Corso di Laurea in INFORMATICA

Algoritmi e Strutture Dati
MODULO 6

Insiemi

Specifiche, rappresentazione e confronto tra realizzazioni alternative.

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.

INSIEMI

UN INSIEME È UNA COLLEZIONE (O FAMIGLIA) DI ELEMENTI (COMPONENTI O MEMBRI) DI TIPO OMOGENEO.
A DIFFERENZA DELLE LISTE GLI ELEMENTI NON SONO CARATTERIZZATI DA UNA POSIZIONE NÉ POSSONO APPARIRE PIÙ DI UNA VOLTA.

SOLITAMENTE RAPPRESENTATI GRAFICAMENTE



IN MATEMATICA POSSONO ESSERE DEFINITI

ESTENSIONALMENTE

A = { GIALLO, ROSSO, BLU }

OPPURE INTENSIONALMENTE ATTRAVERSO LE PROPRIETA' CHE DEVONO AVERE I COMPONENTI

B = { ELEMENTI NEL COLLEGIO BA19 NEL 1994 }

C = { NUMERI REALI COMPRESI TRA 0 E 1 }

IN INFORMATICA CI RIFERIAMO AL MODO ESTENSIONALE

2

IL NUMERO DI ELEMENTI (|A| DETTO CARDINALITÀ) RAPPRESENTA LA DIMENSIONE DELL'INSIEME

|A| = 3

|B| È FINITA

|C| È INFINITA

LA RELAZIONE FONDAMENTALE È QUELLA DI APPARTENENZA x ∈ A , DA CUI DERIVA L'INCLUSIONE B ⊆ A .

OPERAZIONI PRINCIPALI SONO :

UNIONE A ∪ B

INTERSEZIONE A ∩ B

DIFFERENZA A − B

3

SPECIFICA SINTATTICA

TIPI :

INSIEME , BOOLEAN , TIPOELEM

OPERATORI :

CREAINSIEME : ( ) → INSIEME

INSIEMEVUOTO : (INSIEME) → BOOLEAN

APPARTIENE : (TIPOELEM , INSIEME) → BOOLEAN

INSERISCI : (TIPOELEM , INSIEME) → INSIEME

CANCELLA : (TIPOELEM , INSIEME) → INSIEME

UNIONE : (INSIEME , INSIEME) → INSIEME

INTERSEZIONE : (INSIEME , INSIEME) → INSIEME

DIFFERENZA : (INSIEME , INSIEME) → INSIEME

4

SPECIFICA SEMANTICA

TIPI :

INSIEME ⇒ FAMIGLIA DI INSIEMI COSTITUITA DA ELEMENTI DI TIPO TIPOELEM

BOOLEAN ⇒ INSIEME VALORI VERITÀ

OPERATORI :

CREAINSIEME = A

POST : A = Æ

INSIEMEVUOTO(A) = b

POST : b = VERO SE A = Æ

b = FALSO ALTRIMENTI

5

APPARTIENE (x , A) = b

POST : b = VERO SE x ∈ A

b = FALSO ALTRIMENTI

INSERISCI (x , A) = A'

PRE : x ∉ A (OPPURE SENZA PRECONDIZIONE)

POST : A' = A ∪ {x} (SE x ∈ A , A = A')

CANCELLA (x , A) = A'

PRE : x ∈ A (OPPURE SENZA PRECONDIZIONE)

POST : A' = A − {x} (SE x ∉ A , A = A')

6

UNIONE (A , B) = C

POST : C = A È B

INTERSEZIONE (A , B) = C

POST : C = A Ç B

DIFFERENZA (A , B) = C

POST : C = A – B

IN PASCAL INSIEME È UN DATO PRIMITIVO (SET)

(TYPE <IDENT.> = SET OF <TIPO-b>)

GLI OPERATORI:

APPARTENENZA		IN
INCLUSIONE		Ø
UNIONE	SONO INDICATI	+
INTERSEZIONE		*
DIFFERENZA		–

PER GLI ASSEGNAMENTI

C:= [ ] ; B := B + [SUCC(TIPO SCALARE)];

