Τυπική επαλήθευση και χρονοπρογραμματισμός για συστήματα πραγματικού χρόνου

Ελένη Καλυβιανάκη

Μεταπτυχιακο Προγραμμα στη Λογικη, Θεωρια Αλγοριθμων και Υπολογισμου μΠλ∀

Ιούλιος 2000

Περιεχόμενα

Πρόλογος **2**

1 Εισαγωγή **5**

1.1 Γενικά περί τυπικών μεθόδων 7

1.1.1 Τυπική επαλήθευση 7

2 Θεωρητικά εργαλεία **11**

2.1 Συγχρονισμένα αυτόματα 12

2.1.1 Συγχρονισμένα αυτόματα με προθεσμίες 18

2.2 Λογικές πραγματικού χρόνου 20

2.2.1 Λογικές με φραγμένους χρονικούς τελεστές 22

2.2.2 Λογικές με ποσοδείκτες δέσμευσης 25

2.2.3 Λογικές με μεταβλητή ρολογιού 25

3 Τυπική επαλήθευση **27**

3.1 Απόδειξη θεωρήματος 28

3.2 Έλεγχος μοντέλου 29

3.2.1 Χρήση συγχρονισμένων αυτόματων 31

3.2.2 Συνδυασμός χρήσης συγχρονισμένων αυτομάτων και λογικών 32

3.3 Σχετικά με τις υλοποιήσεις 34

4 Μία τεχνική χρονοπρογραμματισμού **37**

4.1 Το πρόβλημα 38

4.2 Τυπική περιγραφή του προβλήματος 40
4.3 Σύνθεση του εργαλείου χρονοπρογραμματισμού 45

5 Επίλογος **49**

6 Βιβλιογραφία **54**

Πρόλογος

Οι τυπικές μέθοδοι και ιδιαίτερα το θέμα της τυπικής επαλήθευσης είναι αυτή τη στιγμή στο κέντρο της επιστημονικής έρευνας στην επιστήμη των υπολογιστών. Η εξέλιξή τους υπήρξε ραγδαία και μία από τις επόμενες κατευθύνσεις στην ανάπτυξή τους είναι αυτή τη στιγμή η διερεύνηση της εφαρμογής των μεθόδων τους σε συστήματα πραγματικού χρόνου.

Η μελέτη των συστημάτων αυτών έχει ιδιαίτερο θεωρητικό ενδιαφέρον αλλά οδηγεί και σε πολύ χρήσιμες εφαρμογές στην επιστήμη των υπολογιστών. Η ενασχόληση στο πλαίσιο της τυπικής επαλήθευσης έχει οδηγήσει τα τελευταία χρόνια στην δημιουργία και ανάπτυξη αξιόλογων φορμαλισμών όπως τα συγχρονισμένα αυτόματα και οι λογικές πραγματικού χρόνου. Οι φορμαλισμοί αυτοί μας επιτρέπουν να χρησιμοποιήσουμε γνωστές τεχνικές της τυπικής επαλήθευσης - κυρίως μεθόδους ελέγχου μοντέλων - με ανάλογα αποτελέσματα στα συστήματα πραγματικού χρόνου.

Προχωρώντας ένα βήμα παραπέρα, η τυπική επαλήθευση για συστήματα πραγματικού χρόνου αγγίζει το θέμα του χρονοπρογραμματισμού - ένα θέμα που απαιτεί διαφορετική προσέγγιση λόγω των ιδιαιτεροτήτων του αλλά και που προκαλεί ιδιαίτερο ενδιαφέρον. Παρουσιάζεται εδώ μία πρωτοποριακή τεχνική η οποία δίνει ένα στίγμα για το που θα μπορούσε να οδηγηθεί η μελλοντική έρευνα.

Κεφάλαιο 1

Εισαγωγή

Αρχίζει σχεδόν να γίνεται δύσκολο να φανταστούμε πως ήταν η ζωή πριν την "εισβολή" των υπολογιστών περίπου μισό αιώνα πριν. Τα συστήματα υπολογιστών - σε όλες τους τις μορφές- αποτελούν πλέον αναπόσπαστο κομμάτι της ανθρώπινης δραστηριότητας σε όλους τους τομείς και μάλιστα, με όλο και πιο μεγάλο ποσοστό ευθύνης σε αυτούς. Ταυτόχρονα, αυτά τα συστήματα γίνονται όλο και πιο περίπλοκα λόγω της εξέλιξης της επιστήμης υπολογιστών σε όλες τις περιοχές αλλά και εξαιτίας της έντονης διείσδυσής τους σε όλες τις εφαρμογές. Μέσα στο πλαίσιο αυτό, προβάλλει το προφανές ερώτημα: πόσο αξιόπιστο μπορούμε να εγγυηθούμε ότι είναι ένα σύστημα δεδομένης της πολυπλοκότητας αλλά και της καίριας σημασίας του;

Η προσέγγιση του προβλήματος δεν είναι εύκολη και σίγουρα μπορεί να έχει πολλές κατευθύνσεις. Μία από αυτές η οποία φαίνεται να κερδίζει αρκετό έδαφος κατά την τελευταία εικοσαετία είναι η χρήση των *τυπικών μεθόδων (formal methods)*. Με τον όρο αυτό περιγράφουμε γενικά το σύνολο γλωσσών, τεχνικών και εργαλείων που βασίζονται στα μαθηματικά και χρησιμοποιούνται για την καταγραφή των προδιαγραφών και την επαλήθευση συστημάτων υπολογιστών.

Οι τυπικές μέθοδοι προσπαθούν να "εκμεταλλευτούν" τη σαφήνεια και την ακρίβεια των μαθηματικών εργαλείων ώστε να βοηθήσουν τους μηχανικούς σχεδιασμού και ανάπτυξης συστημάτων υπολογιστών να κατανοήσουν καλύτερα τα συστήματα αυτά. Η συνεισφορά τους άπτεται τόσο του σχεδιασμού -

με την καταγραφή των *προδιαγραφών (specifications)* των συστημάτων - όσο και του ελέγχου ορθής λειτουργίας τους - με την *τυπική επαλήθευση (formal verification)*.

Από αυτή την προσπάθεια που έχει ξεκινήσει τα τελευταία χρόνια, εδώ θα εστιάσουμε την προσοχή μας στη χρήση των τυπικών μεθόδων σε συστήματα πραγματικού χρόνου. Όταν μιλάμε στις τυπικές μεθόδους για *συστήματα πραγματικού χρόνου (real time systems)* αναφερόμαστε σε μία "αφαίρεση" που αφορά συστήματα στο χώρο της επιστήμης υπολογιστών και έχει δύο καίρια χαρακτηριστικά.

Είναι προφανές ότι όλα αυτά τα συστήματα λειτουργούν υπό το καθεστώς του χρόνου ("όπως όλα τα ανθρώπινα"). Στα συστήματα όμως πραγματικού χρόνου η χρονική αλληλουχία των εργασιών/διαδικασιών τους αποτελεί κρίσιμο στοιχείο της λειτουργίας τους. Για παράδειγμα, μπορείτε να θεωρήσετε ως τέτοια το σύστημα ελέγχου εναέριας κυκλοφορίας αλλά και ένα έγγραφο πολυμέσων όπως αυτό περιγράφεται στο κεφάλαιο τέσσερα. Το δεύτερο χαρακτηριστικό ενός συστήματος πραγματικού χρόνου προσδιορίζεται από τον τρόπο με τον οποίο θέλουμε να το μελετήσουμε. Πιο συγκεκριμένα, το μοντέλο που εκφράζει ένα σύστημα πραγματικού χρόνου αφορά την περιγραφή του χρόνου όχι ως διακριτού αλλά ως *πυκνού χρόνου (dense time)*, όπως (ή περίπου όπως) είναι στην πραγματικότητα. Το εγχείρημα επομένως με αντικείμενο τα συστήματα πραγματικού χρόνου είναι ενδιαφέρον και μάλλον φιλόδοξο.

Σε αυτήν εδώ την προσέγγιση θα δούμε, λοιπόν, τα συγκεκριμένα εργαλεία που έχουν αναπτυχθεί για την περιγραφή των συστημάτων πραγματικού χρόνου και τις τεχνικές που χρησιμοποιούνται για την τυπική επαλήθευση των ιδιοτήτων τους. Επίσης, θα προσεγγίσουμε και το πρόβλημα του *χρονοπρογραμματισμού (scheduling)* - δηλαδή της εύρεσης και της απόδειξης της ορθότητας ενός προγράμματος ως προς το χρόνο των εργασιών ενός τέτοιου συστήματος. Το πρόβλημα είναι παλιό - όσο και τα συστήματα στα οποία η χρονική αλληλουχία των εργασιών τους παίζει σημαντικό ρόλο - αλλά τώρα αντιμετωπίζεται μέσω των τυπικών μεθόδων με ένα νέο τρόπο.

Πριν όμως προχωρήσουμε στην εξέταση των τυπικών μεθόδων για τα συστήματα πραγματικού χρόνου, είναι σκόπιμο να παρουσιάσουμε γενικά το πλαίσιο τεχνικών των τυπικών μεθόδων. Άλλωστε, οι τεχνικές αυτές χρησιμοποιούνται με ανάλογο τρόπο και στα συστήματα πραγματικού χρόνου.

1.1 Γενικά περί τυπικών μεθόδων

Οι τυπικές μέθοδοι παρέχουν ένα πλαίσιο τεχνικών οι οποίες ακολουθούν όλη τη διαδικασία ανάπτυξης ενός συστήματος - από το σχεδιασμό έως και τον έλεγχο της ορθής λειτουργίας. Μπορούμε να αναφερθούμε σε δύο διαφορετικά στάδια εργασιών: αυτό της διαδικασίας καταγραφής προδιαγραφών και αυτό της τυπικής επαλήθευσης.

Με τον όρο καταγραφή προδιαγραφών αναφερόμαστε στην διαδικασία περιγραφής ενός συστήματος και των επιθυμητών ιδιοτήτων του. Η περιγραφή αυτή γίνεται με μία γλώσσα της οποίας τόσο το συντακτικό όσο η σημασιολογία είναι μαθηματικά ορισμένα.

Έχουν αναπτυχθεί πολλές γλώσσες οι οποίες ανταποκρίνονται κάθε φορά σε συγκεκριμένες εφαρμογές. Ωστόσο τα τελευταία χρόνια υπάρχει η τάση να χρησιμοποιούνται διαφορετικές γλώσσες οι οποίες περιγράφουν διαφορετικές πλευρές του συστήματος και οι οποίες, στη συνέχεια, συγκροτούν ένα ενοποιημένο σύνολο.

