Corso di Laurea in INFORMATICA

Algoritmi e Strutture Dati

MODULO 2 - corsi A, B, C – a.a. 2002-2003

Algebre di dati

Dati e rappresentazioni, requisiti delle astrazioni di dati, costrutti. Astrazioni di dati e dati primitivi. Specifica sintattica e semantica. La realizzazione.

Questi lucidi sono stati preparati da Oronzo Altamura, Maria Francesca Costabile e Floriana Esposito, Università degli Studi di Bari, per uso didattico. Essi contengono materiale originale di proprietà dell'Università degli Studi di Bari e/o figure di proprietà di altri autori, società e organizzazioni di cui e' riportato il riferimento. Tutto o parte del materiale può essere fotocopiato per uso personale o didattico ma non può essere distribuito per uso commerciale. Qualunque altro uso richiede una specifica autorizzazione da parte dell'Università degli Studi di Bari e degli altri autori coinvolti.

UN PROGRAMMA CORRISPONDE ALLA TRIPLA

{ D , A , R }

SI PUÒ DIRE CHE, POICHÉ TRASFORMA DATI INIZIALI IN RISULTATI, DEFINISCE UN NUOVO *OPERATORE* SUI DATI.

IL REPERTORIO DI OPERATORI DISPONIBILI SUI DATI PUÒ ESSERE AMPLIATO SCRIVENDO PROGRAMMI.

L'ASTRAZIONE FUNZIONALE È LA TECNICA CHE PERMETTE DI POTENZIARE IL LINGUAGGIO DISPONIBILE INTRODUCENDO *NUOVI OPERATORI.* QUESTI SONO DEFINITI ATTRAVERSO SOTTO PROGRAMMI IN CUI SI FA USO DEGLI OPERATORI BASE, GIÀ DISPONIBILI.

L'INTESTAZIONE DEL SOTTOPROGRAMMA INTRODUCE IL NOME E GLI ARGOMENTI DEL SOTTOPROGRAMMA MENTRE IL CORPO NE DESCRIVE LE REGOLE DI COMPORTAMENTO.

2

CONSIDERIAMO LA FORMULA PER IL CALCOLO DEL COEFFICIENTE BINOMIALE

$n, m \in \mathbb{N} \quad (n > m)$

$$\binom{n}{m} = n! / (m! (n - m)!)$$

CHE FORNISCE IL NUMERO DELLE POSSIBILI COMBINAZIONI DI N OGGETTI PRESI A GRUPPI DI M.

IL CALCOLO È IMMEDIATO SE SI SUPPONE DI AVERE UN OPERATORE FATT(K) CHE CALCOLA IL FATTORIALE K! PER UN GENERICO VALORE K

$$K! = K \cdot (K-1) \cdot (K-2) \cdot \ldots\ldots \cdot 3 \cdot 2 \cdot 1$$

3

SI USERA' DA QUI IN POI UNA NOTAZIONE IN UNO PSEUDO LINGUAGGIO DI PROGRAMMAZIONE SIMILE AL PASCAL (Pascalike)

```
PROGRAM CBIN;
        VAR N,M,R:INTEGER;
        BEGIN
            R:= FATT(N)/(FATT(M) • FATT(N-M));
        END;
```

SE FATT(K) NON È DISPONIBILE È NECESSARIO SCRIVERE UN SOTTOPROGRAMMA PER AMPLIARE IL REPERTORIO DEGLI OPERATORI.

L'ASTRAZIONE FUNZIONALE CI CONSENTE QUESTO.

4

I LINGUAGGI DI PROGRAMMAZIONE AD ALTO LIVELLO PERMETTONO L'USO DI

ASTRAZIONE FUNZIONALE

CIOÈ CONSENTONO DI

CREARE DELLE UNITÀ DI PROGRAMMA DANDO UN NOME AD UN GRUPPO DI ISTRUZIONI E STABILISCONO LE MODALITÀ DI COMUNICAZIONE TRA L'UNITÀ DI PROGRAMMA CREATA ED IL RESTO DEL PROGRAMMA IN CUI ESSA SI INSERISCE.

AL MOMENTO DELLA ATTIVAZIONE (SU CHIAMATA) DELLA UNITÀ DI PROGRAMMA VIENE SOSPESA L'ESECUZIONE DEL PROGRAMMA (O DELL'UNITÀ) CHIAMANTE: IL CONTROLLO PASSA ALLA UNITÀ ATTIVATA E, ALL'ATTO DEL COMPLETAMENTO DELLA SUA ESECUZIONE, L'ATTIVAZIONE TERMINA E IL CONTROLLO TORNA AL PROGRAMMA CHIAMANTE.

5

I COSTRUTTI LINGUISTICI PER REALIZZARE LA ASTRAZIONE FUNZIONALE CONSENTONO DI DEFINIRE

UNA SPECIFICA E UNA REALIZZAZIONE

LA SPECIFICA DEFINISCE "COSA" CI SI ASPETTA DALLA FUNZIONE CIOÈ DEFINIRÀ

"LA RELAZIONE CHE CARATTERIZZA IL LEGAME TRA DATI DI INGRESSO E RISULTATI"

NEL CASO DEL FATTORIALE LA SPECIFICA DIRA' CHE:

1. L'INTESTAZIONE È
   FUNCTION FATT(K:INTEGER): INTEGER;
   L'INGRESSO K (argomento) E' UNA VARIABILE DI TIPO INTERO; IL RISULTATO E' ANCORA DI TIPO INTERO E SARA' ASSOCIATO ALLA VARIABILE DI NOME FATT
1. LA FUNZIONE DA CALCOLARE E' DEFINITA COME
   $FATT(K)=K \cdot (K-1) \cdot \ldots\ldots\ldots \cdot 2 \cdot 1 \qquad K \geq 1$

6

LA REALIZZAZIONE DEFINISCE “COME” IL RISULTATO VIENE OTTENUTO.