D:= ['d' , 'f' , 'g']; C := D – [ch]

REALIZZAZIONI

RAPPRESENTAZIONE CON VETTORE BOOLEANO

PER LINGUAGGI CHE NON DISPONGONO DEL TIPO INSIEME, È POSSIBILE RAPPRESENTARE UN INSIEME A, I CUI ELEMENTI SIANO, AD ESEMPIO, INTERI Î [1 , N], ATTRAVERSO UN VETTORE BOOLEANO DI N BIT, IL CUI K-ESIMO VALORE SARÀ "VERO" SE K Î A E "FALSO" SE K Ï A. (VETTORE CARATTERISTICO)

UN'ALTRA POSSIBILE RAPPRESENTAZIONE SI AVVALE DI UNA LISTA I CUI ELEMENTI SONO QUELLI DELL'INSIEME.

IN TAL MODO SI PUO' EVITARE CHE GLI ELEMENTI SIANO ASSOLUTAMENTE DEGLI INTERI

UTILIZZIAMO, COMUNQUE, IL LINGUAGGIO PASCAL PER RAPPRESENTARE LE REALIZZAZIONI.

REALIZZAZIONE CON VETTORE CARATTERISTICO

```
CONST
      N = {intero};
TYPE
   TIPOELEM = 1 .. N ;
    INSIEME = PACKED ARRAY [1..N] OF BOOLEAN ;
--------------------
PROCEDURE CREAINSIEME(VAR A:INSIEME );
VAR
     i : INTEGER;
BEGIN
      FOR i:= 1 TO N DO A[i]:=FALSE;
END;
```



```
FUNCTION    INSIEMEVUOTO(A:INSIEME ):BOOLEAN;
VAR     i:INTEGER;
BEGIN
   i:=1;
   WHILE ( i < N ) AND NOT A[i] DO i:=i + 1;
   INSIEMEVUOTO := A[i]
END;
-----------------------
FUNCTION APPARTIENE(i:TIPOELEM; VAR A:INSIEME):
                                          BOOLEAN;
BEGIN
APPARTIENE:=A[i];
END;
```



```
PROCEDURE INSERISCI(i:TIPOELEM; VAR A:INSIEME );
BEGIN
  A[i]:=TRUE;
END;
-----------------------
PROCEDURE CANCELLA(i:TIPOELEM; VAR A:INSIEME );
BEGIN
  A[i]:=FALSE;
END;
```

```
PROCEDURE   UNIONE(A , B:INSIEME;VAR C:INSIEME);
VAR      i:INTEGER;
BEGIN
    FOR i:= 1 TO N DO C[i]:=A[i]  OR B[i] ;
END;
-----------------------
PROCEDURE  INTERSEZIONE(A , B:INSIEME;
                                        VAR C:INSIEME);
VAR      i:INTEGER;
BEGIN
    FOR i:= 1 TO N DO C[i]:=A[i]  AND  B[i] ;
END;
```

13

```
PROCEDURE DIFFERENZA(A,B:INSIEME; VAR C:INSIEME);
VAR      i:INTEGER;
BEGIN
    FOR i:= 1 TO N DO C[i]:=A[i]  AND NOT B[i] ;
END;
```

14

REALIZZAZIONE CON LISTE NON ORDINATE

GLI ELEMENTI DELLA LISTA SONO QUELLI DELL'INSIEME. NEL CASO SI USINO REALIZZAZIONI CON STRUTTURE DINAMICHE, L'OCCUPAZIONE DI MEMORIA È PROPORZIONALE AL NUMERO DEGLI ELEMENTI PRESENTI NELL'INSIEME

```
TYPE
    INSIEME = ^CELLA ;
    POSIZIONE = ^CELLA ;
    CELLA = RECORD
                ELEMENTO : TIPOELEM ;
                SUCCESSIVO : ^CELLA ;
            END;
```

