **Nomi per indicare gli operatori**

E possibile utilizzare nomi e simboli per funzioni come, ad esempio, + per la somma, - per la sottrazione, *concat* per la concatenazione.

5. **Costanti per indicare elementi singoli degli insiemi di dati**

Queste costanti sono necessarie per indicare univocamente un dato, come, ad esempio, le stringhe che vengono scritte tra virgolette.

III. Strutture Dati

Come detto in precedenza, il dato è una struttura matematica che consiste di un dominio sul quale sono ammesse alcune funzioni.

Una **struttura dati** è un particolare tipo di dato caratterizzato dall'organizzazione degli elementi al suo interno (più che dal tipo).

Le strutture solitamente disponibili nei linguaggi sono gli array.

Abbiamo diversi **tipi** di strutture dati:

- **lineari**: dati disposti in sequenza;
- **statiche**: in cui il numero di elementi non può variare nel tempo;
- **omogenee**: dati dello stesso tipo
- **non lineari**: nessuna sequenza specifica;
- **dinamiche**: in cui il numero di elementi può variare nel tempo;
- **non omogenee**: dati non dello stesso tipo

IV. Tecniche di Specifica

Due **tecniche per la scrittura di specifiche** sono:

- **Specifiche Assiomatiche (o Algebriche)**: queste si dicono self-contained, cioè specificano ogni oggetto come composizione di funzioni;
- **Modelli Astratti (approccio costruttivo)**: definiscono la semantica delle operazioni in termini di un altro tipo di dato ben definito.

La chiave di queste e di altre tecniche di specifica sta nel fatto che la descrizione della semantica non fa riferimento alla realizzazione.

4.1 - Specifica di un Dato Astratto

Possiamo definire formalmente una **specificazione di un tipo di dato astratto** come una **tripla (D, F, A)**:

- **D: insieme di tutti i domini usati per la definire il dato**
Se stiamo definendo una struttura, essa stessa sarà il **dominio designato**, tutti gli altri, usati per la sua implementazione, saranno i **domini ausiliari**.
- **F: insieme degli operatori**
- **A: insieme delle regole che descrivono la semantica degli operatori**
Devono definire le caratteristiche della struttura, ad esempio la proprietà LIDO dello stack etc.

4.2 - Tipi di Operatori

Tra i **diversi tipi di operatori** abbiamo:

Operatori di Base

- **Costruttori**
Inizializzano una nuova istanza del dato.
- **Modificatori**
Cambiano il dato in qualche modo.
- **Osservatori**
Osservano lo stato senza modificarlo (es. *isEmpty*).

Operatori Aggiuntivi

- **Distruttori**
Liberano la memoria occupata dal dato.
- **Iteratori**
Permettono di iterare nella struttura componente per componente.

specifica assiomatica e costruttiva