

Στο σχήμα 2.14 απεικονίζεται η ανάπτυξη υποσυνόλων της ίδιας έκφρασης μέσω της χρήσης αυτόματου Gushkov (σχήμα 2.9β).



Τα προσδιοριστικά αυτόματα των σχημάτων 2.13 και 2.14 παρά τις διαφορετικές επιγραφές είναι ουσιαστικά ίδια. Επιπλέον, σε αντίθεση με το αυτόματο του σχήματος 2.11 είναι *ελάχιστο*, δηλαδή δεν υπάρχει άλλο αυτόματο με λιγότερες καταστάσεις που αναγνωρίζει ακριβώς την ίδια γλώσσα. Γενικά η ανάπτυξη υποσυνόλων δεν είναι απαραίτητο να δίνει τα ίδια αποτελέσματα, όταν εφαρμόζεται μέσω Thompson και μέσω Gushkov και σίγουρα δε μας δίνει πάντα τα ελάχιστα αυτόματα. Το γεγονός αυτό καθιστά αναγκαία την εφαρμογή του αλγορίθμου ελαχιστοποίησης που περιγράφουμε στην επόμενη παράγραφο.

2.3.9 Ελαχιστοποίηση προσδιοριστικού πεπερασμένου αυτόματου

Τα προσδιοριστικά αυτόματα, που προκύπτουν ως αποτέλεσμα ανάπτυξης υποσυνόλων, διακρίνονται συχνά από μεγάλους πίνακες μεταβάσεων. Αυτό οφείλεται στην αύξηση του αριθμού των καταλήξεων, όπου θα μπορούσε να ολοκληρωθεί η αναγνώριση μιας δεδομένης συμβολοσειράς. Γενικά, η μετατροπή ενός μη προσδιοριστικού αυτόματου, σε προσδιοριστικό, μπορεί να προκαλέσει εκθετική αύξηση του αριθμού των καταστάσεων, που στη συνέχεια πρέπει να περιοριστεί με την εφαρμογή ενός αλγορίθμου ελαχιστοποίησης.

Εξάλλου, έχει θεμελιωθεί και θεωρητικά, ότι για δοθέν προσδιοριστικό αυτόματο, υπάρχει ένα μοναδικό ισοδύναμο αυτόματο με τον ελάχιστο αριθμό καταστάσεων. Ο αλγόριθμος των Hopcroft & Ullman ([HU79]), που περιγράφεται στη συνέχεια, παράγει ως αποτέλεσμα το ελάχιστο αυτόματο, διαιρώντας τις καταστάσεις σε κλάσεις ισοδυναμίας, έτσι ώστε όλες οι καταστάσεις της ίδιας κλάσης να αναγνωρίζουν τις ίδιες συμβολοσειρές. Οι κλάσεις που προκύπτουν, αποτελούν τις «καταστάσεις» του νέου ελάχιστου αυτόματου.

Ορισμός 2.13 Λέμε ότι, σε ένα αυτόματο, η συμβολοσειρά w διακρίνει την κατάσταση S από την κατάσταση T , αν ξεκινώντας από την S και διοχετεύοντας την w , οδηγούμαστε σε μία κατάληξη, ενώ ξεκινώντας από την T και διοχετεύοντας την w , οδηγούμαστε σε κατάσταση, που δεν είναι κατάληξη.

Θεωρούμε ότι η συμβολοσειρά ε διακρίνει τις καταλήξεις από τις υπόλοιπες καταστάσεις του αυτόματου. Ο αλγόριθμος λειτουργεί ως μία συνεχής βελτίωση της αρχικής διαίρεσης του χώρου των καταστάσεων, σε καταλήξεις και μη καταλήξεις. Οι κλάσεις ισοδυναμίας συγκροτούνται σταδιακά από καταστάσεις, με τέτοιο τρόπο, ώστε για κάθε ζεύγος καταστάσεων από διαφορετικές κλάσεις, αυτές να έχουν διακριθεί από κάποια συμβολοσειρά εισόδου, ενώ για κάθε ζεύγος καταστάσεων της ίδιας κλάσης, να μην έχει συμβεί ακόμη κάτι τέτοιο.