Οι γλώσσες αυτές με την μαθηματική σαφήνεια και ακρίβειά τους αποτελούν ένα σημαντικό εργαλείο για την ανάδειξη τυχόν προβλημάτων, ασαφειών ή ασυνεπειών κατά τη σχεδίαση ενός συστήματος. Παράλληλα, μπορούν να λειτουργήσουν ως ένας κοινός τόπος συνεννόησης και επικοινωνίας μεταξύ των διαφορετικών συμμετεχόντων κατά τη δημιουργία ενός συστήματος: του σχεδιαστή, του μηχανικού ανάπτυξης, του τελικού χρήστη.

1.1.1 Τυπική επαλήθευση

Ενώ οι γλώσσες προδιαγραφών δεν θα μας απασχολήσουν περαιτέρω στην παρούσα προσέγγιση, η τυπική επαλήθευση - το άλλο κομμάτι των τυπικών μεθόδων - θα βρεθεί στο κέντρο του προβληματισμού μας.

Με δύο λόγια, η τυπική επαλήθευση μας δίνει τη δυνατότητα της ανάλυσης ενός συστήματος για την ύπαρξη επιθυμητών ιδιοτήτων (ή ανάλογα την απουσία ανεπιθύμητων). Έχουν αναπτυχθεί δύο διαφορετικές μεγάλες ομάδες τεχνικών οι οποίες ακολουθούν διαφορετική προσέγγιση για τη λύση του προβλήματος της επαλήθευσης.

Απόδειξη θεωρήματος

Η τεχνική της *απόδειξης θεωρήματος (theorem proving)* προϋποθέτει ότι τόσο το σύστημα όσο και η ιδιότητα που μας ενδιαφέρει εκφράζονται ως τύποι μίας μαθηματικής (χρονικής) λογικής. Η λογική βέβαια αυτή είναι ένα τυπικό σύστημα, που αποτελείται από ένα σύνολο αξιωμάτων και ένα σύνολο συμπερασματικών κανόνων. Η τεχνική λοιπόν αποτελεί ακριβώς την εύρεση μιας απόδειξης της υπό εξέταση ιδιότητας στα πλαίσια του συγκεκριμένου τυπικού συστήματος.

Η τεχνική αυτή ουσιαστικά προσπαθεί με αυτοματοποιημένο τρόπο να μιμηθεί τον τρόπο που κάνουμε μαθηματικές αποδείξεις - ξεκινά από τα αξιώματα καθώς και τον τύπο που εκφράζει το σύστημα και το κάθε βήμα της απόδειξης βασίζεται στους συμπερασματικούς κανόνες έως ότου φτάσουμε στο τύπο που εκφράζει την ιδιότητα.

Υπάρχουν αρκετά παραδείγματα εφαρμογών όπου χρησιμοποιείται αυτή η τεχνική για την επαλήθευση συστημάτων τόσο για εφαρμογές στη βιομηχανία όσο και για περιπτώσεις επαλήθευσης γνωστών μαθηματικών αποδείξεων. Στο [CW96] υπάρχει μία συνοπτική παράθεση τέτοιων ενδιαφερόντων παραδειγμάτων.

Έλεγχος μοντέλου

Η τεχνική του *ελέγχου μοντέλου (model checking)* είναι σαφέστατα πιο διαδεδομένη και αφορά την οικοδόμηση ενός πεπερασμένου μοντέλου του συστήματος και στη συνέχεια τον έλεγχο για το αν η ιδιότητα που μας ενδιαφέρει ισχύει σε αυτό το μοντέλο. Βασικός φορμαλισμός σε αυτή την τεχνική είναι το αυτόματο το οποίο περιγράφει το υπο εξέταση σύστημα. Ως προς την ιδιότητα σε κάποιες μεθόδους περιγράφεται και αυτή μέσω ενός αυτομάτου ενώ σε άλλες προσεγγίσεις μέσω ενός τύπου μίας χρονικής λογικής.

Η τεχνική χρησιμοποιεί διάφορους αλγορίθμους των οποίων κοινό σημείο αποτελεί το γεγονός ότι ψάχνουν εξαντλητικά όλο το μοντέλο το οποίο αποτελείται από καταστάσεις στις οποίες μπορεί να περιέλθει το σύστημα και τις πιθανές μεταβάσεις μεταξύ των καταστάσεων αυτών. Η τεχνική βασίζεται στο γεγονός ότι το μοντέλο είναι πεπερασμένο κι επομένως, η αναζήτηση επιτυχώς ή ανεπιτυχώς θα τερματίσει.

Το ζητούμενο σε αυτή την τεχνική είναι κατά κύριο λόγο η βελτιστοποίηση των αλγορίθμων και μεθόδων αναζήτησης ώστε να είναι σε θέση να ανταποκρίνονται σε περιπτώσεις μεγάλων χώρων αναζήτησης. Αυτό είναι και το μεγαλύτερο μειονέκτημα αυτή της τεχνικής, το λεγόμενο *πρόβλημα της έκρηξης καταστάσεων (state explosion problem)*. Το πρόβλημα αυτό έγκειται στο γεγονός ότι ο αριθμός των πιθανών καταστάσεων του μοντέλου ενός συστήματος αυξάνεται εκθετικά ως προς τον αριθμό των μεταβλητών του συστήματος. Να αναφέρουμε ότι ένα διακριτό σύστημα με k μεταβλητές με δύο πιθανές τιμές (αληθές ή ψευδές) θα πρέπει να έχει στο μοντέλο του 2^k καταστάσεις.

Το πρόβλημα αυτό είναι εγγενές στην τεχνική ελέγχου μοντέλων και η αντιμετώπισή του έχει οδηγήσει στην ανάπτυξη διαφόρων τεχνικών οι οποίες προσπαθούν να μειώσουν τον αριθμό των καταστάσεων με την απαλοιφή περιττών καταστάσεων ή τη συμβολική αναπαράστασή τους.

Η τεχνική ελέγχου μοντέλων έχει χρησιμοποιηθεί ευρέως - με σημαντικότερες υλοποιήσεις την επαλήθευση πρωτοκόλλων όπως κάποια πρότυπα του IEEE και το ISDN/IUPP. (Περισσότερες εφαρμογές μπορεί να δει κανείς στο [CW96].)

Στο κεφάλαιο τρία, θα δούμε πιο συγκεκριμένα πως αυτές οι τεχνικές της απόδειξης θεωρήματος και του ελέγχου μοντέλων χρησιμοποιούνται στην περίπτωση που μελετάμε ζητήματα τυπικής επαλήθευσης για συστήματα πραγματικού χρόνου.

Κλείνοντας την εισαγωγή, ας δούμε συνοπτικά την οργάνωση της θεματολογίας της παρουσίασης που θα ακολουθήσουμε εδώ. Στο κεφάλαιο δύο θα αναφερθούμε στους μαθηματικούς φορμαλισμούς τους οποίους χρησιμοποιούμε για να μιλήσουμε για ένα σύστημα πραγματικού χρόνου. Θα παρουσιάσουμε εν συντομία τα κυριότερα θεωρητικά αποτελέσματα που σχετίζονται με αυτούς. Στο κεφάλαιο τρία, θα χρησιμοποιήσουμε αυτούς τους φορμαλισμούς για να εκφράσουμε το πρόβλημα της επαλήθευσης για συστήματα πραγματικού χρόνου και θα εκμεταλλευτούμε το θεωρητικό τους οπλοστάσιο για να αντιμετωπίσουμε μαθηματικά το πρόβλημα της επαλήθευσης. Ενώ στο κεφάλαιο τέσσερα, θα παρουσιάσουμε μία συγκεκριμένη προσέγγιση σε ένα πρόβλημα επαλήθευσης - αυτού του χρονοπρογραμματισμού - κάνοντας χρήση των μεθόδων που θα έχουμε εξηγήσει στα προηγούμενα κεφάλαια. Την παρουσίαση μας θα κλείσει μία σύντομη αποτίμηση του ενδιαφέροντος αλλά και των αποτελεσμάτων της ερευνητικής αυτής περιοχής.

Κεφάλαιο 2

Θεωρητικά εργαλεία

Σε αυτό το κεφάλαιο, θα προσπαθήσουμε να περιγράψουμε δύο σημαντικές θεωρητικές κατασκευές: τα συγχρονισμένα αυτόματα και τις λογικές πραγματικού χρόνου των οποίων η έμπνευση προέκυψε από τη θεωρία αυτομάτων και τη λογική αντίστοιχα.

Οι χρονικές λογικές είναι ουσιαστικά τροπικές λογικές και ως τέτοιες η καταγωγή τους φτάνει μέχρι τη φιλοσοφία. Απαντώνται στη μαθηματική βιβλιογραφία από τις αρχές του 20ου αιώνα, με το ενδιαφέρον γι' αυτές να αναζωπυρώνεται από το δεύτερο μισό του και στο εξής. Ανάλογο παρελθόν έχουν και τα αυτόματα τα οποία αποτελούν δομικό στοιχείο της θεωρίας υπολογισμού. Το ενδιαφέρον και για τους δύο αυτούς φορμαλισμούς είναι ιδιαίτερα ευρύ και έχει αποδώσει πολλές ενδιαφέρουσες ιδέες.

Η εμπλοκή των χρονικών λογικών στην τυπική επαλήθευση, ως φορμαλισμού για την περιγραφή ιδιοτήτων των συστημάτων ως προς το χρόνο, ήρθε αργότερα στη δεκαετία του '70.[1] Τα αυτόματα επίσης αποτελούν αναπόσπαστο κομμάτι της τυπικής επαλήθευσης ως φορμαλισμός περιγραφής συστημάτων ειδικότερα σε τεχνικές ελέγχου μοντέλου.

Τόσο οι λογικές πραγματικού χρόνου όσο και τα συγχρονισμένα αυτόματα προέκυψαν από την αρχή της ανάπτυξης των τεχνικών της τυπικής επαλήθευσης για συστήματα πραγματικού χρόνου ως "φυσικές" επεκτάσεις των παραπάνω

[1] A.Pnueli, The temporal logic of programs, *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, 46-57, IEEE Computer Society Press, 1977.

φορμαλισμών. Και τα δύο εργαλεία είναι πολύ ενδιαφέροντα στην προσπάθειά τους να αιχμαλωτίσουν την έννοια του “χρόνου”.