15

```
FUNCTION APPARTIENE(i:TIPOELEM; VAR A:INSIEME):
                                              BOOLEAN;
VAR
      P:POSIZIONE; TROVATO: BOOLEAN;
BEGIN
    P:= A; TROVATO:= FALSE;
    WHILE (P ≠ NIL) AND (NOT TROVATO)   DO
            IF P^.ELEMENTO=i THEN TROVATO:= TRUE
            ELSE P:= P^.SUCCESSIVO;
APPARTIENE:=TROVATO;
END;
```

16

```
PROCEDURE   INSERISCI(VAR  A:INSIEME; ELEM:TIPOELEM);
VAR
      B:INSIEME;TEMP:POSIZIONE;
BEGIN
  IF NOT APPARTIENE(ELEM,A)     THEN
                BEGIN        TEMP:= A;
                             NEW(B);
                             B^.ELEMENTO:= ELEM;
                             B^.SUCCESSIVO:=TEMP;
                             A:=B;
                END;
 END;
```

L'INSERIMENTO AVVIENE IN TESTA ALLA LISTA SEMPLICE CON CUI È REALIZZATO L'INSIEME

17

```
PROCEDURE DIFFERENZA(A,B:INSIEME;VAR C:INSIEME);
VAR
      P,TEMP:POSIZIONE;
BEGIN
   C:=NIL;   P:=A;
   WHILE P ≠ NIL DO BEGIN
          IF NOT APPARTIENE(P^.ELEMENTO, B)   THEN
              BEGIN      TEMP:= C;
                         NEW(C);
                         B^.SUCCESSIVO:=TEMP;
                         B^.ELEMENTO:= P^.ELEMENTO;
               END;
          P:= P^.ELEMENTO;
        END;
 END;
```

18

REALIZZAZIONE CON LISTE ORDINATE

SE È DEFINITA UNA RELAZIONE £ DI ORDINAMENTO TOTALE SUGLI ELEMENTI DELL'INSIEME, ESSO PUÒ ESSERE RAPPRESENTATO CON UNA LISTA ORDINATA PER VALORI CRESCENTI DEGLI ELEMENTI UTILIZZANDO DUE PUNTATORI CHE SCORRONO OGNUNO SU UN INSIEME.

LA REALIZZAZIONE DEGLI OPERATORI NON PRESENTA PARTICOLARI DIFFICOLTA'.

DI SEGUITO VENGONO DATI COME ESEMPIO IL DIAGRAMMA DI FLUSSO E LA REALIZZAZIONE IN PASCAL DEL SOLO OPERATORE **INTERSEZIONE**.

19

```
PROCEDURE INTERSEZIONE (A,B:INSIEME;  VAR C:INSIEME);
VAR    p,q,r: POSIZIONE;
BEGIN
  C:=NIL;  {Crea la lista C; per motivi di efficienza non si usano gli
operatori della lista;  cio' e' possibile poiche' INTERSEZIONE è
operatore base}
   p:=A;      {p punta al primo elemento della lista A}
   q:=B;      {q punta al primo elemento della lista B}
   r:= C;     {r punta al primo elemento della lista C, ora vuota}
   WHILE (p ¹ NIL) AND  (q ¹ NIL) DO BEGIN
      IF p-.ELEMENTO= q-.ELEMENTO THEN
         {elemento in A = elemento in B, quindi si inserisce in C}
              BEGIN
                   NEW(r);
                   r-.ELEMENTO:= p-.ELEMENTO;
```

20

```
                   r:= r-.SUCCESSIVO;
                   p:= p-.SUCCESSIVO;
                   q:= q-.SUCCESSIVO
              END
        ELSE
{elementi in A e B diversi, si passa al successivo del  piu' piccolo
senza inserire in C}
       IF  p-.ELEMENTO < q-.ELEMENTO
                THEN    p:= p-.SUCCESSIVO
                 ELSE   q:= q-.SUCCESSIVO
   END
END;
```