72 Μεταγλωττιστές γλωσσών προγραμματισμού

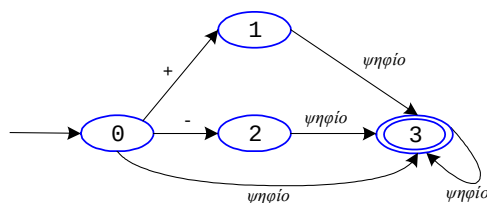
Αυτό επιτυγχάνεται ως εξής: για κάθε κλάση ισοδυναμίας από τις ήδη σχηματισθείσες και για μία είσοδο a , ελέγχονται όλες οι μεταβάσεις από όλες τις καταστάσεις. Αν υπάρχουν μεταβάσεις σε καταστάσεις διαφορετικών κλάσεων, τότε δημιουργούνται δύο κλάσεις καταστάσεων, με τέτοιο τρόπο, ώστε όλες οι καταστάσεις της κάθε κλάσης ισοδυναμίας να οδηγούν στην ίδια κλάση καταστάσεων. Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου διαπιστωθεί ότι στην τρέχουσα διαίρεση του χώρου των καταστάσεων δεν υπάρχουν κλάσεις ισοδυναμίας, που να μπορούν να διαιρεθούν περισσότερο.

Για να διευκολύνουμε την κατανόηση του αλγορίθμου παρουσιάζουμε με εποπτικό τρόπο την εφαρμογή του στα δύο παραδείγματα ελαχιστοποίησης που ακολουθούν.

Παράδειγμα 2.3.9

Στο παράδειγμα αυτό παρουσιάζουμε την ελαχιστοποίηση του αυτόματου του σχήματος 2.15, που αντιστοιχεί στην κανονική έκφραση

$$(+|-)? \{ψηφίο\} \{ψηφίο\}^*$$



Σχήμα 2.15 Προσδιοριστικό αυτόματο με πλεονασματικές καταστάσεις

Στο αυτόματο του σχήματος δεν απεικονίζεται η κατάσταση «λάθος», που όπως τονίστηκε στην παράγραφο 2.3.1 υπάρχει σε κάθε τυπικά ορισμένο αυτόματο. Κατά την ελαχιστοποίηση του αυτόματου θα ληφθεί υπόψη και η κατάσταση «λάθος». Στον πίνακα που ακολουθεί για κάθε ζεύγος καταστάσεων καταγράφουμε το αν αυτές διακρίνονται από κάποια συμβολοσειρά εισόδου. Σε πρώτη φάση βρίσκουμε τα ζεύγη καταστάσεων, που διακρίνονται από τη συμβολοσειρά ε , δηλαδή τα ζεύγη καταστάσεων που αποτελούνται από μία κατάληξη και μία κατάσταση που δεν είναι κατάληξη.

0				
	1			
		2		
x	x	x	3	
			x	«λάθος»

Στη συνέχεια παρατηρούμε ότι $0 \rightarrow 3$ και «λάθος» $\rightarrow$ «λάθος». Εφόσον όμως για τις καταστάσεις 3 και «λάθος» έχει ήδη καταγραφεί στον πίνακα ότι διακρίνονται μεταξύ τους, συνάγεται το ίδιο και για τις 0 και «λάθος». Με ανάλογο τρόπο συμπεραίνουμε

ότι διακρίνονται η 1 από την «λάθος» και η 2 από την «λάθος» και ο πίνακας παίρνει πλέον τη μορφή :

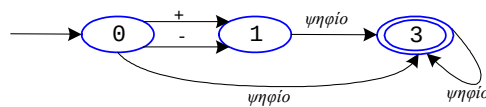
0				
	1			
		2		
x	x	x	3	
x	x	x	x	«λάθος»

Η διαδικασία συμπλήρωσης του πίνακα συνεχίζεται μέχρι το σημείο εκείνο στο οποίο δεν είναι δυνατή η εύρεση νέου ζεύγους καταστάσεων, που διακρίνονται από κάποιο χαρακτήρα εισόδου .

Παρατηρούμε ότι $0 \xrightarrow{+} 1$ και $1 \xrightarrow{+} \text{«λάθος»}$, αλλά ήδη έχει καταγραφεί στον πίνακα ότι η 1 διακρίνεται από την «λάθος». Συμπεραίνουμε λοιπόν ότι η 0 διακρίνεται από την 1 και με ανάλογο τρόπο ότι η 0 διακρίνεται επίσης από την 2. Ο συμπληρωμένος πίνακας είναι τελικά ο

0				
x	1			
x		2		
x	x	x	3	
x	x	x	x	«λάθος»

από όπου διαπιστώνεται ότι τελικά μόνο οι καταστάσεις 1 και 2 ανήκουν στην ίδια κλάση ισοδυναμίας. Μετονομάζουμε την κατάσταση 2 σε 1 και τελικά παίρνουμε το ελάχιστο αυτόματο του σχήματος 2.16.