Για την περιγραφή μας για τα συγχρονισμένα αυτόματα, η βασικότερη αναφορά ήταν το [AD94] το οποίο αναπτύσσει μία ολόκληρη θεωρία συγχρονισμένων αυτομάτων με σκοπό τη χρήση τους από την τυπική επαλήθευση. Για τις λογικές πραγματικού χρόνου, η κύρια αναφορά ήταν το [AH91], αλλά χρησιμοποιήθηκαν και στοιχεία από τα [E90],[ACD90].

2.1 Συγχρονισμένα αυτόματα

Ένα αυτόματο, όπως είναι γνωστό από την θεωρία αυτομάτων, έχει μία μόνο “διάσταση”, η οποία μπορούμε να θεωρήσουμε ότι είναι ο “χώρος” - έννοια που είναι και διαισθητικά προσιτή αφού τα αυτόματα απεικονίζονται ως γράφοι. Οι καταστάσεις του αυτομάτου μπορούν να ειδωθούν ως αντικείμενα στο χώρο και οι μεταβάσεις μεταξύ τους ως μετακινήσεις από το ένα στο άλλο.

Το συγχρονισμένο αυτόματο[2] είναι ένα πεπερασμένο αυτόματο το οποίο έχει μία “διάσταση” επιπλέον, την οποία μπορούμε να χαρακτηρίσουμε ως “χρόνο”. Η διάσταση αυτή εκφράζεται με ένα πεπερασμένο σύνολο “ρολογιών” τα οποία με τις τιμές τους προσδιορίζουν κάθε στιγμή τη θέση του αυτομάτου σε αυτή τη διάσταση. Κάθε κατάστασή του είναι ένα ζεύγος που αποτελείται από ένα αντικείμενο στο χώρο - τη διακριτή κατάσταση - και ένα αντικείμενο στο χρόνο - ένα σύνολο τιμών των ρολογιών -, ένα ζεύγος δηλαδή που προσδιορίζει πλήρως τις δύο διαστάσεις του. Οι μεταβάσεις του επίσης επιτρέπουν την κίνηση από το ένα αντικείμενο στο άλλο ως προς τη μία διάσταση ή ως προς και τις δύο - μπορούμε δηλαδή να μετακινηθούμε στο χώρο χωρίς να αλλάξουμε τίποτα ως προς το χρόνο και το αντίθετο.

Στο κλασικό μονοδιάστατο αυτόματο οι μετακινήσεις γίνονται υπό το καθεστώς ελέγχου της συνάρτησης μετάβασης και του συμβόλου της εισόδου που διαβάζουμε κάθε φορά. Στο συγχρονισμένο αυτόματο, η συνάρτηση μετάβασης περιλαμβάνει τώρα και στοιχεία για τη δεύτερη διάσταση, τη διάσταση του χρόνου. Ο έλεγχος στη διάσταση του χώρου θα εξακολουθεί να είναι το

[2] Ο όρος είναι αδόκιμος και αποπειράται να αποδώσει τον αντίστοιχο αγγλικό “timed automaton”.

στοιχείο εισόδου ενώ ο έλεγχος στη διάσταση του χρόνου θα είναι μία σειρά περιορισμών στις τιμές των ρολογιών οι οποίες πρέπει να ικανοποιούνται για να μπορεί να πραγματοποιηθεί η αντίστοιχη μετάβαση.

Ορισμός 2.1.1 *Έστω Σ ένα πεπερασμένο αλφάβητο. Ορίζουμε σε αυτό ένα* συγχρονισμένο αυτόματο - ΣΑ *(Timed Automaton - TA) $\mathcal{A} = (S, S_0, X, E)$, όπου*

- *S είναι ένα πεπερασμένο σύνολο διακριτών καταστάσεων*
- *$S_0 \subseteq S$ είναι το σύνολο των αρχικών διακριτών καταστάσεων του $\mathcal{A}$.*
- *$X = \{x_1, x_3, \ldots, x_m\}$ είναι ένα σύνολο μεταβλητών που παίρνουν θετικές πραγματικές τιμές και ονομάζονται* ρολόγια.
- *$E \subseteq S \times \Sigma \times S \times 2^X \times G_X$ είναι ένα σύνολο ακμών, ένα σύνολο μεταβάσεων.*
  Μία ακμή $e = (s, \sigma, s', r, g)$ αναπαριστά μία μετάβαση από την διακριτή κατάσταση s στην διακριτή επίσης κατάσταση s' όταν το σύμβολο της εισόδου είναι σ. Το r είναι ένα υποσύνολο του συνόλου των ρολογιών X και εκφράζει τα ρολόγια τα οποία θα μηδενιστούν κατά την εκτέλεση της μετάβασης και το $g \in G_X$.
  Το G_X είναι ένα σύνολο κατηγορημάτων-περιορισμών ως προς το σύνολο των ρολογιών X. Τα κατηγορήματα αυτά ορίζονται ως εξής: $g ::= x \leq c | x < c | x - y \leq c | x - y < c | g_1 \wedge g_2 | \neg g$ όπου $x, y \in X$ και c είναι ακέραιος.[3]

Η *κατάσταση (state)* ενός συγχρονισμένου αυτομάτου είναι ένα ζεύγος $\langle s, v \rangle$ όπου s είναι μία διακριτή κατάσταση (δηλαδή, $s \in S$) και v είναι μία αποτίμηση των τιμών των ρολογιών (δηλαδή, $v \in R_+^m$)[4]. Στο συγχρονισμένο αυτόματο, δεν μας αρκεί να ξέρουμε τη διακριτή κατάσταση στην οποία βρίσκεται το αυτόματο. Κάθε κατάσταση χαρακτηρίζεται και από τις τιμές όλων των ρολογιών που χαρακτηρίζουν τη λειτουργία του αυτομάτου.

[3] Ο κάθε περιορισμός ουσιαστικά ορίζει ένα σύνολο επιτρεπτών αποτιμήσεων των τιμών του αντίστοιχου ρολογιού και όλοι μαζί ορίζουν το σύνολο των επιτρεπτών αποτιμήσεων για όλα τα ρολόγια.

[4] Με R_+ συμβολίζονται οι θετικοί πραγματικοί αριθμοί.

Είναι προφανές ότι, σύμφωνα με αυτό τον ορισμό, δύο καταστάσεις ενός συγχρονισμένου αυτομάτου μπορούν να διαφέρουν είτε ως προς το διακριτό τους μέρος - δηλαδή, ως προς τη διακριτή κατάσταση - είτε ως προς την αποτίμηση των ρολογιών - δηλαδή, ως προς τις τιμές που παίρνει ένα ή περισσότερα ρολόγια - είτε ως προς και τα δύο μέρη τους.

Ανάλογα με το κλασικό αυτόματο, μπορούν να οριστούν και οι τελικές καταστάσεις του συγχρονισμένου αυτομάτου οι οποίες είναι ένα υποσύνολο (ή ένα στοιχείο) του συνόλου των διακριτών καταστάσεών του.

Ορισμός 2.1.2 *Ένας υπολογισμός p σε ένα συγχρονισμένο αυτόματο $\mathcal{A} = (S, S_0, X, E)$ ως προς ένα αλφάβητο Σ, είναι μία άπειρη ακολουθία της μορφής*

$$p : \langle s_0, v_0 \rangle \rightarrow_{\sigma_1}^{t_1} \langle s_1, v_1 \rangle \rightarrow_{\sigma_2}^{t_2} \langle s_2, v_2 \rangle \rightarrow_{\sigma_3}^{t_3} \cdots$$

όπου $s_i \in S$, v_i είναι αποτιμήσεις των τιμών των ρολογιών, $\sigma_i \in \Sigma$ ενώ οι t_i ορίζουν μία ακολουθία χρονικών στιγμών με δύο σημαντικές ιδιότητες: η ακολουθία είναι αυστηρά αύξουσα και συνεχίζεται έπ' άπειρον.
Οι ιδιότητες που πρέπει να πληροί η παραπάνω ακολουθία για να χαρακτηριστεί ως υπολογισμός είναι οι εξής:

- *$s_0 \in S_0$, δηλαδή το διακριτό μέρος της πρώτης κατάστασης να είναι αρχική διακριτή κατάσταση και $v_x = 0, \forall x \in X$, δηλαδή όλα τα ρολόγια να έχουν αρχικά τη τιμή μηδέν.*

- *Για κάθε βήμα της i, υπάρχει μία ακμή $(s_{i-1}, \sigma_i, s_i, r_i, g_i)$ που ανήκει στο σύνολο E τέτοια ώστε:*

  (α) η αποτίμηση $v_{i-1} + t_i - t_{i-1}$ ικανοποιεί τον περιορισμό g_i.

  (β) η αποτίμηση v_i ισούται με την αποτίμηση $[r_i := 0](v_{i-1} + t_i - t_{i-1})$

Για κάθε αποτίμηση v, ορίζουμε τις αποτιμήσεις $v \pm t$ και $v[r := t]$ ως εξής:
$(v \pm t)(x) = v(x) \pm t, \forall x$ και $v[r := t](x) = \begin{cases} t & x = r \\ v(x) & x \neq r \end{cases}$

Ο παραπάνω ορισμός του υπολογισμού δεν αποκλείει την περίπτωση για κάποιο i να έχουμε $s_i = s_{i-1}$, δηλαδή να μην έχουμε αλλαγή ως προς τη διακριτή κατάσταση. Τότε έχουμε μία *χρονική μετάβαση* (time transition) η οποία έχει τη μορφή

$$\langle s, v\rangle \to^{t} \langle s, v+t\rangle$$

όπου $t \in R_+$ και εκφράζει την ποσότητα του χρόνου που περνάει και $v+t$ είναι μία νέα αποτίμηση των ρολογιών του αυτομάτου η οποία αυξάνει τις τιμές τους κατά t και η οποία ικανοποιεί όλους τους περιορισμούς οι οποίοι αφορούν την διακριτή κατάσταση s, δηλαδή: (όλα τα $g : (s, \sigma, s', r, g) \in E, \forall s', \sigma, r$). Η μετάβαση αυτή επιτρέπει στο χρόνο να "κυλάει" σεβόμενος όμως πάντα τους κανόνες του παιχνιδιού, δηλαδή μην παραβιάζοντας τους περιορισμούς που καθορίζουν τα χρονικά όρια της κάθε κατάστασης.