ALLA STESSA SPECIFICA, INFATTI, POTRANNO CORRISPONDERE SCELTE DI REALIZZAZIONE DIVERSE.

LA FUNZIONE FATT POTREBBE ESSERE REALIZZATA CON UN PROGRAMMA ITERATIVO

```
FUNCTION FATT(K: INTEGER): INTEGER;
        VAR I,F : INTEGER ;
BEGIN
        IF (K = 1) THEN F:=1  ELSE
        FOR I:= 2 TO K  DO F :=F * I ;
 FATT := F ;
END;
```

7

MA E’ ANCHE POSSIBILE REALIZZARLA CON UN PROGRAMMA RICORSIVO

```
FUNCTION FATT(K: INTEGER): INTEGER;
        VAR F : INTEGER ;
BEGIN
        IF (K = 1) THEN F:=1  ELSE
        F:= K * FATT(K-1);
 FATT := F ;
END;
```

8

LA ASTRAZIONE DI DATI RICALCA ED ESTENDE IL CONCETTO DI ASTRAZIONE FUNZIONALE.

COSI’ COME L'ASTRAZIONE FUNZIONALE PERMETTE DI AMPLIARE L'INSIEME DEI MODI DI OPERARE SUI DATI, CIOÈ GLI OPERATORI SUI TIPI DI DATI GIÀ DISPONIBILI, LA ASTRAZIONE DI DATI PERMETTE DI AMPLIARE I TIPI DI DATI DISPONIBILI ATTRAVERSO L'INTRODUZIONE SIA DI NUOVI TIPI DI DATI CHE DI NUOVI OPERATORI.

L’ASTRAZIONE FUNZIONALE STIMOLA GLI SFORZI PER EVIDENZIARE OPERAZIONI RICORRENTI O BEN CARATTERIZZATE ALL’INTERNO DELLA SOLUZIONE DI UN PROBLEMA

L’ASTRAZIONE DI DATI SOLLECITA AD INDIVIDUARE LE ORGANIZZAZIONI DEI DATI PIÙ ADATTE ALLA SOLUZIONE DEL PROBLEMA

9

UN'ASTRAZIONE DI DATI CONSENTE UNA ESTENSIONE DELL’ALGEBRA DEI DATI DISPONIBILE IN UN LINGUAGGIO DI PROGRAMMAZIONE.

COSA E’ UN’ALGEBRA? È UN SISTEMA MATEMATICO COSTITUITO DA UN DOMINIO, CIOÈ UN INSIEME DI VALORI. E DA UN INSIEME DI FUNZIONI APPLICABILI SU TALI VALORI.

IL NUMERO, IL TIPO E LE PROPRIETÀ DELLE FUNZIONI SONO GLI ELEMENTI ESSENZIALI DI UN ALGEBRA.

LA CORRISPONDENZA TRA TIPO ASTRATTO E ALGEBRA SI BASA SUL PARALLELO TRA GLI INSIEMI DI VALORI DI UN ALGEBRA E I DOMINI DI DEFINIZIONE DI UN TIPO ASTRATTO, E TRA LE FUNZIONI DI UN’ALGEBRA E LE OPERAZIONI ASSOCIATE AD UN TIPO ASTRATTO.

10

MA COME ESTENDERE L’ALGEBRA DEI DATI DISPONIBILE IN UN LINGUAGGIO DI PROGRAMMAZIONE?

MOLTI LINGUAGGI DI PROGRAMMAZIONE FORNISCONO I MEZZI PER DEFINIRE E CREARE NUOVI TIPI DI DATI, MA NON TUTTI CONSENTONO DI FARE ASTRAZIONE DATI O CREARE TIPI DI DATI ASTRATTI.

UN TIPO DI DATI SI DICE ASTRATTO SE LE OPERAZIONI APPLICABILI SUGLI OGGETTI CHE LI RAPPRESENTANO SONO ISOLATE DAI DETTAGLI USATI PER REALIZZARE IL TIPO.

11

UN ‘ALGEBRA DEI DATI PUÒ ESSERE DEFINITA COME COSTITUITA DA:

1. UNA FAMIGLIA DI INSIEMI (INSIEME DI DATI)
2. UNA FAMIGLIA DI OPERATORI SUI DATI (OPERATORI)
3. UN REPERTORIO DI SIMBOLI (O NOMI) PER INDICARE L’INSIEME DI DATI
4. UN REPERTORIO DI SIMBOLI (O NOMI) PER INDICARE GLI OPERATORI
5. UN REPERTORIO DI SIMBOLI (DETTI COSTANTI) PER INDICARE ELEMENTI SINGOLI DEGLI INSIEMI DI DATI

12

Esempio-1 di Algebra di dati

1. LA FAMIGLIA DI INSIEMI COMPRENDE:
I NUMERI INTERI, I BOOLEANI, LE STRINGHE (SEQUENZE DI CARATTERI ALFANUMERICI).

2. LA FAMIGLIA DI OPERATORI COMPRENDE:
- OPERATORI ARITMETICI (SOMMA, SOTTRAZIONE, MOLTIPLICAZIONE E DIVISIONE)
- OPERATORI DI CONGIUNZIONE, DISGIUNZIONE E NEGAZIONE PER I BOOLEANI
- OPERATORI DI CONFRONTO TRA INTERI (MAGGIORE, MINORE, UGUALE ETC.)
- OPERATORI DI CONCATENAZIONE PER LE SEQUENZE DI CARATTERI E DI SELEZIONE DI SOTTOSTRINGA (QUESTO RESTITUISCE COME RISULTATO LA PARTE DI STRINGA COMPRESA TRA DUE NUMERI INTERI CHE INDICANO LA POSIZIONE DEL PRIMO E DELL'ULTIMO CARATTERE).

