1.2 Υλοποίηση του ΑΤΔ Σύνολο με Πίνακα

Στην παρούσα ενότητα, θα παρουσιάσουμε τον τρόπο υλοποίησης του ΑΤΔ σύνολο χρησιμοποιώντας τη δομή δεδομένων του πίνακα.

Συγκεκριμένα, θα χρησιμοποιήσουμε ένα λογικό πίνακα S, όπου S[i] είναι TRUE αν το στοιχείο που αντιστοιχεί στο i-οστό στοιχείο του καθολικού συνόλου ανήκει στο σύνολο S, διαφορετικά είναι FALSE.

θέση	**τιμή**
0	FALSE
1	TRUE
2	FALSE
3	TRUE
4	FALSE
5	TRUE
6	FALSE
7	TRUE
8	FALSE
9	TRUE

Για παράδειγμα, αν θεωρήσουμε το σύνολο OddNumbers = {1, 3, 5, 7,9}, όπου το καθολικό σύνολο είναι τα ψηφία 0..9, τότε αυτό μπορεί να παρασταθεί με τον πίνακα 10 θέσεων που παρουσιάζεται αριστερά.
Η 1η θέση του πίνακα αντιστοιχεί στο 1ο στοιχείο του καθολικού συνόλου, δηλαδή στο ψηφίο 0, η 2η στο ψηφίο 1 κ.τ.λ.

Για την υλοποίηση του ΑΤΔ σύνολο με πίνακα χρησιμοποιούνται οι παρακάτω δηλώσεις:

```
 #define megisto_plithos 10      /*μέγιστο πλήθος στοιχείων συνόλου, ενδεικτικά 10*/
typedef enum {
              FALSE, TRUE
} boolean;

typedef boolean typos_synolou[megisto_plithos];
typedef int stoixeio_synolou;
```

Οι **βασικές πράξεις/λειτουργίες** που συνδέονται με τα σύνολα υλοποιούνται εύκολα χρησιμοποιώντας τη δομή δεδομένων του πίνακα. Στη συνέχεια περιγράφεται ο τρόπος υλοποίησής τους, θεωρώντας ότι οι μεταβλητές synolo, s1, s2, enosi, tomi και diafora είναι μεταβλητές πίνακα τύπου typos_synoloy:

- **Δημιουργία ενός κενού συνόλου** (Dimiourgia): για τη δημιουργία του κενού συνόλου, δηλαδή ενός συνόλου που δεν έχει καθόλου στοιχεία, εκχωρείται σε όλες τις θέσεις του πίνακα synolo η τιμή FALSE.

- **Δημιουργία καθολικού συνόλου** (Katholiko): για τη δημιουργία του καθολικού συνόλου, δηλαδή του συνόλου που περιλαμβάνει όλα τα στοιχεία του συγκεκριμένου τύπου βάσης που έχει δηλωθεί, εκχωρείται σε όλες τις θέσεις του πίνακα synolo η τιμή TRUE.

- **Εισαγωγή στοιχείου** (Eisagogi): για την εισαγωγή, σε ένα σύνολο, του στοιχείου που βρίσκεται στην i-οστή θέση του καθολικού συνόλου απλά εκχωρούμε στην i-οστή θέση του πίνακα synolo την τιμή TRUE.

- **Διαγραφή στοιχείου** (Diagrafi): για την διαγραφή ενός στοιχείου από ένα σύνολο εκχωρούμε στην θέση του πίνακα synolo που βρίσκεται το στοιχείο την τιμή FALSE.

- **Μέλος** (Melos): για να διαπιστώσουμε αν ένα στοιχείο είναι μέλος ενός συνόλου, ελέγχουμε την τιμή που υπάρχει στην θέση του πίνακα synolo στην οποία αντιστοιχεί το στοιχείο. Αν η τιμή της συγκεκριμένης θέσης του πίνακα είναι TRUE τότε το στοιχείο είναι μέλος του συνόλου, διαφορετικά όχι.

- **Κενό** (KenoSynolo): για να ελέγξουμε αν ένα σύνολο είναι κενό εξετάζουμε τα στοιχεία του πίνακα synolo μέχρι να βρούμε:

  (1) ότι κάποιο στοιχείο έχει τιμή TRUE, γεγονός που σημαίνει ότι το σύνολο δεν είναι κενό, ή
  (2) να εξαντληθούν όλα τα στοιχεία του πίνακα, γεγονός που σημαίνει ότι το σύνολο είναι κενό.