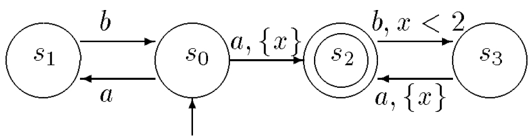
Αντίστοιχα, μία *διακριτή μετάβαση* (discrete transition) έχει τη μορφή

$$\langle s, v\rangle \to_{e} \langle s', v'\rangle$$

όπου $e \in E$ τέτοια ώστε $e = (s, \sigma, s', r, g)$ για την οποία ισχύει ότι η αποτίμηση v ικανοποιεί τον περιορισμό g και η $v' = v[r := 0]$ ικανοποιεί όλους τους περιορισμούς οι οποίοι αφορούν τη διακριτή κατάσταση s', δηλαδή: (όλα τα g' : $(s', \sigma', s'', r', g') \in E, \forall s'', a', r'$).

Να παρατηρήσουμε εδώ ότι ένα αυτόματο, όπως το γνωρίζαμε μέχρι τώρα, είναι ένα συγχρονισμένο αυτόματο σε κάθε ακμή του οποίου το σύνολο των ρολογιών που μηδενίζονται είναι το κενό σύνολο ($r = \emptyset$) - ουσιαστικά το ίδιο το X είναι το κενό σύνολο - και δεν υπάρχουν περιορισμοί ως προς την χρονική αλληλουχία δηλαδή ας πούμε $g :=$ **true**.

Ας προσπαθήσουμε τώρα να περιγράψουμε τη λειτουργία ενός συγχρονισμένου αυτόματου. Το αυτόματο έχει ένα σύνολο ρολογιών τα οποία βέβαια λειτουργούν μετρώντας το χρόνο με τον ίδιο ρυθμό. Στην αρχή, όλα είναι μηδενισμένα. Κάθε μετάβαση πραγματοποιείται μόνο εφόσον ικανοποιεί τον περιορισμό που αντιστοιχεί σε αυτήν. Ταυτόχρονα, με τη μετάβαση γίνεται και ο μηδενισμός των ρολογιών που προβλέπονται κάθε φορά. Με αυτό το μηχανισμό, έχουμε κάποιο είδος "ιστορικής μνήμης" ως προς τις καταστάσεις από τις οποίες περνάμε - τα ρολόγια λειτουργούν, θα μπορούσε να πει κανείς, ως "χρονόμετρα" .



Σχήμα 2.1 Ένα παράδειγμα συγχρονισμένου αυτομάτου

Στο σχήμα 2.1, φαίνεται η απεικόνιση ενός συγχρονισμένου αυτομάτου η οποία μοιάζει με αυτή που έχουμε συνηθίσει. Το αλφάβητο του αυτομάτου είναι το $\{a, b\}$, η αρχική διακριτή κατάσταση είναι η s_0, η τελική διακριτή κατάσταση είναι η s_2 και το σύνολο των ρολογιών του αυτομάτου είναι το $\{x\}$. Το αυτόματο ξεκινά τη λειτουργία του από την s_0 και πηγαίνει από αυτήν στην s_1 και ξανά πίσω στην s_0 ανάλογα με την είσοδο. Μη ντετερμινιστικά, κάποια στιγμή περνάει στην s_2 μηδενίζοντας το ρολόι x αν στην είσοδο διαβάσει το a. Σύμφωνα με τους περιορισμούς που έχουμε στο αυτόματο, είμαστε σίγουροι ότι το b που θα ακολουθήσει μετά το a θα εμφανιστεί μέσα σε δύο χρονικές μονάδες. Αυτό σημαίνει, ότι το αυτόματο ουσιαστικά μοντελοποιεί την ιδιότητα ενός συστήματος η οποία είναι η εξής: μετά το a, το b θα εμφανιστεί "τελικά" πάντα σε χρόνο λιγότερο από δύο χρονικές μονάδες. Θα δούμε αργότερα πως συνδέεται μία τέτοια ιδιότητα με το πρόβλημα της τυπικής επαλήθευσης.

Με βάση το συγχρονισμένο αυτόματο, αναπτύσσεται μία θεωρία συγχρονισμένων αυτομάτων ανάλογη της θεωρίας αυτομάτων που ήδη ξέρουμε. Μπορούμε να ορίσουμε, επομένως, *συγχρονισμένα Büchi αυτόματα - ΣΒΑ (timed Büchi automata)* καθώς και *συγχρονισμένες κανονικές γλώσσες (timed regular languages)*. Τα ΣΒΑ είναι και αυτά που μας ενδιαφέρουν κατά κανόνα στην τυπική επαλήθευση αφού οι συμπεριφορές των συστημάτων μοντελοποιούνται από άπειρες λέξεις που ακριβώς γίνονται αποδεκτές από αυτά τα αυτόματα. Επίσης, μπορούμε να ορίσουμε *συγχρονισμένα Muller αυτόματα -ΣΜΑ (timed Muller automata)* και μάλιστα, αποδεικνύεται η ισοδυναμία των Büchi ΣΑ με τα Muller ΣΑ, σε αναλογία με την ισοδυναμία των αντίστοιχων αυτομάτων χωρίς τη διάσταση του χρόνου.

Επίσης, ανάλογα μπορούμε να ορίσουμε και τις ντετερμινιστικές εκδοχές των παραπάνω αυτομάτων, δηλαδή τα *ντετερμινιστικά συγχρονισμένα Büchi αυτόματα - ΝΣΒΑ (deterministic timed Büchi automata)* και τα *ντετερμινιστικά συγχρονισμένα Muller αυτόματα - ΝΣΜΑ (deterministic timed Muller automata)*. Ο ντετερμινισμός καθορίζεται στα συγχρονισμένα αυτόματα από τη μοναδικότητα της αρχικής κατάστασης και τον αμοιβαίο αποκλεισμό των περιορισμών σε όλες τις μεταβάσεις με κοινή αρχική κατάσταση.

Το ενδιαφέρον βέβαια σημείο αυτή της θεωρίας από τη σκοπιά των τυπικών μεθόδων είναι η συμπεριφορά αυτών των αυτομάτων ως προς τις πράξεις της ένωσης, της τομής και του συμπληρώματος. Τα σημαντικότερα αποτελέσματα σχετικά με αυτές συνοψίζονται στα εξής:

- Η κλάση των ντετερμινιστικών συγχρονισμένων Muller αυτομάτων είναι κλειστή ως προς την ένωση, την τομή και το συμπλήρωμα, όπως και στην περίπτωση των αντίστοιχων μη συγχρονισμένων.

- Η κλάση των ντετερμινιστικών συγχρονισμένων Büchi αυτομάτων είναι κλειστή ως προς την ένωση και την τομή αλλά όχι ως προς το συμπλήρωμα, όπως και στην περίπτωση των αντίστοιχων μη συγχρονισμένων αυτομάτων.

- Η κλάση των συγχρονισμένων Büchi αυτομάτων, άρα και των συγχρονισμένων Muller αυτομάτων, είναι κλειστή ως προς την ένωση και την τομή αλλά όχι ως προς το συμπλήρωμα.

- Τέλος, οι σχέσεις μεταξύ των διαφόρων κλάσεων των συγχρονισμένων αυτομάτων έχουν ως εξής: $\mathrm{N\Sigma BA} \subset \mathrm{N\Sigma MA} \subset \mathrm{\Sigma MA} = \mathrm{\Sigma BA}$.

Μιλώντας μέχρι τώρα για τα συγχρονισμένα αυτόματα κατά κάποιο τρόπο αποσιωπήσαμε το γεγονός (ίσως γιατί θα προτιμούσαμε να το ξεχάσουμε) ότι ο αριθμός των καταστάσεων ενός ΣΑ είναι άπειρος. Αυτό προκύπτει λόγω του γεγονότος ότι η κάθε κατάσταση "εμπεριέχει" μία χρονική στιγμή ως το σύνολο των τιμών όλων των ρολογιών - και έχουμε κάνει την παραδοχή ότι ο χρόνος μας είναι "πυκνός", συνεχής.

Από την άλλη πλευρά, η εισαγωγή μας εδώ των συγχρονισμένων αυτομάτων έχει γίνει με τον απώτερο σκοπό να τα χρησιμοποιήσουμε ως εργαλεία στην τυπική επαλήθευση. Κι αν υπάρχει μία λέξη που δε συμπαθεί οποιαδήποτε αλγοριθμική διαδικασία, αυτή είναι η λέξη "άπειρο"! Ο τρόπος που προσεγγίζεται αυτό το πρόβλημα είναι, με βάση το συγχρονισμένο αυτόματο, η παραγωγή ενός νέου αυτομάτου με πεπερασμένο αριθμό καταστάσεων το οποίο βέβαια εξακολουθεί να διατηρεί τις ιδιότητες του αρχικού αυτομάτου που μας ενδιαφέρουν. Θα δούμε περισσότερες λεπτομέρειες σχετικά με αυτό τον επονομαζόμενο "γράφο περιοχής" στην ενότητα 3.2.

Ωστόσο, αξίζει εδώ να αναφέρουμε ένα αποτέλεσμα πολυπλοκότητας σχετικά με τα συγχρονισμένα αυτόματα που απορρέει από τη δυνατότητα κατασκευής του γράφου αυτού.

Θεώρημα 2.1.1 *Το πρόβλημα απόφασης για το αν η γλώσσα ενός συγχρονισμένου αυτομάτου είναι κενή είναι $PSPACE$-πλήρες.*

Απόδειξη. Σχηματικά, η απόδειξη αποτελείται από δύο μέρη: (α) την εύρεση ενός μονοπατιού μη ντετερμινιστικά και με χρήση πολυωνυμικού χώρου το οποίο αποδεικνύει ότι η γλώσσα δεν είναι κενή, αποδεικνύοντας έτσι ότι το πρόβλημα ανήκει στην κλάση $PSPACE$, και (β) την αναγωγή ενός γνωστού $PSPACE$-πλήρους προβλήματος σε αυτό το πρόβλημα.

2.1.1 Συγχρονισμένα αυτόματα με προθεσμίες

Τα συγχρονισμένα αυτόματα, όπως ορίστηκαν παραπάνω, είναι μια πολύ ενδιαφέρουσα πρόταση θεωρητικού εργαλείου η οποία αφήνει πολλά περιθώρια αλλαγών και τροποποιήσεων με σκοπό την έκφραση συγκεκριμένων ιδιοτήτων ή την χρήση της σε συγκεκριμένες εφαρμογές. Ένα τέτοιο παράδειγμα είναι τα συγχρονισμένα αυτόματα με προθεσμίες τα οποία θα παρουσιάσουμε εδώ για δύο λόγους.