13

3. IL REPERTORIO DEI SIMBOLI O NOMI PER L'INSIEME DI DATI POTREBBE ESSERE:
INTEGER, BOOLEAN, STRING

4. IL REPERTORIO DI SIMBOLI O NOMI PER GLI OPERATORI POTREBBE ESSERE:

+ , -, * , /	PER GLI OPERATORI ARITMETICI
AND , OR , NOT	PER GLI OPERATORI BOOLEANI
> , < , = , ¹ , £ , ³	PER I CONFRONTI TRA INTERI
SUBSTR E &	PER LA SELEZIONE E LA CONCATENAZIONE DI STRINGHE

14

5. GLI ELEMENTI SINGOLI PER GLI INSIEMI DI DATI (LE COSTANTI) POTREBBERO ESSERE:

- OGNI INTERO VIENE INDICATO A UNA SEQUENZA DI CIFRE DECIMALI, EVENTUALMENTE PRECEDUTA DA + O - , INTERPRETATA SECONDO LA NOTAZIONE DECIMALE;
- I VALORI BOOLEANI SONO INDICATI DA TRUE E FALSE.
- OGNI SEQUENZA DI CARATTERI VIENE INDICATA DALLA SEQUENZA RACCHIUSA TRA APICI;

15

GLI OPERATORI POSSONO ESSERE COMBINATI PER OTTENERE DELLE ESPRESSIONI.

LE CARATTERISTICHE DI UN'ALGEBRA DIPENDONO DALLA FAMIGLIA DI INSIEMI E DALLA FAMIGLIA DI OPERATORI. NEI LINGUAGGI DI PROGRAMMAZIONE IL BINOMIO

DATI – OPERATORI

È INSCINDIBILE.

DUNQUE, È NECESSARIO DEFINIRE UN TIPO DI DATI COME UN INSIEME DI DATI PIÙ GLI OPERATORI CHE AD ESSO SI RIFERISCONO, NEL CONTESTO DI UNA DEFINITA ALGEBRA.

16

ANCHE UN'ASTRAZIONE DI DATI È COSTITUITA DA :

UNA SPECIFICA :
HA IL COMPITO DESCRIVERE SINTETICAMENTE IL TIPO DEI DATI E GLI OPERATORI CHE LI CARATTERIZZANO

UNA REALIZZAZIONE :
STABILISCE COME I DATI E GLI OPERATORI VENGONO RICONDOTTI AI TIPI DI DATI E AGLI OPERATORI GIÀ DISPONIBILI

L'ASTRAZIONE DI DATI STIMOLA GLI SFORZI PER INDIVIDUARE LE ORGANIZZAZIONE DI DATI PIÙ ADATTE ALLA SOLUZIONE DI UN PROBLEMA MENTRE LA ASTRAZIONE FUNZIONALE PORTA A EVIDENZIARE OPERAZIONI RICORRENTI O FUNZIONALITÀ CARATTERIZZATE ALL'INTERNO DI UN PROBLEMA.

LA SCELTA DELLE STRUTTURE DATI E' IL PRIMO PASSO PER UN BUON RISULTATO NELL'ATTIVITA' DI PROGRAMMAZIONE

17

OBIETTIVO

ATTUARE UNA CORRETTA ASTRAZIONE DI DATI

BISOGNA IDENTIFICARE I COSTRUTTI DI PROGRAM-MAZIONE CHE LA CONSENTONO

NON E' SUFFICIENTE USARE SOTTOPROGRAMMI CHE IMPLEMENTANO I NUOVI OPERATORI

18

Esempio-2:

SUPPONIAMO DI VOLER EFFETTUARE UN'ASTRAZIONE DI DATI AGGIUNGENDO AI TIPI DI DATI STANDARD I NUMERI COMPLESSI.

COME E' NOTO I NUMERI COMPLESSI HANNO FORMA:

$$r_1 + i\,r_2$$

DOVE r_1 , r_2 SONO NUMERI REALI

E i È $\sqrt{-1}$

PER DENOTARLI MATEMATICAMENTE BASTA LA COPPIA

$$(r_1 , r_2)$$

DOVE r_1 E' IL COEFFICIENTE DELLA PARTE REALE E r_2 QUELLO DELLA PARTE IMMAGINARIA.

È NECESSARIO DEFINIRE OPERATORI DI SELEZIONE E COSTRUZIONE CHE OPERANO SU COMPLESSI E SU REALI, OPERATORI DI SOMMA E PRODOTTO CHE OPERANO ESCLUSIVAMENTE SUI COMPLESSI

19

Esempio-2 (cont.):

IL CALCOLO DI SOMMA E PRODOTTO SU NUMERI COMPLESSI NON È DIFFICILE SE RICORDIAMO CHE

LA SOMMA DI DUE NUMERI COMPLESSI C_1 E C_2 DÀ LUOGO AD UN NUOVO NUMERO COMPLESSO C_3 LA CUI PARTE REALE È LA SOMMA DELLE PARTI REALI E LA CUI PARTE IMMAGINARIA È LA SOMMA DELLE PARTI IMMAGINARIE.

PARTE REALE(C_3) = PARTE REALE (C_1)+PARTE REALE(C_2)

PARTE IMMG(C_3) = PARTE IMMG(C_1) + PARTE IMMG(C_2)

20

Esempio-2 (cont.):

IL PRODOTTO DI DUE NUMERI COMPLESSI C_1 E C_2 DÀ LUOGO AD UN NUOVO NUMERO COMPLESSO C_4:

LA CUI PARTE REALE È LA DIFFERENZA TRA IL PRODOTTO DELLA PARTE REALE DEL 1° NUMERO PER LA PARTE REALE DEL 2° NUMERO E IL PRODOTTO DELLA PARTE IMMAGINARIA DEL 1° NUMERO PER LA PARTE IMMAGINARIA DEL 2° NUMERO.