- **Ίσα** (IsaSynola): για να ελέγξουμε αν δύο σύνολα s1 και s2 είναι ίσα, συγκρίνουμε τα αντίστοιχα στοιχεία τους μέχρι να βρούμε:

  (1) ότι κάποιο στοιχείο είναι μέλος ενός μόνο εκ των s1 και s2, γεγονός που σημαίνει τα 2 σύνολα δεν είναι ίσα, ή
  (2) να εξαντληθούν όλα τα στοιχεία του πίνακα, γεγονός που σημαίνει ότι τα σύνολα είναι ίσα.

- **Υποσύνολο** (Yposynolo): για να ελέγξουμε αν ένα σύνολο s1 είναι υποσύνολο του s2, εξετάζουμε όλα τα στοιχεία τους μέχρι να βρούμε:

(1) ότι κάποιο στοιχείο που είναι μέλος του s1 δεν είναι μέλος του s2, γεγονός που σημαίνει ότι το s1 δεν είναι υποσύνολο του s2, ή
(2) να εξαντληθούν όλα τα στοιχεία, γεγονός που σημαίνει ότι το το s1 είναι υποσύνολο του s2.

- **Ένωση** (EnosiSynolou): για να βρούμε την ένωση δύο συνόλων s1 και s2 εξετάζουμε τα αντίστοιχα στοιχεία των s1 και s2 (δηλαδή τα στοιχεία που βρίσκονται στις ίδιες θέσεις των πινάκων αυτών) και αν ένα τουλάχιστον από αυτά έχει την τιμή TRUE τότε εκχωρούμε την τιμή TRUE και στην αντίστοιχη θέση του συνόλου της ένωσης, έστω enosi, διαφορετικά εκχωρούμε την τιμή FALSE.

- **Τομή** (TomiSynolou): για να βρούμε την τομή δύο συνόλων s1 και s2, εξετάζουμε τα αντίστοιχα στοιχεία των s1 και s2 και αν και τα 2 έχουν την τιμή TRUE τότε εκχωρούμε την τιμή TRUE και στην αντίστοιχη θέση του συνόλου της τομής, έστω tomi, διαφορετικά εκχωρούμε την τιμή FALSE.

- **Διαφορά** (DiaforaSynolou): για να βρούμε τη διαφορά s1 - s2 δύο συνόλων, εξετάζουμε τα αντίστοιχα στοιχεία των s1 και s2 και αν και για κάθε στοιχείο που είναι μέλος του s1 και δεν είναι μέλος του s2 εκχωρούμε την τιμή TRUE στην αντίστοιχη θέση του συνόλου της διαφοράς, έστω diafora, και σε κάθε άλλη περίπτωση την τιμή FALSE.

Στην πραγματικότητα, όπως θα δούμε και στη συνέχεια, οι πράξεις της ένωσης, της τομής και της διαφοράς υλοποιούνται πολύ εύκολα χρησιμοποιώντας της λογικές πράξεις της διάζευξης (ή - or), της σύζευξης (και - and) και της άρνησης (όχι - not) αντίστοιχα.