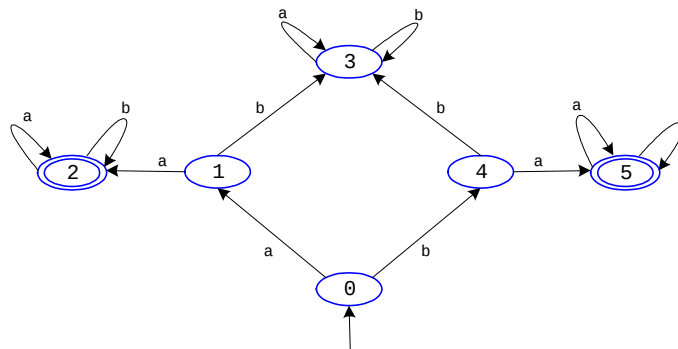
Σχήμα 2.16 Ελάχιστο προσδιοριστικό αυτόματο για την έκφραση $(+|-)^* \{\text{ψηφίο}\}$

■

Παράδειγμα 2.3.10

Στο παράδειγμα αυτό εφαρμόζουμε τον αλγόριθμο ελαχιστοποίησης σε ένα προσδιοριστικό αυτόματο με πλεονασματικές καταστάσεις, που περιλαμβάνει περισσότερες από μία καταλήξεις (σχήμα 2.17).

74 Μεταγλωττιστές γλωσσών προγραμματισμού



Σχήμα 2.17 Προσδιοριστικό αυτόματο με πλεονασματικές καταστάσεις

Οι καταστάσεις 2 και 5 είναι τελικές και άρα η συμβολοσειρά ε τις διακρίνει από τις υπόλοιπες αλλά δεν τις διακρίνει μεταξύ τους :

0					
	1				
x	x	2			
		x	3		
		x		4	
x	x		x	x	5

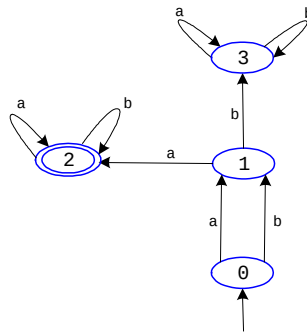
Παρατηρούμε ότι $0 \xrightarrow{a} 1$ και $1 \xrightarrow{a} 2$, αλλά οι 1 και 2 έχει ήδη καταγραφεί ότι διακρίνονται μεταξύ τους. Συμπεραίνουμε λοιπόν ότι διακρίνεται και η 0 από την 1 και με τον ίδιο τρόπο διαπιστώνεται ότι διακρίνονται η 0 από την 4, η 4 από την 3 και η 3 από την 1. Οι καταστάσεις 0 και 3 δε διακρίνονται μεταξύ τους γιατί με μετάβαση a έχουμε τις καταστάσεις 1 και 3 και με μετάβαση b τις καταστάσεις 4 και 3, που και στις δύο περιπτώσεις δεν έχουν βρεθεί να διακρίνονται μεταξύ τους. Ο πίνακας έχει ως εξής :

0					
x	1				
x	x	2			
	x	x	3		
x		x	x	4	
x	x		x	x	5

Επανεξετάζοντας τώρα τις καταστάσεις 0 και 3 παρατηρούμε ότι με ανάγνωση του a υπάρχουν μεταβάσεις αντίστοιχα στις 1 και 3, που όμως τώρα έχει καταγραφεί στον πίνακα ότι διακρίνονται μεταξύ τους. Κατά συνέπεια διακρίνονται και οι 0 και 3, ενώ δεν προκύπτει κάποια επιπλέον πληροφορία για το ζεύγος καταστάσεων 1 και 4, καθώς και το ζεύγος των 2 και 5.

0					
x	1				
x	x	2			
x	x	x	3		
x		x	x	4	
x	x		x	x	5

Εξετάζοντας τα δύο εναπομείναντα ζεύγη δεν παρατηρούνται αλλαγές στον πίνακα και άρα η διαδικασία ολοκληρώνεται με τη διαπίστωση ότι οι καταστάσεις 1 και 4 είναι ισοδύναμες, όπως επίσης και οι καταστάσεις 2 και 5. Το ελάχιστο αυτόματο απεικονίζεται στο σχήμα 2.18.



Σχήμα 2.18 Ελάχιστο προσδιοριστικό αυτόματο για το αυτόματο του σχ. 2.17

Αλγόριθμος 2.8: Ελαχιστοποίηση προσδιοριστικού αυτόματου

Είσοδος: Ένα προσδιοριστικό πεπερασμένο αυτόματο M , με χώρο καταστάσεων S , αλφάβητο Σ , αρχική κατάσταση s_0 και σύνολο καταλήξεων A

Έξοδος: Ένα ελάχιστο προσδιοριστικό αυτόματο M' με γλώσσα $L(M)$

Περιγραφή:

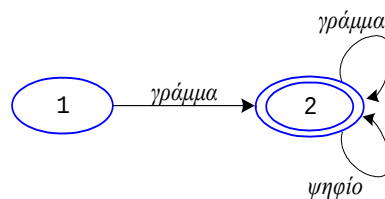
1. Θεωρούμε την αρχική διαίρεση, Δ_0 , του χώρου των καταστάσεων, στην κλάση των καταλήξεων A και στην κλάση των καταστάσεων $S-A$, που δεν αποτελούν κατάληξη.
2. Για (κάθε κλάση ισοδυναμίας G της Δ_1) επανέλαβε
διαίρεσε το G σε κλάσεις, έτσι ώστε δύο καταστάσεις s και t του G να είναι στην ίδια κλάση, αν και μόνο αν, για όλους τους χαρακτήρες του αλφαβήτου οι αντίστοιχες μεταβάσεις οδηγούν σε καταστάσεις της ίδια κλάσης της Δ_1 .
αντικατέστησε στη Δ_{i+1} , την κλάση G , με τις νέες κλάσεις, που σχηματίζονται
Τέλος επανάληψης
3. Αν $\Delta_{i+1} = \Delta_i$ τότε $\Delta_{\text{τελική}} = \Delta_{i+1}$, διαφορετικά επανέλαβε το βήμα 2.
4. Επιλέγεται μία αντιπροσωπευτική κατάσταση από κάθε κλάση της $\Delta_{\text{τελική}}$. Οι καταστάσεις αυτές θα αποτελέσουν τις καταστάσεις του ελάχιστου αυτόματου M' .
5. Αν στο M' υπάρχει νεκρή κατάσταση, κατάσταση δηλαδή, που δεν είναι κατάληξη και χαρακτηρίζεται από μεταβάσεις στον εαυτό της, για όλους τους χαρακτήρες του αλφαβήτου, τότε αυτή διαγράφεται. Επίσης διαγράφονται όλες οι καταστάσεις, που δεν είναι προσεγγίσιμες από την αρχική κατάσταση και αγνοούνται όλες οι μεταβάσεις στις καταστάσεις, που διαγράφονται.

Παράδειγμα 2.3.11

Το προσδιοριστικό αυτόματο του σχήματος 2.11 προήλθε από τη μετατροπή της ανάπτυξης Thompson της έκφρασης,

$$\{\text{γράμμα}\}((\{\text{γράμμα}\} \mid \{\text{ψηφίο}\})^*)$$

για τα ονόματα μιας γλώσσας προγραμματισμού. Οι μεταβάσεις όλων των καταλήξεων οδηγούν σε καταλήξεις, τόσο στην περίπτωση, που ο χαρακτήρας εισόδου είναι γράμμα, όσο και στην περίπτωση, που είναι ψηφίο.



Σχήμα 2.19 Το ελάχιστο αυτόματο της ανάπτυξης υποσυνόλων από το σχήμα 2.11

Άρα οι τρεις καταλήξεις δε διακρίνονται από κανένα χαρακτήρα και για το λόγο αυτό, στο ελάχιστο αυτόματο του σχήματος 2.19, εκφράζονται από μία ενιαία κατάσταση. ■

2.4 Γεννήτριες λεξικών αναλυτών

Η πρώτη ευρέως διαθέσιμη γεννήτρια λεξικού αναλυτή, ήταν το εργαλείο `lex`, που εμφανίστηκε και εξακολουθεί να περιλαμβάνεται στα συνοδευτικά προγράμματα του λειτουργικού συστήματος `UNIX`. Το `lex` δέχεται ως είσοδο ένα αρχείο περιγραφής του προς ανάπτυξη λεξικού αναλυτή και παράγει ως αποτέλεσμα τον πίνακα μεταβάσεων ενός αυτόματου και τον κώδικα `C` της λεξικής ανάλυσης.

Το `flex` (`fast lex`) είναι ένα εργαλείο μεταγενέστερο του `lex`, που αν και σε μεγάλο βαθμό συμβατό με αυτό, διαθέτει κάποιες μικρές επεκτάσεις και τη δυνατότητα να παράγει αποδοτικότερο κώδικα λεξικής ανάλυσης.

Πάντως και τα δύο εργαλεία είναι ελεύθερα διαθέσιμα στο διαδίκτυο και κυκλοφορούν και σε εκδόσεις κατάλληλες, για χρήση σε προσωπικούς υπολογιστές.

Η δημιουργία ενός λεξικού αναλυτή, ξεκινά με τη σύνταξη ενός αρχείου περιγραφής του, που διοχετεύεται στη συνέχεια, ως είσοδος, στη γεννήτρια. Ο κώδικας, που προκύπτει ως αποτέλεσμα, περιλαμβάνεται στο αρχείο με όνομα `lex.yy.c` ή `lex.yy.c` και ουσιαστικά αποτελείται από ένα πίνακα μεταβάσεων, που περιλαμβάνεται σε μία κατάλληλα διατυπωμένη συνάρτηση ανάγνωσης λεξικών μονάδων, με το όνομα `yylex()`.