PARTE REALE(C4) = PARTE REALE (C1)*PARTE REALE(C2) -
PARTE IMMG(C1) * PARTE IMMG(C2)

E LA CUI PARTE IMMAGINARIA È LA SOMMA DEL PRODOTTO DELLA PARTE REALE DEL 1° NUMERO PER LA PARTE IMMAGINARIA DEL 2° NUMERO CON IL PRODOTTO DELLA PARTE IMMAGINARIA DEL 1° NUMERO CON LA PARTE REALE DEL 2° NUMERO

PARTE IMMG(C4) = PARTE REALE (C1)* PARTE IMMG(C2)+
PARTE IMMG(C1) * PARTE REALE(C2)

21

Esempio-2 (cont.):

PASSANDO AGLI OPERATORI NECESSARI, ABBIAMO BISOGNO DI DEFINIRE, ATTRAVERSO PROCEDURE E FUNZIONI:

CXSUM (C_1 , C_2) = C_3 SOMMA DI COMPLESSI

CXPROD (C_1 , C_2) = C_4 PRODOTTO DI COMPLESSI

REALPART (C) = R' SELEZIONA PARTE REALE DEL COMPLESSO C

IMMAPART (C) = I' SELEZIONA PARTE IMMAGINARIA DEL COMPLESSO C

CONSCX (R_1, R_2) = C' COMBINAZIONE DI DUE REALI, IL PRIMO COME PARTE REALE, IL SECONDO COME IMMAGINARIA DEL COMPLESSO C'

22

Esempio-2 (cont.):

PER POTER FARE ASTRAZIONE DATI, BASTA DISPORRE DI LINGUAGGI CHE CONSENTONO ASTRAZIONE FUNZIONALE?

CON IL FORTRAN, AD ESEMPIO, SI PUÒ FARE ASTRAZIONE FUNZIONALE CON L'USO DI SUBROUTINE, MA NON È POSSIBILE FARE ASTRAZIONE DATI.

I NUMERI COMPLESSI SONO RAPPRESENTATI MATEMATICAMENTE COME COPPIE (r_1 , r_2): I VETTORI POSSONO ESSERE USATI A TALE SCOPO.

```
DIMENSION C1(2), C2(2), C3(2), C4(2)
```

E' LA DICHIARATIVA FORTRAN PER I VETTORI CHE RAPPRESENTANO I NUMERI COMPLESSI

LA SOMMA PUÒ ESSERE REALIZZATA CON UN SOTTOPROGRAMMA

```
SUBROUTINE SOMMACOMPLESSI (C1(2), C2(2), C3(2))
```

23

Esempio-2 (cont.):

LA STESSA COSA PUÒ FARSI PER GLI ALTRI OPERATORI.

VA, PERÒ, EVIDENZIATO CHE INPUT A QUESTA SUBROUTINE POSSONO ESSERE TUTTI I VETTORI DI REALI A DUE ELEMENTI, ANCHE QUELLI CHE NON HANNO NULLA A CHE VEDERE CON LA NOSTRA ASTRAZIONE.

NON VI È NESSUN MODO, IN QUESTO TIPO DI LINGUAGGIO, DI CONSENTIRE CHE SOLO

I DATI ASTRATTI DEFINITI

ABBIANO L'ACCESSO AGLI OPERATORI APPOSITAMENTE COSTRUITI (NEL NOSTRO CASO AI VETTORI CHE RAPPRESENTANO I COMPLESSI)

IL DISCORSO È DIVERSO SE HO A DISPOSIZIONE UN LINGUAGGIO A TIPIZZAZIONE FORTE COME IL PASCAL

24

Esempio-2 (cont.):

IN PASCAL SI PUÒ DICHIARARE

```
TYPE
      COMPLEX= ARRAY [2] OF REAL;
VAR
     C1, C2, C3, C4 :COMPLEX;
     .
     .
     .
PROCEDURE     SOMMACOMPLESSI (C1, C2:COMPLEX;
                              VAR C3 :COMPLEX);
                  . . . . .
                  . . . . .
```

25

Esempio-2 (cont.):

DI FATTO IN PASCAL POTREMMO ANCHE REALIZZARE LA ASTRAZIONE IN ALTRO MODO

```
TYPE
      COMPLEX= RECORD OF
                  PARTE REALE :REAL;
                  PARTE IMMAGINARIA :REAL;
                  END;
VAR
     C1, C2, C3, C4 :COMPLEX
     .
     .
PROCEDURE     SOMMACOMPLESSI (C1, C2:COMPLEX;
                              VAR C3:COMPLEX);
                  . . . . .
                  . . . . .
```

26

COME SI PUÒ NOTARE, DISPONENDO DI UN LINGUAGGIO CON TIPI, POSSIAMO:

- UTILIZZARE DATI CHE DEFINIAMO E DICHIARIAMO DIRETTAMENTE COME COMPLESSI, PRESCINDENDO DALLA EFFETTIVA REALIZZAZIONE
- FARE IN MODO CHE ALLE PROCEDURE (OPERATORI) COSTRUITE PER I NOSTRI DATI ABBIANO ACCESSO ESCLUSIVAMENTE QUEI DATI

QUESTE CARATTERISTICHE NECESSARIE A UNA BUONA ASTRAZIONE DI DATI SONO NOTE COME

I REQUISITI DELLA ASTRAZIONE DI DATI

27

I REQUISITI DELLA ASTRAZIONE DI DATI

<REQUISITO DI ASTRAZIONE>

È VERIFICATO QUANDO I PROGRAMMI CHE USANO UN'ASTRAZIONE POSSONO ESSERE SCRITTI IN MODO DA NON DIPENDERE DALLE SCELTE DI REALIZZAZIONE.

NELL' ESEMPIO-2, IN PASCAL LE DIVERSE REALIZZAZIONI NON PROVOCANO CAMBI NELLE INTESTAZIONI DEI SOTTOPROGRAMMI CHE REALIZZANO GLI OPERATORI.

LA MANCANZA DEL REQUISITO DI ASTRAZIONE

SI MANIFESTA CON L'IMPOSSIBILITÀ DI DICHIARARE "DIRETTAMENTE" VARIABILI CON IL NUOVO TIPO DEFINITO.

