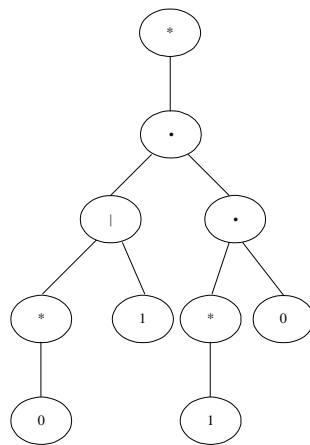


Λυμένες Ασκήσεις

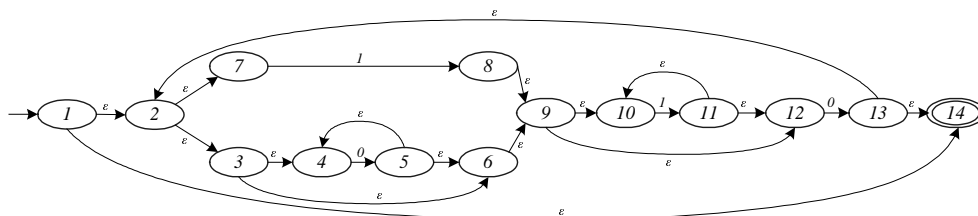
1. Να κατασκευάσετε ένα προσδιοριστικό πεπερασμένο αυτόματο και τον κώδικα για την αναγνώριση συμβολοσειρών που αντιστοιχούν στην κανονική έκφραση: $((0^* | 1) (1^* 0))^*$

ΛΥΣΗ:

- Συντακτική ανάλυση της έκφρασης $((0^* | 1) (1^* 0))^*$ σύμφωνα με τον αλγόριθμο 2.2:



- Ανάπτυξη Thompson σύμφωνα με τον αλγόριθμο 2.3:

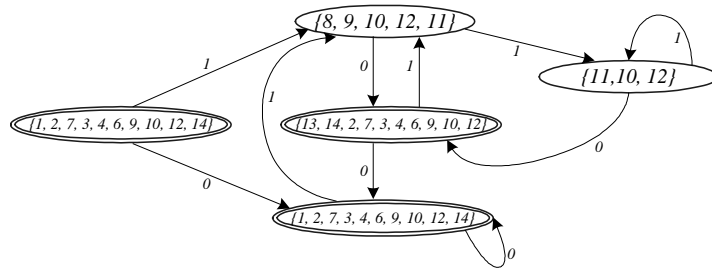


- Υπολογισμός ε-κλείσιμο συνόλων καταστάσεων με τον αλγόριθμο 2.5:

esucc(1)	{1, 2, 7, 3, 4, 6, 9, 10, 12, 14}	esucc(8)	{8, 9, 10, 12}
esucc(2)	{2, 7, 3, 4, 6, 9, 10, 12}	esucc(9)	{9, 10, 12}
esucc(3)	{3, 4, 6, 9, 10, 12}	esucc(10)	{10}
esucc(4)	{4}	esucc(11)	{11, 10, 12}
esucc(5)	{5, 4, 6, 9, 10, 12}	esucc(12)	{12}
esucc(6)	{6, 9, 10, 12}	esucc(13)	{13, 14, 2, 7, 3, 4, 6, 9, 10, 12}
esucc(7)	{7}	esucc(14)	{14}

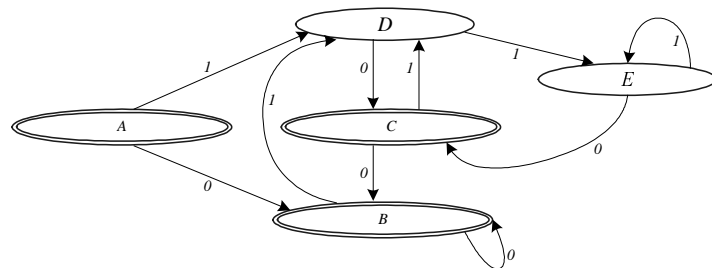
- Ανάπτυξη υποσυνόλων σύμφωνα με τον αλγόριθμο 2.7:

Αρχική κατάσταση προσδιοριστικού αυτόματου $Init_St := esucc(1)$
 Στην $Init_St$ περιέχεται η κατάσταση 14 που είναι κατάληξη στο αυτόματο Thompson. Αυτό έχει ως συνέπεια να δηλώνεται ως κατάληξη και η $Init_St$. Από τις καταστάσεις της $Init_St$ ορίζονται μεταβάσεις στο αυτόματο Thompson, τόσο με ανάγνωση του χαρακτήρα 0, όσο και με ανάγνωση του χαρακτήρα 1. Στην πρώτη περίπτωση εκτελείται μετάβαση είτε στην κατάσταση 5, είτε στη 13, είτε στις καταστάσεις των ε-κλείσιμο συνόλων τους. Έτσι, στο προσδιοριστικό αυτόματο που κατασκευάζεται ορίζεται μετάβαση από την $Init_St$ στη νέα κατάσταση $esucc(5) \cup esucc(13)$. Στη δεύτερη περίπτωση ορίζεται μετάβαση από την $Init_St$ στη νέα κατάσταση $esucc(8) \cup esucc(11)$. Στη συνέχεια εφαρμόζεται η ίδια τεχνική για τις νέες καταστάσεις που δημιουργήθηκαν. Το αποτέλεσμα της ανάπτυξης υποσυνόλων είναι το προσδιοριστικό αυτόματο του σχήματος.



- Ελαχιστοποίηση προσδιοριστικού αυτόματου σύμφωνα με τον αλγόριθμο 2.8:

Για το προσδιοριστικό αυτόματο

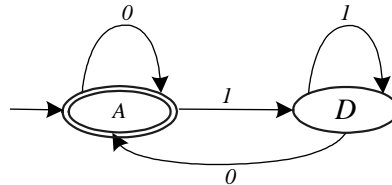


ο πίνακας διάκρισης καταστάσεων σε κλάσεις ισοδυναμίας (παραδείγματα 2.3.9 και 2.3.10) έχει ως εξής:

A				
	B			
		C		
x	x	x	D	
x	x	x		E

90 Μεταγλωττιστές γλωσσών προγραμματισμού

Άρα υπάρχει διάκριση μόνο μεταξύ καταλήξεων και μη καταλήξεων και το ελάχιστο προσδιοριστικό αυτόματο που προκύπτει είναι το:



- Κώδικας αναγνώρισης συμβολοσειρών που αντιστοιχούν στην έκφραση:

```
typedef enum {A,D} STATES;
STATES state;
char c;
...
state=A;
while((c=getc()) != EOF)
{
    switch(state)
    {
        case A:
            if(c == '0')
            {
                state=A;
            }
            else if(c == '1')
            {
                state=D;
            }
            else
            {
                printf("ERROR\n");
                exit(1);
            }
            break;
        case D:
            if(c == '0')
            {
                state=A;
            }
            else if(c == '1')
            {
                state=D;
            }
            else
            {
                printf("ERROR\n");
                exit(1);
            }
            break;
    } /* -- end switch state -- */
    if (c == EOF && state == A) {
        printf("ACCEPTED\n");
    } else {
        printf("NOT ACCEPTED\n");
    }
}
```