Πρώτον, είναι μια ενδιαφέρουσα παραλλαγή του αρχικού ορισμού η οποία με εύκολο τρόπο μας δείχνει τους διαφορετικούς τρόπους με τους οποίους μπορούμε να χρησιμοποιήσουμε ένα θεωρητικό εργαλείο. Και δεύτερον, μας είναι απαραίτητο για την παρουσίαση της τεχνικής χρονοπρογραμματισμού στο κεφάλαιο τέσσερα. (Ο ορισμός που δίνεται εδώ καθώς και περισσότερα στοιχεία ως προς τα αυτόματα με προθεσμία μπορούν να βρεθούν στα [BST98] και [AGP99].)

Ορισμός 2.1.3 *Ένα συγχρονισμένο αυτόματο με προθεσμίες-ΣΑΠ (Timed Automaton with Deadlines-TAD)* $\mathcal{A} = (S, \rightarrow, A, X, h)$ *απαρτίζεται από τα εξής στοιχεία:*

- S *είναι ένα πεπερασμένο σύνολο διακριτών καταστάσεων*
- A *είναι ένα πεπερασμένο λεξιλόγιο πράξεων*
- $\rightarrow \subseteq S \times A \times S$ *είναι μία σχέση μετάβασης.*
- $X = \{x_1, x_3, \ldots, x_m\}$ *είναι ένα σύνολο μεταβλητών που παίρνουν θετικές πραγματικές τιμές και ονομάζονται ρολόγια.*
- *Τέλος,* h *είναι μία συνάρτηση η οποία εκφράζει όλη την πληροφορία σχετικά με το χρόνο. Η συνάρτηση αυτή αντιστοιχεί κάθε στοιχείο της σχέσης μετάβασης σε ένα στοιχείο το οποίο έχει και πληροφορία χρόνου.*
  Δηλαδή, $h(s, a, s') = h(s, (a, g, d, r), s')$*, όπου τα* g, d *είναι ένα κατηγόρημα - περιορισμός και ένα κατηγόρημα - προθεσμία αντίστοιχα, τα οποία ορίζονται συντακτικά ως εξής* $g|d ::= x \leq c|x < c|x - y \leq c|x - y < c|g \wedge g|\neg g$ *όπου* $x, y \in X$ *και* c *είναι ακέραιος και* $r \subseteq X$*. Η σχέση μεταξύ των δύο σε κάθε μετάβαση είναι* $d \Rightarrow g$*.*

Το πρώτο πράγμα που παρατηρεί κανείς στον παραπάνω ορισμό είναι ότι, κατά κάποιο τρόπο, διαχωρίζεται το διακριτό μέρος του αυτομάτου από αυτό με την πληροφορία του χρόνου. Αυτό επιτυγχάνεται με τη συνάρτηση h η οποία μας επιτρέπει να "δώσουμε" την πληροφορία σχετικά με τους χρονικούς περιορισμούς εκ των υστέρων στο αυτόματο. Το γεγονός αυτό μας δίνει μία ευελιξία της οποίας η χρησιμότητα θα φανεί όταν θα χρησιμοποιήσουμε τα συγχρονισμένα αυτόματα με προθεσμία.

Το δεύτερο σημείο στο οποίο διαφοροποιούνται αυτά τα αυτόματα σε σχέση με τον ορισμό 2.1.1 είναι ότι η κάθε ακμή έχει εκτός από ένα κατηγόρημα-περιορισμό και ένα κατηγόρημα-προθεσμία. Το κατηγόρημα-περιορισμός επιτρέπει τις μεταβάσεις που το ικανοποιούν ενώ το κατηγόρημα-προθεσμία χαρακτηρίζει αυτές τις χρονικές μεταβάσεις σε μία διακριτή κατάσταση που δεν το παραβιάζουν. Η σχέση μεταξύ των δύο κατηγορημάτων μοντελοποιεί κατά κάποιο τρόπο το πόσο επείγουσα είναι μία μετάβαση.

2.2 Λογικές πραγματικού χρόνου

Μία χρονική λογική είναι μία τροπική λογική στην οποία οι τροπικοί τελεστές "ερμηνεύονται" ως προς την ισχύ τους κατά τη διάρκεια του χρόνου. Πιο συγκεκριμένα, ο τροπικός τελεστής $\Box$ ερμηνεύεται ως "πάντα" ενώ ο τελεστής $\Diamond$ ως "τελικά". (Για μία συνολική θεωρητική παρουσίαση των χρονικών λογικών, ενδιαφέρον είναι το [E90].)

Οι χρονικές λογικές είναι αρκετά πλούσιες και εκφραστικές αλλά, όπως φαίνεται, δεν είναι επαρκείς για την χρησιμοποίηση τους στη μοντελοποίηση συστημάτων πραγματικού χρόνου. Ας δούμε που οφείλεται η αδυναμία τους και πως την ξεπερνάμε για να οδηγηθούμε στις λογικές πραγματικού χρόνου.

Οι χρονικές λογικές, όπως όλες οι λογικές, μπορεί να είναι προτασιακές ή πρωτοβάθμιες. Μπορεί δηλαδή, εκτός των χρονικών τελεστών τους, να περιλαμβάνουν το προτασιακό ή το πρωτοβάθμιο κομμάτι της κλασικής λογικής. Στην παρούσα παρουσίαση, θα χρησιμοποιήσουμε προτασιακές χρονικές λογικές οι οποίες είναι πιο απλές αλλά εξίσου εκφραστικές για τους σκοπούς που τις θέλουμε εδώ.

Ένα άλλο χαρακτηριστικό που διαφοροποιεί τις χρονικές λογικές μεταξύ τους είναι η αντίληψη που έχουν για το χρόνο. Οι λογικές γραμμικού χρόνου θεωρούν ότι κάθε χρονική στιγμή έχει μία και μοναδική επόμενη χρονική στιγμή, δηλαδή η χρονική ακολουθία εκτείνεται σε μία γραμμή. Αντίθετα, οι λογικές διακλαδιζόμενου χρόνου θεωρούν ότι κάθε χρονική στιγμή μπορεί να έχει παραπάνω από μία επόμενες χρονικές στιγμές, άρα η χρονική ακολουθία για αυτές τις λογικές είναι ένα δέντρο. Στην τυπική επαλήθευση, χρησιμοποιούνται και τα δύο είδη - το καθένα για να εκφράσει διαφορετικές ιδιότητες στην διάρκεια του χρόνου.

Τέλος, το πιο σημαντικό για μας εδώ είναι το γεγονός ότι οι χρονικές λογικές μπορούν να εκφράσουν μόνο "ποιοτικές" ιδιότητες του χρόνου - κάτι θα συμβεί "τελικά" κ. λ. π. Δεν μπορούν να εκφράσουν "ποσοτικές" ιδιότητες

- κάτι θα συμβεί "τελικά" μετά από δύο μονάδες χρόνου. Δεν μπορούμε να εκφράσουμε με το λεξιλόγιο τους τον περιορισμό "μετά από δύο μονάδες χρόνου". Τις εμπλουτίζουμε λοιπόν με τα κατάλληλα εκφραστικά εργαλεία γι' αυτό το σκοπό. Σε αυτές τις χρονικές λογικές, το σημασιολογικό πλαίσιο μέσα στο οποίο ερμηνεύονται θεωρεί το χρόνο άλλοτε διακριτό και άλλοτε συνεχή.

Θα παρουσιάσουμε τώρα μία από τις ενδιαφέρουσες - τόσο για το θέμα της επαλήθευσης συστημάτων πραγματικού χρόνου όσο και γενικότερα - προτασιακές χρονικές λογικές διακλαδιζόμενου χρόνου, την *Computational Tree Logic (CTL)*. Υπάρχουν διάφοροι ισοδύναμοι τρόποι να οριστεί αυτή η λογική - θα ακολουθήσουμε εδώ τον ορισμό που δίνεται στο [ACD90].

Συντακτικό της CTL. Έστω ένα σύνολο ατομικών προτασιακών τύπων P τέτοιο ώστε $p \in P$. Οι τύποι της CTL ορίζονται αναδρομικά ως εξής:

$$\phi := p \mid \mathbf{false} \mid \phi_1 \rightarrow \phi_2 \mid \exists \bigcirc \phi \mid \exists(\phi_1 \mathcal{U} \phi_2) \mid \forall(\phi_1 \mathcal{U} \phi_2)$$

Ο τελεστής $\bigcirc$ ερμηνεύεται ως "στην αμέσως διάδοχη κατάσταση" ενώ ο τελεστής $\mathcal{U}$ ερμηνεύεται ως "μέχρι".[5] Οι ποσοδείκτες $\exists$ και $\forall$ αναφέρονται σε μονοπάτια. Οι τελεστές $\Box$ και $\Diamond$ σε αυτή τη λογική, "γράφονται" ως εξής:

$$\exists\Diamond\phi = \exists(true\mathcal{U}\phi), \quad \exists\Box\phi = \neg\forall\Diamond\neg\phi$$

Σημασιολογία της CTL. Η σημασιολογία της CTL ορίζεται ως προς ένα μοντέλο Kripke $\mathcal{M} = (S, \mu, E)$ όπου S είναι ένα αριθμήσιμο σύνολο καταστάσεων, $\mu : S \rightarrow 2^P$ μία αποτίμηση τιμών αλήθειας στις ατομικές προτάσεις κάθε κατάστασης και E είναι μία διμερής σχέση μεταξύ των καταστάσεων η οποία εκφράζει τις πιθανές μεταβάσεις. Σε ένα τέτοιο μοντέλο, ένα *μονοπάτι* είναι μία άπειρη ακολουθία καταστάσεων $(s_0, s_1, \ldots)$ τέτοιο ώστε $(s_i, s_{i+1}) \in E$ για όλα τα $i \geq 0$, δηλαδή μία ακολουθία καταστάσεων την οποία μπορούμε να διασχίσουμε χρησιμοποιώντας τη σχέση μετάβασης. Η σχέση ικανοποιησιμότητας των τύπων της CTL ορίζεται επαγωγικά ως προς τη δομή των τύπων ως εξής:

[5] Το $\mathcal{U}$ είναι από την αγγλική λέξη Until που σημαίνει "μέχρι".