NON POSSO, NEL CODICE, FARE RIFERIMENTO AI NUOVI DATI CREATI PRESCINDENDO DALLA RAPPRESENTAZIONE USATA E DALLA REALIZZAZIONE.

28

I REQUISITI DELLA ASTRAZIONE DI DATI

<REQUISITO DI PROTEZIONE >

È VERIFICATO SE SUI NUOVI DATI SI PUÒ LAVORARE ESCLUSIVAMENTE CON GLI OPERATORI DEFINITI ALL'ATTO DELLA SPECIFICA.

LA MANCANZA DEL REQUISITO DI PROTEZIONE

SI MANIFESTA CON LA POSSIBILITA' DI LAVORARE CON GLI OPERATORI DEFINITI PER I NUOVI DATI ANCHE SU DATI CON RAPPRESENTAZIONI SIMILI, MA NON OMOGENEI PER TIPO.

NELL' ESEMPIO APPENA VISTO, RELATIVO AI NUMERI COMPLESSI, IN FORTRAN, IL TIPO DI DICHIARATIVE E LA REALIZZAZIONE CON VETTORI E SUBROUTINE CONSENTE L'ACCESSO AGLI OPERATORI DA PARTE DI QUALUNQUE VETTORE DI REALI

29

I REQUISITI DELLA ASTRAZIONE DI DATI

IL REQUISITO DI ASTRAZIONE, IN ACCORDO CON I PRINCIPI DELLA ASTRAZIONE DI DATI, IMPONE CHE I PROGRAMMI SIANO SENSIBILI SOLO A VARIAZIONI DELLA SPECIFICA DEI NUOVI TIPI.

IL REQUISITO DI PROTEZIONE È PARTICOLARMENTE IMPORTANTE, SE VOGLIAMO CHE L'ESECUTORE DEL LINGUAGGIO ASSICURI I CONTROLLI DI CONSISTENZA TRA I TIPI DI DATI E GLI OPERATORI, NON SOLO AI DATI PRIMITIVI, MA ANCHE AI DATI OTTENUTI PER ASTRAZIONE.

30

SI E' DETTO CHE UN'ASTRAZIONE DI DATI È COSTITUITA DA :
UNA SPECIFICA E UNA REALIZZAZIONE

UNA SPECIFICA, CHE HA IL COMPITO DESCRIVERE SINTETICAMENTE IL TIPO DEI DATI E GLI OPERATORI CHE LI CARATTERIZZANO, SI DISTINGUE IN:

SPECIFICA SINTATTICA CHE FORNISCE :
- L'ELENCO DEI NOMI DEI TIPI DI DATO UTILIZZATI PER DEFINIRE LA STRUTTURA, DELLE OPERAZIONI SPECIFICHE DELLA STRUTTURA E DELLE COSTANTI
- I DOMINI DI PARTENZA E DI ARRIVO, CIOÈ I TIPI DEGLI OPERANDI E DEL RISULTATO PER OGNI NOME DI OPERATORE

31

SPECIFICA SEMANTICA CHE DEFINISCE IL SIGNIFICATO DEI NOMI INTRODOTTI CON LA SPECIFICA SINTATTICA. ASSOCIA:
- UN INSIEME AD OGNI NOME DI TIPO INTRODOTTO NELLA SPECIFICA SINTATTICA
- UN VALORE AD OGNI COSTANTE
- UNA FUNZIONE AD OGNI NOME DI OPERATORE ESPLICITANDO LE SEGUENTI CONDIZIONI SUI DOMINI DI PARTENZA E DI ARRIVO:
  1) PRECONDIZIONE CHE DEFINISCE QUANDO L'OPERATORE È APPLICABILE
  2) POSTCONDIZIONE CHE STABILISCE COME IL RISULTATO SIA VINCOLATO AGLI ARGOMENTI DELL'OPERATORE

LA REALIZZAZIONE SFRUTTA AL MEGLIO LE POSSIBILITA' OFFERTE DALL'AMBIENTE DI PROGRAMMAZIONE

32

LA DEFINIZIONE DI UN FORMALISMO PER LE SPECIFICHE SINTATTICHE NON PONE PARTICOLATI PROBLEMI

L'INDIVIDUAZIONE DI FORMALISMI PER LE SPECIFICHE SEMANTICHE E' ASSAI PIU' DIFFICILE

IN GENERE, LE SPECIFICHE SEMANTICHE SONO DATE CON FRASI ESTRATTE DAL LINGUAGGIO NATURALE O DA QUELLO MATEMATICO

33

Esempio-3: I NUMERI INTERI E I BOOLEANI

SPECIFICA SINTATTICA

TIPI :
INTEGER, BOOLEAN

OPERATORI :

+ , -	:	(INTEGER, INTEGER) ® INTEGER
< , >	:	(INTEGER, INTEGER) ® BOOLEAN
AND, OR	:	(BOOLEAN, BOOLEAN) ® BOOLEAN
NOT	:	(BOOLEAN) ® BOOLEAN
SEQCIF	:	( ) ® INTEGER SEQUENZA DI CIFRE DECIMALI CON O SENZA SEGNO
TRUE	:	( ) ® BOOLEAN
FALSE	:	( ) ® BOOLEAN

34

SPECIFICA SEMANTICA

TIPI :
INTEGER : L'INSIEMI DEI NUMERI INTERI
BOOLEAN : L'INSIEME DEI VALORI DI VERITÀ

OPERATORI :
+ E - : FORNISCONO COME RISULTATO SOMME E DIFFERENZE
< E > : DANNO VERO SE IL 1° OPERATORE È MINORE(MAGGIORE) DEL 2°
AND E OR : FORNISCONO LA CONGIUNZIONE E LA DISGIUNZIONE LOGICHE DEI DUE OPERATORI
NOT : FORNISCE LA NEGAZIONE LOGICA DELL'OPERATORE

35

SEQCIF : FORNISCE L'INTERO SECONDO NOTAZIONE DECIMALE
TRUE : CORRISPONDE AL VALORE DI VERITÀ VERO
FALSE : CORRISPONDE AL VALORE DI VERITÀ FALSO

IN QUESTA SPECIFICA IL TIPO DI DATO INTEGER POSSIEDE UN NUMERO INFINITO DI COSTANTI (OGNI POSSIBILE SEQUENZA DI CIFRE DECIMALI È UNA COSTANTE)

MA VI E' UN SOLO MODO DI DEFINIRE UNA SPECIFICA?