Παρακάτω παρουσιάζεται το αρχείο κεφαλίδας SetADT.h και το αρχείο υλοποίησης SetADT.c που υλοποιεί όλες τις πράξεις που περιγράψαμε για τον ΑΤΔ Σύνολο με πίνακα. Μπορεί να χρησιμοποιηθεί σε ένα πρόγραμμα-πελάτη της C με την εντολή

```
#include "SetADT.h";
```

```
                    /*Πακέτο για τον ΑΤΔ ΣΥΝΟΛΟ με πίνακα*/

//filename SetADT.h



   #define megisto_plithos 10        /*ανώτατο όριο για το πλήθος των στοιχείων του συνόλου,
                                      ενδεικτικά επιλέχθηκε ως μέγιστο πλήθος 10*/
typedef enum {
              FALSE, TRUE
} boolean;

typedef boolean typos_synolou[megisto_plithos];
typedef int stoixeio_synolou;

void Dimiourgia(typos_synolou synolo);
void Katholiko(typos_synolou synolo);
void Eisagogi(stoixeio_synolou stoixeio, typos_synolou synolo);
void Diagrafi(stoixeio_synolou stoixeio, typos_synolou synolo);
boolean Melos(stoixeio_synolou stoixeio, typos_synolou synolo);
boolean KenoSynolo(typos_synolou synolo);
boolean IsaSynola(typos_synolou s1, typos_synolou s2);
boolean Yposynolo(typos_synolou s1, typos_synolou s2);
void EnosiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou enosi);
void TomiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou tomi);
void DiaforaSynolou(typos_synolou s1, typos_synolou s2, typos_synolou diafora);

 //filename SetADT.c

void Dimiourgia(typos_synolou synolo)
/*Λειτουργία:       Δημιουργεί ένα σύνολο χωρίς στοιχεία, δηλαδή το κενό σύνολο.
 Επιστρέφει:       Το κενό σύνολο.*/
{
    stoixeio_synolou i;

    for (i = 0; i < megisto_plithos; i++)
        synolo[i] = FALSE;
}

void Katholiko(typos_synolou synolo)
/*Δέχεται:          Ένα σύνολο.
 Λειτουργία:       Δημιουργεί ένα σύνολο με όλα τα στοιχεία παρόντα, έτσι όπως ορίστηκε στο τμήμα
                   δηλώσεων του προγράμματος.
 Επιστρέφει:       Το καθολικό σύνολο που δημιουργήθηκε.*/
{
    stoixeio_synolou i;

    for (i = 0; i < megisto_plithos; i++)
        synolo[i] = TRUE;
}
```

```
void Eisagogi(stoixeio_synolou stoixeio, typos_synolou synolo)
/*Δέχεται:          Ένα σύνολο και ένα στοιχείο.
 Λειτουργία:        Εισάγει το στοιχείο στο σύνολο.
 Επιστρέφει:        Το τροποποιημένο σύνολο.*/
 {
  synolo[stoixeio] = TRUE;
}

void Diagrafi(stoixeio_synolou stoixeio, typos_synolou synolo)
/*Δέχεται:          Ένα σύνολο και ένα στοιχείο.
 Λειτουργία:        Διαγράφει το στοιχείο από το σύνολο.
 Επιστρέφει:        Το τροποποιημένο σύνολο.*/
 {
  synolo[stoixeio] = FALSE;
}

boolean Melos(stoixeio_synolou stoixeio, typos_synolou synolo)
/*Δέχεται:          Ένα σύνολο και ένα στοιχείο.
 Λειτουργία:        Ελέγχει αν το στοιχείο είναι μέλος του συνόλου.
 Επιστρέφει:        Επιστρέφει TRUE αν το στοιχείο είναι μέλος του και FALSE διαφορετικά.*/
{
  return synolo[stoixeio];
}

boolean KenoSynolo(typos_synolou synolo)
/*Δέχεται:          Ένα σύνολο.
 Λειτουργία:        Ελέγχει αν το σύνολο είναι κενό.
 Επιστρέφει:        Επιστρέφει TRUE αν το σύνολο είναι κενό και FALSE διαφορετικά.*/
{
  stoixeio_synolou i;
  boolean keno;

  keno = TRUE;                    /*θέσε στη μεταβλητή keno, η οποία δηλώνει αν το σύνολο
                                    είναι κενό ή όχι, την τιμή TRUE*/
  i = 0;                          /*θέσε στη μεταβλητή i που χρησιμοποιείται ως δείκτης στα
                                    στοιχεία του πίνακα που αναπαριστά το σύνολο την τιμή 0*/
  while (i < megisto_plithos && keno)
                                  /*όσο δεν έχουν εξαντληθεί όλα τα στοιχεία
                                    (i<megisto_plithos) και δεν έχει βρεθεί ότι κάποιο στοιχείο
                                    του καθολικού συνόλου υπάρχει στο σύνολο synolo, δηλαδή η
                                    μεταβλητή keno έχει τιμή TRUE*/
    if (Melos(i, synolo))         /*αν το στοιχείο που βρίσκεται στην i θέση του καθολικού
                                    συνόλου είναι μέλος του συνόλου synolo*/