- $s \models p$ ανν $p \in \mu(s)$
- $s \not\models$ **false**
- $s \models \phi_1 \rightarrow \phi_2$ ανν $s \not\models \phi_1$ ή $s \models \phi_2$
- $s \models \exists \bigcirc \phi$ ανν $t \models \phi$ όπου $t \in S$ τέτοιο ώστε $(s,t) \in E$
- $s \models \exists(\phi_1 \mathcal{U} \phi_2)$ ανν για κάποιο μονοπάτι $(s_0 = s, s_1, \ldots)$ για κάποιο $i \geq 0$, $s_i \models \phi_2$ και $s_j \models \phi_1$ για όλα τα $0 \leq j < i$
- $s \models \forall(\phi_1 \mathcal{U} \phi_2)$ ανν για όλα τα μονοπάτια $(s_0 = s, s_1, \ldots)$ για κάποιο $i \geq 0$, $s_i \models \phi_2$ και $s_j \models \phi_1$ για όλα τα $0 \leq j < i$

Οι χρονικές λογικές είναι αρκετά εκφραστικές αλλά, όπως αναφέραμε και παραπάνω, αδυνατούν να εκφράσουν "ποσοτικές" ιδιότητες του χρόνου. Για παράδειγμα, σε μία χρονική λογική γραμμικού χρόνου μπορούμε να εκφράσουμε την ιδιότητα "κάθε φορά που συμβαίνει το p ακολουθείται από το q" (μπορούμε εύκολα να φανταστούμε ότι το p είναι, για παράδειγμα, μία αίτηση του χρήστη και το q είναι η κατάλληλη απάντηση από το σύστημα). Η ιδιότητα θα εκφραστεί ως

$$\Box(p \rightarrow \Diamond q)$$

Είναι όμως προφανές ότι δεν μπορούμε με αυτό το "λεξιλόγιο" να εκφράσουμε την παραπάνω ιδιότητα με κάποιους χρονικούς περιορισμούς. Δηλαδή την ιδιότητα "κάθε φορά που συμβαίνει το p ακολουθείται από το q μέσα σε δύο μονάδες χρόνου". Ως απάντηση σε αυτό το αίτημα, έχουν βρεθεί διάφοροι φορμαλισμοί οι οποίοι εμπλουτίζουν τις χρονικές λογικές ώστε να μπορούν να εκφράσουν ποσοτικές ιδιότητες του χρόνου, μετατρέποντάς τις σε *λογικές πραγματικού χρόνου (real time logics)*. Θα δούμε εδώ τις πιο ενδιαφέρουσες από αυτές τις προσπάθειες όπως αναφέρονται στο [AH91].

2.2.1 Λογικές με φραγμένους χρονικούς τελεστές

Στις λογικές με φραγμένους χρονικούς τελεστές (logics with bounded temporal operators), οι χρονικοί τελεστές φράσσονται από ένα διάστημα I με ρητά άκρα που συμβολίζει ένα χρονικό διάστημα. Συμβολικά, το διάστημα αυτό γράφεται ως δείκτης στους τελεστές. Έτσι, ο φραγμένος τελεστής $\Diamond_{[1,2]}$ ερμηνεύεται ως "τελικά μέσα σε 1 έως 2 μονάδες χρόνου". Η ιδιότητα που αναφέραμε παραπάνω μπορεί τώρα να εκφραστεί τυπικά ως εξής

$$\Box(p \to \Diamond_{[0,2]} q)$$

Μία από τις ενδιαφέρουσες λογικές πραγματικού χρόνου με φραγμένους τελεστές είναι η *Timed Computation Tree Logic (TCTL)*, η οποία είναι η CTL εμπλουτισμένη με τους φραγμένους τελεστές. (Ακολουθούμε την περιγραφή που δίνεται στο [ACD90].)

Συντακτικό της TCTL. Έστω ένα σύνολο ατομικών προτασιακών τύπων P τέτοιο ώστε $p \in P$. Έστω επίσης ότι $c \in N$, όπου N είναι το σύνολο των φυσικών αριθμών. Ενώ το σύμβολο $\sim$ εκφράζει μία από τις δυαδικές σχέσεις $<, \leq, =, \geq, >$. Οι τύποι της TCTL ορίζονται αναδρομικά ως εξής:

$$\phi := \ p \mid \textbf{false} \mid \phi_1 \to \phi_2 \mid \exists(\phi_1 \mathcal{U}_{\sim c} \phi_2) \mid \forall(\phi_1 \mathcal{U}_{\sim c} \phi_2)$$

Διαισθητικά, ο τύπος $\exists(\phi_1 \mathcal{U}_{<c} \phi_2)$ σημαίνει ότι για κάποιο μονοπάτι υπολογισμού, υπάρχει ένα αρχικό κομμάτι του με χρονική διάρκεια λιγότερο από c μονάδες χρόνου τέτοιο ώστε ο ϕ_2 είναι αληθής στην τελευταία κατάσταση του κομματιού και ο ϕ_1 είναι αληθής σε όλες τις ενδιάμεσες καταστάσεις.

Ορίζουμε κατ' αναλογία με την CTL, φραγμένους τελεστές $\Box$ και $\Diamond$ ως εξής:

$$\exists \Diamond_{\sim c} \phi = \exists(true\, \mathcal{U}_{\sim c} \phi), \quad \exists \Box_{\sim c} \phi = \neg \forall \Diamond_{\sim c} \neg \phi$$

Να σημειώσουμε εδώ ότι η TCTL δεν έχει τελεστή επόμενης στιγμής, γιατί βέβαια όταν μιλάμε για πυκνό χρόνο δεν υπάρχει μοναδική επόμενη στιγμή.

Σημασιολογία της TCTL. Για να ορίσουμε τη σημασιολογία της TCTL χρειαζόμαστε μερικές εισαγωγικές έννοιες που θα μας βοηθήσουν να "συλλάβουμε" την ιδέα του μοντέλου πυκνού χρόνου και του μονοπατιού σε αυτόν. Να σημειώσουμε εδώ ότι ο αντίστοιχος ορισμός που χρησιμοποιήσαμε για την CTL δεν αρκεί για τον απλούστατο λόγο ότι δεν μπορούμε να ορίσουμε ακολουθία διαδοχικών καταστάσεων.

Ορισμός 2.2.1 *Ένα s-μονοπάτι για κάποια κατάσταση $s \in S$ είναι μία απεικόνιση ρ από το R_+ στο S με $\rho(0) = s$.*

Θα ορίσουμε τώρα μία δομή η οποία θα είναι ανάλογη της δομή Kripke που χρησιμοποιήσαμε για τη σημασιολογία της CTL, χρησιμοποιώντας τον παραπάνω ορισμό του s-μονοπατιού.

Ορισμός 2.2.2 *Μία TCTL-δομή $\mathcal{M} = \langle S, \mu, f \rangle$ με*

- *S είναι ένα σύνολο καταστάσεων*
- *$\mu : S \to 2^P$ μία αποτίμηση τιμών αλήθειας στις ατομικές προτάσεις κάθε κατάστασης*
- *f απεικονίζει σε κάθε κατάσταση $s \in S$ ένα σύνολο s-μονοπατιών μέσω του S. Η f ικανοποιεί τις ακόλουθες ιδιότητες:*

  (α) $\forall s \in S, \forall \rho \in f(s), \forall t \in R_+, \rho^t \in f[\rho(t)]$.

  (β) $\forall s \in S, \forall \rho \in f(s), \forall t \in R_+, \rho_t \cdot f[\rho(t)] \subseteq f(s)$.[6]

Η ιδιότητα (α) εκφράζει κατά κάποιο τρόπο την κλειστότητα ως προς το πρόθεμα ρ^t κάθε μονοπατιού ρ της f και η (β) το γεγονός ότι τα μονοπάτια εξαρτώνται κάθε φορά μόνο από την τρέχουσα κατάσταση και όχι από τις παρελθούσες καταστάσεις.

Η σχέση ικανοποιησιμότητας των τύπων της TCTL ως προς μία TCTL-δομή $\mathcal{M} = \langle S, \mu, f \rangle$ ορίζεται επαγωγικά ως προς τη δομή των τύπων ως εξής:

- $s \models p$ ανν $p \in \mu(s)$
- $s \not\models \mathbf{false}$
- $s \models \phi_1 \to \phi_2$ ανν $s \not\models \phi_1$ ή $s \models \phi_2$
- $s \models \exists(\phi_1 \mathcal{U}_{\sim c} \phi_2)$ ανν για κάποιο μονοπάτι $\rho \in f(s)$, για κάποιο $t \sim c$, $\rho(t) \models \phi_2$ και $\rho(t') \models \phi_1$ για όλα τα $0 \leq t' < t$
- $s \models \forall(\phi_1 \mathcal{U}_{\sim c} \phi_2)$ ανν για κάθε μονοπάτι $\rho \in f(s)$, για κάποιο $t \sim c$, $\rho(t) \models \phi_2$ και $\rho(t') \models \phi_1$ για όλα τα $0 \leq t' < t$

[6](α) Το ρ_t απεικονίζει το $[0, t)$ στο S, (β) Το ρ^t ορίζεται ως: $\rho^t(0) = \rho(t)$ και $\rho^t(t') = \rho(t + t')$, (γ) Αν $\rho' : [0, t) \to S$, τότε $(\rho' \cdot \rho)(t') = \rho'(t')$ αν $t' < t$, $\rho(t' - t)$ αλλιώς

2.2.2 Λογικές με ποσοδείκτες δέσμευσης

Αν και δε φαίνεται με την πρώτη ματιά, η εκφραστικότητα των παραπάνω λογικών με φραγμένους τελεστές εξαντλείται σε περιγραφές περιορισμών που αφορούν διαδοχικά γεγονότα. Δηλαδή, μπορούμε να εκφράσουμε χρονικούς περιορισμούς για την εμφάνιση του q ως προς την εμφάνιση του p αλλά αν είχαμε και ένα τρίτο γεγονός r το οποίο θέλουμε να συμβεί μέσα σε τρεις χρονικές μονάδες από το p, τότε τα εκφραστικά μας μέσα θα αδυνατούσαν να μας βοηθήσουν.