L'INSIEME DEGLI OPERATORI DEFINIBILI SU INSIEMI DI DATI E'UNICO?

36

AVREMMO POTUTO UTILIZZARE UN'ALTRA SPECIFICA.

SPECIFICA SINTATTICA

TIPI :

INTEGER, BOOLEAN

OPERATORI :

ZERO : ( ) → INTEGER

TRUE : ( ) → BOOLEAN

FALSE : ( ) → BOOLEAN

SUCC : (INTEGER) → INTEGER

PRED : (INTEGER) → INTEGER

ISZERO : (INTEGER) → BOOLEAN

37

SPECIFICA SEMANTICA

TIPI :

INTEGER : L'INSIEMI DEI NUMERI INTERI

BOOLEAN : L'INSIEME DEI VALORI DI VERITÀ

OPERATORI :

ZERO : DÀ LO ZERO DEI NUMERI INTERI

TRUE : DÀ IL VALORE DI VERITÀ VERO

FALSE : DÀ IL VALORE DI VERITÀ FALSO

SUCC (n) : DÀ n+1

PRED (n) : DÀ n – 1

ISZERO(n) : DÀ VERO SE E SOLO SE N È LO ZERO DEGLI INTERI

38

LA REALIZZAZIONE

RICONDUCE LA SPECIFICA AI TIPI DI DATO PRIMITIVI E AGLI OPERATORI GIÀ DISPONIBILI; IN PARTICOLARE GLI OPERATORI SONO SIMULATI CON SOTTOPROGRAMMI.

ALLA STESSA SPECIFICA POSSONO CORRISPONDERE DIVERSE REALIZZAZIONI PIÙ O MENO EFFICIENTI. LE REALIZZAZIONI POSSONO ESSERE TRASPARENTI ALL'UTENTE.

NEL VALUTARE L'EFFICIENZA DI SOTTOPROGRAMMI CHE UTILIZZANO TIPI DI DATO PRIMITIVI, SI PRESCINDE DALLE CARATTERISTICHE DELLA MACCHINA, ASSUMENDO DI FAR RIFERIMENTO AD UNA ORGANIZZAZIONE MOLTO SEMPLICE

39

SI FA RIFERIMENTO A:

- DATI CONTENUTI IN MEMORIA
- MEMORIA ORGANIZZATA IN CELLE (PAROLA)
- CELLE ACCESSIBILI TRAMITE INDIRIZZO
- INDIRIZZI CHE SONO INTERI CONSECUTIVI
- DATI ELEMENTARI CHE NON DECIDONO LA CAPACITÀ DI UNA CELLA (MAXINT)
- VARIABILI CONTENUTE IN CELLE DI INDIRIZZO NOTO IL CUI ACCESSO RICHIEDE TEMPO COSTANTE.

40

UN COSTRUTTO DI PROGRAMMAZIONE PROPRIAMENTE ADATTO ALLA REALIZZAZIONE DI

ASTRAZIONE DI DATI

DOVRÀ :

•DICHIARARE ESPLICITAMENTE QUALI SONO I NUOVI TIPI DI DATI E QUALI SONO I NUOVI OPERATORI

•DEFINIRE LA RAPPRESENTAZIONE DEI NUOVI DATI IN TERMINI DI DATI PRE-ESISTENTI

•CONTENERE UN INSIEME DI SOTTOPROGRAMMI CHE REALIZZINO GLI OPERATORI DEFINITI SUI NUOVI DATI

41

COME E' NOTO, IL TIPO DI DATO È UN MODELLO MATEMATICO CHE CONSISTE NELLA DEFINIZIONE DI UNA COLLEZIONE DI VALORI (dominio) SUI QUALI SONO AMMESSE CERTE OPERAZIONI (funzioni).

UNA STRUTTURA DI DATI È UN PARTICOLARE TIPO DATO, CARATTERIZZATO PIÙ DALLA ORGANIZZAZIONE IMPOSTA AGLI ELEMENTI CHE LA COMPONGONO CHE DAL TIPO DEGLI ELEMENTI COMPONENTI.

LE STRUTTURE SOLITAMENTE DISPONIBILI NEI LINGUAGGI DI PROGRAMMAZIONE SONO LE TABELLE MONODIMENSIONALI (VETTORI, ARRAY) I CUI ELEMENTI SONO CONTENUTI IN MEMORIA CENTRALE IN LOCAZIONI CONTIGUE

42

IN GENERALE, DISTINGUIAMO STRUTTURE:

- LINEARI : SEQUENZE
- NON LINEARI : IN CUI NON È INDIVIDUATA UNA SEQUENZA
- A DIMENSIONE FISSA : IN CUI IL NUMERO DI ELEMENTI NON PUÒ VARIARE NEL TEMPO
- A DIMENSIONE VARIABILE : IN CUI IL NUMERO DI ELEMENTI PUÒ VARIARE NEL TEMPO

43

LA CLASSICA STRUTTURA VETTORE (ARRAY) E' A DIMENSIONE FISSA.

È UNA TABELLA MONODIMENSIONALE DI ELEMENTI OMOGENEI, SU CUI POSSONO EFFETTUARSI OPERAZIONI DI:

LETTURA (O SELEZIONE) REPERIMENTO DEL VALORE DI UN ELEMENTO

SCRITTURA (O SOSTITUZIONE) DI UN VALORE DI UN ELEMENTO CON UN NUOVO VALORE

UN VETTORE DI N COMPONENTI È INTESO COME SEQUENZA DI OGGETTI IN RELAZIONE D'ORDINE TRA LORO. L'ORGANIZZAZIONE IN MEMORIA CONSENTE L'ACCESSO DIRETTO AL SINGOLO COMPONENTE ATTRAVERSO L'INDICE.