      keno = FALSE;               /*θέσε στη μεταβλητή keno την τιμή FALSE, αφού βρέθηκε ότι
                                    το σύνολο δεν είναι κενό*/
```

```
    else                              /*αν το στοιχείο που βρίσκεται στην i θέση του καθολικού
                                      συνόλου είναι μέλος του συνόλου synolo*/
      i = i+1;                        /*αύξησε την μεταβλητή i κατά 1 ώστε να δείχνει στο επόμενο
                                      στοιχείο του πίνακα*/
  return keno;                        /*η συνάρτηση επιστρέφει την τιμή της μεταβλητής keno*/
}

boolean IsaSynola(typos_synolou s1, typos_synolou s2)
/*Δέχεται:          Δύο σύνολα s1 και s2.
 Λειτουργία:       Ελέγχει αν τα δύο σύνολα είναι ίσα.
 Επιστρέφει:       Επιστρέφει TRUE αν τα δύο σύνολα έχουν τα ίδια στοιχεία και FALSE
                   διαφορετικά.*/
{
  stoixeio_synolou i;
  boolean isa;

  isa = TRUE;                         /*θέσε στη μεταβλητή isa, η οποία δηλώνει αν τα 2 σύνολα
                                      είναι ίσα ή όχι, την τιμή TRUE*/
  i = 0;                              /*θέσε στη μεταβλητή i που χρησιμοποιείται ως δείκτης στα
                                      στοιχεία του πίνακα που αναπαριστά το σύνολο την τιμή 0*/
  while (i < megisto_plithos && isa)
                                      /*όσο δεν έχουν εξαντληθεί όλα τα στοιχεία
                                      (i<megisto_plithos) και δεν έχει βρεθεί ότι κάποιο στοιχείο
                                      που υπάρχει σε ένα από τα σύνολα s1 και s2 δεν υπάρχει στο
                                      άλλο σύνολο, δηλαδή η μεταβλητή isa έχει τιμή TRUE*/
    if (Melos(i,s1)!= Melos(i,s2))
                                      /*αν το στοιχείο που βρίσκεται στην i θέση του καθολικού
                                      συνόλου είναι μέλος ενός μόνο από τα σύνολα s1και s2*/
      isa = FALSE;                    /*θέσε στη μεταβλητή isa την τιμή FALSE, αφού βρέθηκε ότι τα
                                      2 σύνολα δεν είναι ίσα*/
    else                              /*αν το στοιχείο που βρίσκεται στην i θέση του καθολικού
                                      συνόλου υπάρχει ή δεν υπάρχει και στα 2 σύνολαs1 και s2*/
      i = i+1;                        /*αύξησε την μεταβλητή i κατά 1 ώστε να δείχνει στο επόμενο
                                      στοιχείο των πινάκων των 2 συνόλων*/
  return isa;                         /*η συνάρτηση επιστρέφει την τιμή της μεταβλητής isa*/
}

boolean Yposynolo(typos_synolou s1, typos_synolou s2)
/*Δέχεται:          Δύο σύνολα s1 και s2.
 Λειτουργία:       Ελέγχει αν το σύνολο s1 είναι υποσύνολο του s2.
 Επιστρέφει:       Επιστρέφει TRUE αν το σύνολο s1 είναι ένα υποσύνολο του s2, δηλαδή αν κάθε
                   στοιχείο του s1 είναι και στοιχείο του s2.*/
{
  stoixeio_synolou i;
  boolean yposyn;
```

```
  yposyn = TRUE;                          /*θέσε στη μεταβλητή yposyn, η οποία δηλώνει αν
                                            το σύνολο s1 είναι υποσύνολο του s2 ή όχι, την
                                            τιμή TRUE*/
  i = 0;                                  /*θέσε στη μεταβλητή i που χρησιμοποιείται ως
                                            δείκτης στα στοιχεία του πίνακα που αναπαριστά
                                            το σύνολο την τιμή 0*/
  while (i < megisto_plithos && yposyn)
                                          /*όσο δεν έχουν εξαντληθεί όλα τα στοιχεία
                                            (i<megisto_plithos) και δεν έχει βρεθεί ότι
                                            κάποιο στοιχείο που υπάρχει στο σύνολο s1 δεν
                                            υπάρχει στο σύνολο s2, δηλαδή η μεταβλητή yposyn
                                            έχει τιμή TRUE*/
    if (Melos(i,s1) && !Melos(i,s2))
                                          /*αν το στοιχείο που βρίσκεται στην i θέση του