Στις λογικές λοιπόν με ποσοδείκτες δέσμευσης (logics with freeze quantification), εισάγουμε την έννοια των μεταβλητών χρόνου οι οποίες συνδέονται άμεσα με (ουσιαστικά φράσσονται από) τη χρονική στιγμή που συμβαίνει κάτι, δηλαδή ουσιαστικά αναφέρονται στο χρόνο μίας κατάστασης κατά την οποία συμβαίνει το γεγονός. Οι μεταβλητές αυτές λειτουργούν ως ποσοδείκτες με το συμβολισμό: $x.$ όπου x είναι μία μεταβλητή χρόνου.

Για παράδειγμα, στον τύπο $\Diamond y.\phi$, η μεταβλητή χρόνου y ουσιαστικά δεσμεύεται με το χρόνο της κατάστασης στην οποία "τελικά" θα γίνει αληθές το ϕ. Η ιδιότητα που αναφέραμε παραπάνω θα εκφραστεί τώρα ως εξής:

$$\Box x.(p \rightarrow \Diamond y.(q \wedge y \leq x+2)$$

Ενώ η ιδιότητα που μας δημιούργησε το πρόβλημα και αναφέρεται σε τρία συμβάντα θα εκφραστεί ως εξής:

$$\Box x.(p \rightarrow \Diamond(q \wedge \Diamond z.(r \wedge z \leq x+3))$$

Τέλος, να σημειώσουμε εδώ ότι η ύπαρξη αυτών των ποσοδεικτών δεν προσδίδει στη λογική μας τα χαρακτηριστικά μίας πρωτοβάθμιας τροπικής λογικής. Οι ποσοδείκτες μπορούν να κάνουν αναφορά μόνο σε χρόνο που αφορά τις καταστάσεις.

2.2.3 Λογικές με μεταβλητή ρολογιού

Οι λογικές με μεταβλητή ρολογιού (logics with explicit clock variable) είναι άλλη μια προσπάθεια να εκφραστούν περιορισμοί σε χρόνο που δεν είναι "τοπικοί" , δηλαδή δεν αναφέρονται σε γειτονικά συμβάντα. Βασίζονται στην πρωτοβάθμια χρονική λογική με ποσοδείκτες σε γενικές μεταβλητές χρόνου. Σε αυτές τις λογικές έχουμε μία δυναμική μεταβλητή καταστάσεων T - που ονομάζεται μεταβλητή ρολογιού. Η μεταβλητή αυτή T θεωρούμε ότι έχει σε κάθε κατάσταση την τιμή του αντίστοιχου χρόνου.

Η αρχική ιδιότητα στην οποία έχουμε αναφερθεί θα γραφόταν με μια τέτοια λογική ως:

$$\forall x.\Box((p \land T = x) \to \Diamond(q \land T \leq x + 2))$$

Κλείνοντας, θα αναφέρουμε μερικές χαρακτηριστικές λογικές πραγματικού χρόνου. Εκτός από τη λογική TCTL, μία άλλη λογική διακλαδιζόμενου χρόνου είναι η *Real-time computation tree logic (RTCTL)*. Η λογική αυτή χρησιμοποιεί επίσης φραγμένους χρονικούς τελεστές, είναι προτασιακή και σε αντίθεση με την TCTL, η σημασιολογία της είναι διακριτού χρόνου.

Ας δούμε και μερικά παραδείγματα γραμμικών λογικών πραγματικού χρόνου. Η *Metric temporal logic (MTL)* είναι προτασιακή λογική η οποία χρησιμοποιεί φραγμένους χρονικούς τελεστές και διακριτό χρόνο. Η *Timed temporal logic (TPTL)* είναι επίσης προτασιακή λογική και χρησιμοποιεί ποσοδείκτες δέσμευσης. Σε αντίθεση με αυτές τις δύο, η προτασιακή *Metric interval temporal logic (MITL)* χρησιμοποιεί φραγμένους τελεστές χωρίς να επιτρέπει τα διαστήματα μηδενικού μήκους αλλά η σημασιολογία της είναι πραγματικού χρόνου.

Έχουν προταθεί και άλλες λογικές πραγματικού χρόνου συνδυάζοντας διάφορα χαρακτηριστικά. Αναπτύσσεται γι' αυτές μία θεωρία, ανεξάρτητη από το θέμα της τυπικής επαλήθευσης, η οποία εξετάζει θέματα εκφραστικότητάς τους, ισοδυναμίας της μίας με την άλλη και θέματα πολυπλοκότητας της ικανοποιησιμότητάς τους.

Κεφάλαιο 3

Τυπική επαλήθευση

Το πρόβλημα της τυπικής επαλήθευσης στα συστήματα πραγματικού χρόνου μπορεί να εκφραστεί σε μια αφηρημένη μορφή ως εξής:

> Δοσμένων δύο (τυπικών) περιγραφών συμπεριφορών πραγματικού χρόνου S και P, να βρεθεί κατά πόσο η S ταιριάζει με την P.

Συνήθως, η S είναι η τυπική περιγραφή ενός συστήματος πραγματικού χρόνου και η P είναι μία ιδιότητα που αφορά τις απαιτήσεις σε χρόνο του συστήματος. Το ρήμα "ταιριάζει" είναι αρκετά ουδέτερο αλλά (ή επομένως) δεν είναι αρκετά περιγραφικό. Στη θέση του θα μπορούσαμε να πούμε: "ικανοποιεί" ή "υλοποιεί".

Το μυστικό στην παραπάνω περιγραφή κρύβεται στη λέξη "τυπική" - εδώ είναι που θα χρησιμοποιήσουμε τα εργαλεία που παρουσιάσαμε στο κεφάλαιο δύο. Τα συγχρονισμένα αυτόματα μπορούν να χρησιμοποιηθούν τόσο για να εκφράσουν το σύστημα όσο και την ιδιότητα, και το ίδιο μπορεί να συμβεί και με τις λογικές πραγματικού χρόνου. Μπορούμε επίσης να έχουμε συνδυασμό των δύο εργαλείων: το σύστημα να εκφράζεται με ένα αυτόματο και η ιδιότητα μέσω μίας λογικής πραγματικού χρόνου. Ειδικά για την περίπτωση της ιδιότητας, όταν αυτή δεν χρειάζεται να εκφράσει κάποιο χρονικό περιορισμό, χρησιμοποιούνται πολύ συχνά και χρονικές λογικές όπως η CTL κ. α. Σε κάθε περίπτωση, το πρόβλημα της τυπικής επαλήθευσης παίρνει διαφορετική μορφή και απαιτεί διαφορετική λύση.

Οι τεχνικές και οι τρόποι χρήσης των θεωρητικών αυτών εργαλείων είναι αυτά που χρησιμοποιούνται γενικά για την τυπική επαλήθευση. Θα είμαστε πιστοί στο διαχωρισμό που κάναμε στο κεφάλαιο ένα μεταξύ των δύο διαφορετικών προσεγγίσεων της τυπικής επαλήθευσης.

Σχετικά με όσα αναφέρονται εδώ, η κυριώτερη αναφορά είναι το [AH91] αλλά υπάρχουν πολλά στοιχεία και από άλλες πηγές της βιβλιογραφίας του παρόντος.

3.1 Απόδειξη θεωρήματος

Έστω λοιπόν ότι έχουμε μία λογική $\mathcal{L}$ μέσω της οποίας έχουμε δύο τύπους ϕ_S και ϕ_P οι οποίοι εκφράζουν το σύστημα πραγματικού χρόνου και την ιδιότητα αντίστοιχα. Τότε το παραπάνω πρόβλημα επαλήθευσης παίρνει μία πιο συγκεκριμένη μορφή:

> Δοσμένων των τύπων ϕ_S και ϕ_P να βρεθεί κατά πόσο η συνεπαγωγή $\phi_S \rightarrow \phi_P$ είναι έγκυρη, ή ισοδύναμα,
>
> Δοσμένων των τύπων ϕ_S και ϕ_P, να βρεθεί κατά πόσο η σύζευξη $\phi_S \wedge \neg\phi_P$ είναι μη ικανοποιήσιμη.

Η τυποποιημένη αυτή μορφή του προβλήματος υποδηλώνει και τη λύση του: το πρόβλημά μας δεν είναι πλέον παρά ένα πρόβλημα ικανοποιησιμότητας. Αυτό βέβαια δε σημαίνει σε καμία περίπτωση ότι είναι απλό και εύκολο. Αντίθετα, ανάγεται σε ένα γνωστό μεν μαθηματικό πρόβλημα αλλά με μακρά και πολύπαθη ιστορία.

Αν θέλουμε να συνοψίσουμε τα αποτελέσματα ως προς αυτό το πρόβλημα για τις χρονικές λογικές θα λέγαμε ότι είναι μη ικανοποιητικά. Κατ' αρχάς, ξέρουμε ότι μόνο οι προτασιακές χρονικές λογικές είναι αποφασίσιμες.

Σε ότι αφορά στις λογικές πραγματικού χρόνου, το αν είναι αποφασίσιμες ή όχι εξαρτάται από δύο παράγοντες: την σημασιολογική ερμηνεία του χρόνου (διακριτός ή πυκνός) και τις πράξεις που επιτρέπουμε για την περιγραφή των χρονικών περιορισμών. Προφανώς, η ερμηνεία του χρόνου ως πυκνού και ο μεγάλος αριθμός επιτρεπόμενων πράξεων οδηγεί σε λογικές των οποίων η ικανοποιησιμότητα δεν είναι αποφασίσιμη.

Αλλά ακόμα για το μικρότερο τέτοιο σύνολο πράξεων (που είναι η χρήση του κατηγορήματος $\leq$ και η πρόσθεση με ακέραιες σταθερές $+c$) σε συνδυασμό με πυκνό πεδίο χρόνου οδηγούμαστε σε μη αποφασίσιμη ικανοποιησιμότητα. Αντίθετα, για τις αντίστοιχες λογικές πραγματικού χρόνου με διακριτό πεδίο χρόνου είναι αποφασίσιμο. Για παράδειγμα, έχει αποδειχτεί ότι το πρόβλημα της ικανοποιησιμότητας για τη λογική TPTL είναι $EXPSPACE$-πλήρες.

3.2 Έλεγχος μοντέλου

Δύο είναι οι κυριότερες μέθοδοι στην τεχνική ελέγχου μοντέλων την οποία περιγράψαμε γενικά στην ενότητα 1.1. Η μία μέθοδος δέχεται ως εισόδους τόσο το σύστημα όσο και την ιδιότητα με τη μορφή αυτομάτων και ελέγχει τη σχέση που έχουν οι γλώσσες που γίνονται αποδεκτές από αυτά (για παράδειγμα, αν η τομή των γλωσσών είναι κενή ή όχι). Στην άλλη μέθοδο, η επιθυμητή ιδιότητα εκφράζεται ως τύπος μιας χρονικής λογικής ενώ το σύστημα ως αυτόματο (ή πιο σωστά, ως το μοντέλο του αυτομάτου που είναι ένας γράφος μεταβάσεων με πεπερασμένες διακριτές καταστάσεις). Σε αυτή τη μέθοδο, ελέγχεται αν το σύστημα αυτό αποτελεί μοντέλο του τύπου της χρονικής λογικής.