21

ESEMPI DI APPLICAZIONE DI INSIEMI

PROBLEMA: TROVARE I NUMERI PRIMI APPARTENENTI ALL'INTERVALLO 2..n CON n > 2

ALGORITMO *(SETACCIO DI ERATOSTENE)*

1. METTI TUTTI I NUMERI TRA 2 E n NEL "SETACCIO"
2. SCEGLI E RIMUOVI IL NUMERO IN "SETACCIO"
3. INCLUDI QUESTO NUMERO IN "NUMERI PRIMI"
4. RIMUOVI DAL "SETACCIO" TUTTI I MULTIPLI DI QUESTO NUMERO
5. SE IL "SETACCIO" NON È VUOTO RIPETI I PASSI 2-5

SIA "SETACCIO" CHE "NUMERI PRIMI" SONO DEFINIBILI COME INSIEMI.

22

L'ALGORITMO IN PASCAL NELLA VERSIONE TRADIZIONALE PROPOSTA DA HOARE È IL SEGUENTE

```
CONST
        N=……
VAR
        SETAC , PRIMI:SET OF 2..N;
        PROSSIMO , J : INTEGER;
```

23

```
BEGIN
   SETAC:=[2..N];
   PRIMI :=[ ];
   PROSSIMO:= 2;
     REPEAT
        WHILE NOT(PROSSIMO IN SETAC) DO
                    PROSSIMO:=SUCC(PROSSIMO);
        PRIMI:= PRIMI + [PROSSIMO];
        J:=PROSSIMO;
        WHILE J £ N  DO BEGIN   SETAC:=SETAC – [J];
                                J:= J + PROSSIMO;
                        END;
  UNTIL SETAC=[ ];
END.
```

24

DIAMO DI SEGUITO IL SETACCIO DI *ERATOSTENE* CON UNA OPPORTUNA ASTRAZIONE DATI DEL DATO INSIEME

```
CONST
        N=……;

TYPE
   TIPOELEM = 1.. N;
    INSIEME = …….;

VAR
        SETAC, PRIMI:I NSIEME;
        PROSSIMO, J : INTEGER;
```

25

```
BEGIN
   CREAINSIEME(SETAC);
   CREAINSIEME(PRIMI);
    FOR J:=2 TO N    DO      INSERISCI(J,SETAC);
    PROSSIMO:=1;
    REPEAT
          PROSSIMO:= PROSSIMO + 1
          WHILE NOT(APPARTIENE(PROSSIMO,SETAC))
               DO    PROSSIMO:=PROSSIMO+1;
          INSERISCI (PROSSIMO,PRIMI);
          J:=PROSSIMO;
          WHILE J £ N  DO BEGIN   CANCELLA(J,SETAC);
                                  J:= J + PROSSIMO
                          END
   UNTIL (INSIEMEVUOTO(SETAC))
END;
```

26

CI SONO SPESSO DELLE APPLICAZIONI CHE NON RICHIEDONO L'USO DI TUTTE LE OPERAZIONI

AD ESEMPIO:

- TENERE LE REGISTRAZIONI DEGLI STIPENDI DEGLI IMPIEGATI DI UNA IMPRESA: NON INTERESSA ALCUNA STRUTTURA CHE RIGUARDA GLI IMPIEGATI E LE OPERAZIONI **UNIONE, INTERSEZIONE E DIFFERENZA** NON SONO NECESSARIE
- IMPLEMENTARE LA TABELLA DEI SIMBOLI: COME E' NOTO LA **TABELLA DEI SIMBOLI** DI UN COMPILATORE È USATA PER MEMORIZZARE I NOMI DELLE COSTANTI, DEI TIPI E DELLE VARIABILI DI UN PROGRAMMA SORGENTE. TRADIZIONALMENTE SI USANO STRUTTURE PARTICOLARI (TAVOLE O DIZIONARI).

27

IN QUESTI CASI SI OPERA ATTRAVERSO LA STRUTTURA

MFSET(MERGE-FIND-SET)

UN MFSET È UNA PARTIZIONE DI UN INSIEME FINITO IN SOTTOINSIEMI DISGIUNTI DETTI COMPONENTI.