44

SE VOLESSIMO DEFINIRE LE SPECIFICHE DI UN VETTORE

SPECIFICA SINTATTICA

TIPI :

VETTORE, INTERO, TIPOELEM

OPERATORI :

CREAVETTORE : ( ) → VETTORE

LEGGI VETTORE : (VETTORE, INTERO) → TIPOELEM

SCRIVIVETTORE : (VETTORE, INTERO, TIPOELEM) → VETTORE

45

SPECIFICA SEMANTICA

TIPI :

INTERO : L'INSIEME DEI NUMERI INTERI

VETTORE : L'INSIEME DELLE SEQUENZE DI N ELEMENTI DI TIPO TIPOELEM

OPERATORI:

CREAVETTORE = V

POST : PER OGNI i, $1 \le i \le n$, L'i-ESIMO ELEMENTO DEL VETTORE V(i) È UGUALE AD UN DEFINITO ELEMENTO DI TIPO TIPOELEM

46

LEGGIVETTORE(V , i) = e

PRE : $1 \le i \le n$

POST : e=V(i)

SCRIVIVETTORE(V , i , e) = V'

PRE : $1 \le i \le n$

POST : V'(i)= e, V'(j)=V(j) PER OGNI j TALE CHE $1 \le j \le n$ E $j \neq i$

47

REALIZZAZIONE

IN MOLTI LINGUAGGI IL VETTORE È ILTIPO DI DATO PRIMITIVO

IN FORTRAN

CREAVETTORE → DIMENSION V (n)

LEGGI VETTORE (V , I) → V (I)

SCRIVIVETTORE (V , I ,e) → V (I)=e

IN PASCAL CORRISPONDE ALL'ARRAY.

LA CORRISPONDENZA TRA LA SPECIFICA INTRODOTTA E QUELLA DEL PASCAL È :

CREAVETTORE → VAR V ARRAY[1..n] OF TIPOELEM

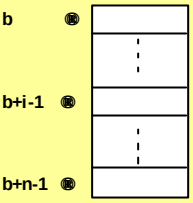
LEGGI VETTORE (V , I) → V [I]

SCRIVIVETTORE (V , I ,e) → V [I]:=e

48

RAPPRESENTAZIONE IN MEMORIA

PER ESEGUIRE UN ASSEGNAMENTO PASCAL **m := V[i]**
SI PRESUPPONE CHE I VALORI DELLE TRE VARIABILI m, I E V[I] SIANO CONTENUTI IN TRE CELLE



V[1] IL VETTORE È ALLOCATO IN MEMORIA IN N CELLE CONSECUTIVE, A PARTIRE
V[I] DA UN INDIRIZZO BASE b.
L'INDIRIZZO DEL GENERICO ELEMENTO V[I] È b+i-1
V[n] CON 1 £ i £ n
L'ACCESSO È DIRETTO

49

LA MATRICE, INTESA COME TABELLA A DUE O A MOLTE DIMENSIONI, È DIRETTA ESTENSIONE DEL VETTORE.
PREVEDE DUE O PIÙ INDICI CHE CONSENTONO L'ACCESSO DIRETTO AL SINGOLO ELEMENTO.
PER UNA MATRICE A TRE DIMENSIONI DI n•m•p ELEMENTI DIAMO LE SPECIFICHE

SPECIFICA SINTATTICA

TIPI :
MATRICE, INTERO, TIPOELEM
OPERATORI :
CREAMATRICE : ( ) ® MATRICE
LEGGIMATRICE : (MATRICE, INTERO, INTERO, INTERO) ® TIPOELEM
SCRIVIMATRICE : (MATRICE, INTERO, INTERO, INTERO, TIPOELEM) ® MATRICE

50

SPECIFICA SEMANTICA

TIPI :
INTERO : INSIEME DI INTERI
MATRICE : INSIEME DI n•m•p ELEMENTI DI TIPO TIPOELEM " L'ELEMENTO CARATTERIZZATO DAGLI INDICI i,j,k E' QUELLO NELLA
i-esima RIGA
j-esima COLONNA
k-esima DIMENSIONE IN PROFONDITA'
etc. ……
……

51

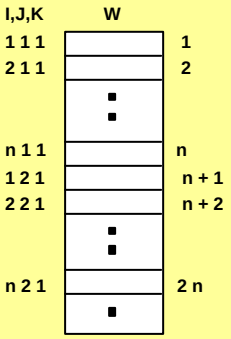
REALIZZAZIONE

IN PASCAL È DISPONIBILE IL DATO PRIMITIVO MATRICE MULTIDIMENSIONALE ED È RIPORTATO AL DATO PRIMITIVO ARRAY

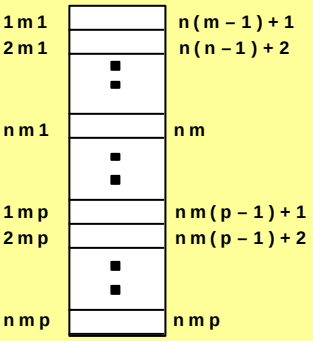
CREAMATRICE ® VAR W ARRAY[1..n,1..m,1..p] OF TIPOELEM
LEGGIMATRICE (W,I,J,K) ® W[I,J,K]

SCRIVIMATRICE (W,I,J,K,e) ® W[I,J,K]:=e

52



53



54

MA SE VOLESSIMO REALIZZARE, ATTRAVERSO ASTRAZIONE DI DATI E ASTRAZIONE FUNZIONALE, PROPRIO UNA STRUTTURA CHE RAPPRESENTI L'ENTE GEOMETRICO MATRICE E LE OPERAZIONI CHE LA MATEMATICA PREVEDE (AD ESEMPIO IL PRODOTTO RIGHE PER COLONNE) COME DOVREMMO OPERARE ?