                                            καθολικού συνόλου είναι μέλος του συνόλου s1 και
                                            δεν είναι μέλος του s2*/
      yposyn = FALSE;                     /*θέσε στη μεταβλητή yposyn την τιμή FALSE, αφού
                                            βρέθηκε ότι το σύνολο s1 δεν είναι υποσύνολο του
                                            s2*/
    else
      i = i+1;                            /*αύξησε την μεταβλητή i κατά 1 ώστε να δείχνει
                                            στο επόμενο στοιχείο των πινάκων των 2 συνόλων*/
  return yposyn;                          /*η συνάρτηση επιστρέφει την τιμή της μεταβλητής
                                            yposyn*/
}

void EnosiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou enosi)
/*Δέχεται:          Δύο σύνολα s1 και s2.
 Λειτουργία:       Δημιουργεί ένα νέο σύνολο με τα στοιχεία που ανήκουν ή στο s1 ή στο s2 ή και
                   στα δύο σύνολα.
 Επιστρέφει:       Επιστρέφει το σύνολο enosi που προκύπτει από την ένωση των συνόλων s1 και s2.*/
{
  stoixeio_synolou i;

  for (i=0; i < megisto_plithos; i++)
       enosi[i] = Melos(i,s1) || Melos(i,s2);
                                          /*Εκτέλεσε τη λογική πράξη melos(i, s1) or melos(i, s2)
                                            και εκχώρησε το αποτέλεσμα στην i θέση του πίνακα enosi.

                                            Στην ουσία δηλαδή, αν το στοιχείο που βρίσκεται στην i
                                            θέση του καθολικού συνόλου είναι μέλος είτε του συνόλου s1
                                            ή του s2 ή και των δύο τότε θα είναι μέλος και της ένωσής
                                            τους, δηλαδή του συνόλου enosi*/
}
```

```
void TomiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou tomi)
/*Δέχεται:          Δύο σύνολα s1 και s2.
 Λειτουργία:       Δημιουργεί ένα νέο σύνολο με τα στοιχεία που ανήκουν και στα δύο σύνολα s1 και
                   s2.
 Επιστρέφει:       Επιστρέφει το σύνολο tomi που προκύπτει από την τομή των συνόλων s1 και s2.*/
{
  stoixeio_synolou i;

  for (i=0; i < megisto_plithos; i++)
    tomi[i] = Melos(i,s1) && Melos(i,s2);
                                    /*Εκτέλεσε τη λογική πράξη melos(i, s1) and
                                    melos(i, s2) και εκχώρησε το αποτέλεσμα στην I θέση του
                                    πίνακα tomi.
                                    Στην ουσία δηλαδή, αν το στοιχείο που βρίσκεται στην i
                                    θέση του καθολικού συνόλου είναι μέλος και των 2 συνόλων
                                    s1 και s2  τότε θα είναι μέλος και της τομής τους, δηλαδή
                                    του συνόλου tomi*/
}
void DiaforaSynolou(typos_synolou s1, typos_synolou s2, typos_synolou diafora)
/*Δέχεται:          Δύο σύνολα s1 και s2.
 Λειτουργία:       Δημιουργεί ένα νέο σύνολο με τα στοιχεία που ανήκουν στο σύνολο s1 και δεν
                   ανήκουν στο s2.
 Επιστρέφει:       Επιστρέφει το σύνολο diafora που προκύπτει από την διαφορά των συνόλων s1-
                   s2.*/
{
  stoixeio_synolou i;

  for (i=0; i < megisto_plithos; i++)
     diafora[i] = Melos(i,s1) && (!Melos(i,s2)) ;
                                    /*Εκτέλεσε τη λογική πράξη melos(i, s1) and (not
                                    melos(i, s2)) και εκχώρησε το αποτέλεσμα στην i
                                    θέση του πίνακα diafora.
                                    Στην ουσία δηλαδή, αν το στοιχείο που βρίσκεται
                                    στην i θέση του καθολικού συνόλου είναι μέλος και
                                    του συνόλου s1 και δεν είναι μέλος του s2  τότε
                                    θα είναι μέλος  της διαφοράς τους, δηλαδή του
                                    συνόλου diafora*/
}
```