LE OPERAZIONI CONSENTITE PERMETTONO DI :

- STABILIRE A QUALE COMPONENTE APPARTIENE UN ELEMENTO GENERICO
- UNIRE DUE COMPONENTI DISTINTE IN UNA SOLA COMPONENTE LASCIANDO INALTERATE LE COMPONENTI RIMANENTI

28

ALGEBRA DI MFSET (MERGE-FIND-SET)

SPECIFICA SINTATTICA

TIPI :

INSIEME , TIPOELEM , MFSET, COMPONENTE

OPERATORI :

CREAMFSET : (INSIEME) ® MFSET
FONDI : (TIPOELEM,TIPOELEM,MFSET) ® MFSET
TROVA : (TIPOELEM,MFSET) ® COMPONENTE

29

SPECIFICA SEMANTICA

TIPI :

INSIEME ⇒ FAMIGLIA DI INSIEMI COSTITUITA DA ELEMENTI DI TIPO TIPOELEM

MFSET ⇒ FAMIGLIA DI PARTIZIONI DI INSIEMI DI ELEMENTI DI TIPO TIPOELEM

COMPONENTE ⇒ SOTTOINSIEME DI INSIEME, CHE E' ELEMENTO DI MFSET

OPERATORI :

CREAMFSET (A)= S

POST : S È UNA FAMIGLIA DI n=|A| COMPONENTI C_1 , C_2 , … C_n OGNUNO DELLE QUALI CONTIENE UNO E UN SOLO ELEMENTO DI A E TALI CHE È C_i = A $1 \le i \le n$

30

FONDI (X , Y , S)= S'

PRE : X E Y SONO APPARTENENTI A COMPONENTE DISTINTE DI S

POST : S' È DA TUTTE LE COMPONENTI CHE NON CONTENGONO X E Y E DA UNA NUOVA

TROVA (X , S)= C

PRE : X APPARTIENE AD UNA COMPONENTE DI S

POST : C È L'IDENTIFICATORE DELLE COMPONENTE CUI X APPARTIENE

31

IN LETTERATURA SI PUÒ TROVARE ANCHE UN ALTRO OPERATORE TROVA TESO A VERIFICARE SE, DATI DUE ELEMENTI, QUESTI APPARTENGONO ALLA STESSA COMPONENTE.

TROVA: (TIPOELEM , TIPOELEM , MFSET) ® BOOLEAN

TROVA (X , Y , S)= b

PRE : X E Y APPARTENGONO A COMPONENTI DI S

POST : b= VERO SE X E Y Î ALLA STESSA COMPONENTE

b= FALSO ALTRIMENTI

32

REALIZZAZIONE EFFICACI DI MFSET PREVEDONO L'USO DI STRUTTURE AD ALBERO CHE SARANNO PRESENTATE PIU' TARDI. ORA E' DATA UNA RAPPRESENTAZIONE INTUITIVA.

SIA A = {1 , 2 , 3 , 4 , 5, 6 , 7}

CREAMFSET RESTITUISCE

S = { [1] , [2] , [3] , [4] , [5] , [6] , [7] }

POICHÉ

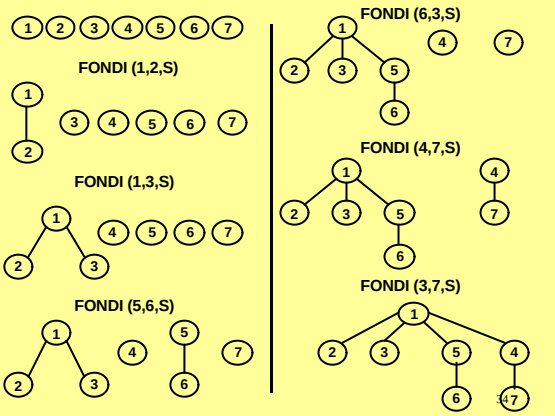
TROVA (1 , S) ¹ TROVA (5 , S)

SI PUO' APPLICARE LA OPERAZIONE DI FUSIONE

FONDI (1 , 5 , S) = S

S' = { [2] , [3] , [4] , [1 , 5] , [6] , [7] }

33



34