CHE ALGEBRA DOVREMMO DEFINIRE?

IL PASCAL PREVEDE GLI ARRAY COME DATI PRIMITIVI , MA COSA FARE PER ASSICURARE LA ASTRAZIONE CORRETTA ?

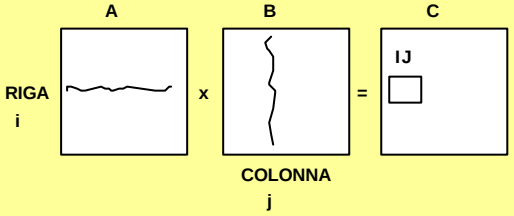
55

NELL'ESEMPIO CHE SEGUE MOSTREREMO COME, PUR ESSENDO IN GRADO DI ASSICURARE L'OPERAZIONE DI PRODOTTO DI MATRICI FACENDO USO DI DATI PRIMITIVI, NON SIAMO IN GRADO DI GARANTIRE I REQUISITI DI ASTRAZIONE SULLE STRUTTURE SE NON UTILIZZIAMO IL CONCETTO DI **TIPO**

Esempio

IL PRODOTTO RIGHE x COLONNE DI DUE MATRICI DI DIMENSIONI, RISPETTIVAMENTE (M x P) E (P x N), È UNA MATRICE DI DIMENSIONI (M x N)

56



$C_{ij} = a_1 * b_{1j} + a_2 * b_{2j} + \ldots\ldots + a_{ip} * b_{pj}$

$C = (C_{IJ} \quad i = 1, \ldots, m \; ; \; j = 1, \ldots, n)$

LA MATRICE PRODOTTO HA COME GENERICO ELEMENTO

$$C(I , J) = \sum_{K=1}^{P} A(I , K) * B(K , J)$$

57

IL CICLO DI CALCOLO È MOLTO SEMPLICE

```
BEGIN
  FOR I:= 1 TO M DO
  BEGIN
    FOR J:= 1 TO N DO
    BEGIN
        S:= 0;
        FOR K:= 1 TO P DO
        S:= S + ( A [I , K] * B [K , J] ) ;
        C[I , J ]:= S ;
    END ;
  END;
END.
```

COSTITUIRA' IL CORPO DELLA PROCEDURA (OPERATORE), MA COME UTILIZZARE LE DICHIARATIVE PER REALIZZARE L'ASTRAZIONE ?

58

SPECIFICA SINTATTICA

DI MATRICE CON L'AGGIUNTA DELL'OPERATORE PRODOTTO MATRICIALE RIGHE x COLONNE

TIPI :

MATRICE , INDICE , TIPOELEM

OPERATORE:

ESEGUIPRODOTTO (A, B , M , N , P) → C

DOVE :

A È UNA MATRICE DI DIMENSIONI (M x P)

B È UNA MATRICE DI DIMENSIONI (P x N)

C È LA MATRICE PRODOTTO DI DIMENSIONI (M x N)

59

SPECIFICA SEMANTICA

ESEGUIPRODOTTO (A, B , M , N , P) → C

PRE : $A \{A_{IK} \quad \forall\, i = 1 , \ldots , m \; ; \; k = 1, \ldots , p \}$

$B \{B_{KJ} \quad \forall\, i = 1 , \ldots , m \; ; \; k = 1, \ldots , p \}$

POST : $C \{C_{IJ} = \sum_{k=1}^{P} A_{IK} * B_{KJ} \; \forall \; i = 1 , \ldots , m$

$j = 1 , \ldots , n \}$

60

REALIZZAZIONE (PASCAL)

UN METODO INADATTO E' QUELLO DI NON USARE I TIPI

```
CONST
      M = … ; P = … ; N = … ;
VAR
      I  : 1 . . M;
      J : 1 . . N;
      K:  1 . . P;
      S: TIPOELEM;
      ELEM : TIPOELEM ;
      A : ARRAY [1 . . m,1 . . p] OF ELEM;
      B : ARRAY [1 . . p ,1 . . n] OF ELEM;
      C : ARRAY [1 . . m,1 . . n] OF ELEM;
……..
```

61

UN METODO PIÙ ADATTO

```
CONST
   DIMMAX = 50 ;
TYPE
    TIPOELEM :
    INDICE : 1 . . DIMMAX;
    MATRICE = ARRAY[1. .DIMMX,1. .DIMMAX] OF TIPOELEM
VAR
 NUMERO_RIGHE_PRIMA, NUMERO_COLONNE_PRIMA,
 NUMERO_RIGHE_SECONDA, NUMERO_COLONNE_SECONDA:
                                        INDICE;
 DIMCOM: INDICE ; (*dimensione comune*)
 PRIMAMAT,SECONDAMAT,MATPRODOTTO:MATRICE;
```

62

INTESTAZIONE PROCEDURA ESEGUIPRODOTTO

```
PROCEDURE   ESEGUI_PRODOTTO (A , B : MATRICE;
                             M , N , M : INDICE ;
                             VAR  C : MATRICE);
```

COME SI VEDE L'OPERATORE E' RISERVATO A MATRICI DEFINITE OMOGENEE (TIPO MATRICE) MA SI DOVRA' CONTROLLARE NEL PROGRAMMA CHE IL NUMERO DI COLONNE DELLA PRIMA MATRICE SIA UGUALE AL NUMERO DI RIGHE DELLA SECONDA (DIMCOM)

CHIAMATA PROCEDURA ESEGUIPRODOTTO

```
ESEGUI_PRODOTTO (PRIMAMAT, SECONDAMAT,
                 NUMERO_RIGHE_PRIMA,
                 NUMERO_COLONNE_SECONDA,
                 DIMCOM, MATPRODOTTO);
```

63