Όποια μέθοδο όμως κι αν διαλέξουμε, η τεχνική του ελέγχου μοντέλων βασίζεται στο γεγονός ότι κατασκευάζουμε ή/και χρησιμοποιούμε ένα πεπερασμένο μοντέλο για το σύστημά μας το οποίο επομένως μπορούμε να το διασχίσουμε αποτελεσματικά. Στην περίπτωση όμως των συγχρονισμένων αυτομάτων, ο γράφος καταστάσεων τους είναι άπειρος. Θεωρούμε ότι τα ρολόγια μας παίρνουν πραγματικές τιμές κι επομένως, η κάθε διακριτή κατάσταση μπορεί να "συμμετέχει" σε άπειρες καταστάσεις του αυτομάτου.

Το θέμα του απείρου μεγέθους του γράφου των καταστάσεων αντιμετωπίζεται με ένα αλγόριθμο ελέγχου μοντέλων ο οποίος κατασκευάζει τον *γράφο περιοχής (region graph)*. Ο γράφος αυτός είναι ένα πεπερασμένο πηλίκο του άπειρου γράφου καταστάσεων. Κόμβοι του γράφου αυτού είναι ζεύγη με πρώτο μέλος τους μία διακριτή κατάσταση του αρχικού γράφου και δεύτερο μέλος του μία "περιοχή ρολογιού".

Η ιδέα είναι ότι μπορούμε να θεωρήσουμε ως μία κλάση ισοδυναμίας όλες τις αποτιμήσεις ρολογιών οι οποίες, δεδομένης μίας διακριτής κατάστασης, θα έχουν το ίδιο αποτέλεσμα ως προς τον υπολογισμό του αυτομάτου - μας "αναγκάζουν" δηλαδή να ακολουθήσουμε την ίδια μετάβαση. Η κλάση αυτή χαρακτηρίζεται από ένα πεπερασμένο σύνολο περιορισμών των ρολογιών τους οποίους ικανοποιεί και οι οποίοι ορίζουν μία "περιοχή ρολογιού" - είναι ουσιαστικά οι δυνατές τιμές που μπορούν να πάρουν τα ρολόγια. (Σχετικά με αυτή την κατασκευή είναι τα [ACD90], [AD94].)

Να τονίσουμε εδώ ότι η κατασκευή αυτή κάτα κάποιο τρόπο κατασκευάζει μία "διάσχιση" του αρχικού αυτομάτου διότι εκφράζει τους δυνατούς υπολογισμούς που επιτρέπει το αρχικό αυτόματο.

Ο γράφος αυτός περιοχής δεν είναι μεν "άπειρος" αλλά είναι "πολύ μεγάλος". Πιο συγκεκριμένα, το μέγεθός του είναι εκθετικό ως προς τον αριθμό των διακριτών μεταβλητών του συστήματος, το γνωστό πρόβλημα έκρηξης καταστάσεων. Στην περίπτωση όμως των συγχρονισμένων αυτομάτων, λόγω της περιγραφή τους χρόνου, έχουμε και άλλο ένα παράγοντα που επηρεάζει το μέγεθος του. Το μέγεθος του γράφου είναι επίσης εκθετικό ως προς τον αριθμό των ρολογιών και το μήκος των σταθερών στους χρονικούς περιορισμούς. Μάλιστα οι παράγοντες αυτοί δεν βελτιώνονται αν περάσουμε σε μοντέλα διακριτού χρόνου.

Τέλος, να αναφέρουμε ότι στις τεχνικές ελέγχου το σύστημα πραγματικού χρόνου S αποτελείται από διαφορετικές διαδικασίες, πεπερασμένες σε αριθμό, η καθεμία από αυτές περιγράφεται τυπικά μέσω ενός συγχρονισμένου αυτομάτου.[1] Το σύστημα πραγματικού χρόνου S περιγράφεται τότε μέσω του αυτομάτου-γινομένου των αυτομάτων που περιγράφουν τις διαδικασίες. Δηλαδή, μέσω ενός συγχρονισμένου αυτομάτου $\mathcal{A}_S = \mathcal{A}_S^1 \times \cdots \times \mathcal{A}_S^m$ αν το σύστημα S αποτελείται από m διαδικασίες και οι διαδικασίες αυτές περιγράφονται αντίστοιχα από τα συγχρονισμένα αυτόματα $\mathcal{A}_S^1, \cdots, \mathcal{A}_S^m$.

[1] Θα μπορούσαμε να θεωρήσουμε ότι το σύστημα περιγράφεται από ένα αυτόματο αλλά αυτό θα ήταν αρκετά μακριά από την πραγματικότητα.

3.2.1 Χρήση συγχρονισμένων αυτόματων

Έστω λοιπόν ότι το σύστημα περιγράφεται S μέσω του συγχρονισμένου αυτομάτου $\mathcal{A}_S$, όπως αναφέρεται παραπάνω, και η ιδιότητα P μέσω ενός συγχρονισμένου αυτόματου $\mathcal{A}_P$. Από αυτό το αυτόματο, ορίζεται το συμπληρωματικό του αυτόματο $\neg\mathcal{A}_P$ το οποίο ουσιαστικά περιγράφει τυπικά όλες τις ακολουθίες μεταβάσεων κατά τις οποίες δεν ισχύει η ιδιότητα P.

Το πρόβλημα επαλήθευσης παίρνει τώρα την εξής μορφή:

> Δοσμένων δύο συγχρονισμένων αυτομάτων $\mathcal{A}_S$ και $\mathcal{A}_P$, να βρεθεί κατά πόσο το αυτόματο $\mathcal{A}_S \times \neg\mathcal{A}_P$ ορίζει την κενή γλώσσα.

Η τυποποιημένη μορφή μας βοηθάει για μια ακόμα φορά και χρησιμοποιώντας τα εργαλεία της θεωρίας συγχρονισμένων αυτομάτων, η λύση στο πρόβλημα της τυπικής επαλήθευσης είναι:

1. Κατασκευή του γινομένου των συγχρονισμένων αυτομάτων που αποτελούν το σύστημα - δηλαδή του $\mathcal{A}_S$.

2. Κατασκευή του συμπληρωματικού αυτομάτου του $\mathcal{A}_P$.

3. Κατασκευή του γινομένου των δύο παραπάνω.

4. Έλεγχος για το αν το αυτόματο που προκύπτει ορίζει τη κενή γλώσσα.

Δυστυχώς, τα πράγματα και εδώ δεν είναι ακριβώς τα καλύτερα δυνατά. Ας πάρουμε τις διαδικασίες με τη σειρά έτσι όπως εμφανίζονται στην παραπάνω λύση.

- Κατά τη κατασκευή του γινομένου συγχρονισμένων αυτομάτων, όπως και στην περίπτωση των κλασικών αυτομάτων, το μέγεθος του γινομένου είναι ανάλογο του γινομένου των μεγεθών των αρχικών αυτομάτων. Συνολικά, το γινόμενο αυτό είναι εκθετικό ως προς τις περιγραφές των επιμέρους διαδικασιών - το πρόβλημα της έκρηξης των καταστάσεων που αναφέραμε στην ενότητα 1.1.

- Όπως έχουμε ήδη δει στην ενότητα 2.1, τα συγχρονισμένα Muller αυτόματα δεν είναι κλειστά ως προς το συμπλήρωμα. Μπορούμε να αντιμετωπίσουμε το πρόβλημα αυτό με δύο τρόπους:

  (α) Να περιοριστούμε σε ντετερμινιστικά αυτόματα για την περιγραφή των ιδιοτήτων των συστημάτων πραγματικού χρόνου, αυτόματα τα οποία είναι κλειστά ως προς το συμπλήρωμα. Να σημειώσουμε εδώ ότι στην περίπτωση που τόσο το σύστημα όσο και η ιδιότητα είναι εκφρασμένα με ντετερμινιστικά συγχρονισμένα αυτόματα, ολόκληρη η διαδικασία είναι εκθετική ως προς και τις δυο περιγραφές.

  (β) Να περιοριστούμε σε συγχρονισμένα Muller αυτόματα με διακριτό χρόνο τα οποία είναι κλειστά ως προς το συμπλήρωμα.

- Έχουμε ήδη δει ότι το πρόβλημα του ελέγχου για το αν ένα συγχρονισμένο αυτόματο ορίζει την κενή γλώσσα είναι $PSPACE$-πλήρες, αν το αυτόματο χρησιμοποιεί το μικρότερο δυνατό σύνολο πράξεων ως προς τους χρονικούς περιορισμούς. (Φυσικά εδώ εννοείται ότι χρησιμοποιούμε το γράφο περιοχής). Σε αυτή την περίπτωση, ο χρόνος της εκτέλεσης του ελέγχου εξαρτάται εκθετικά από το μήκος των περιορισμών - αν θεωρήσουμε ότι η αναπαράστασή τους είναι δυαδική.

3.2.2 Συνδυασμός χρήσης συγχρονισμένων αυτομάτων και λογικών

Στην περίπτωση αυτή, το σύστημα πραγματικού χρόνου περιγράφεται μέσω ενός γινομένου συγχρονισμένων αυτομάτων $\mathcal{A}_S$, όπως και παραπάνω. Η ιδιότητα P εκφράζεται μέσω μίας λογικής $\mathcal{L}$ ως ένας τύπος ϕ_P. Το πρόβλημα και εδώ είναι ουσιαστικά το

> Δοσμένου ενός συγχρονισμένου αυτομάτου $\mathcal{A}_S$ και ενός τύπου ϕ_P,
> να βρεθεί κατά πόσο ο τύπος αληθεύει στο μοντέλο του αυτομάτου.

Η προσέγγιση αυτή είναι από τις πιο δημοφιλείς - γενικότερα στις τυπικές μεθόδους αλλά και ειδικότερα για τα συστήματα πραγματικού χρόνου. Ας δούμε πως διαμορφώνεται το σχεδιάγραμμα της λύσης σε αυτή την περίπτωση.

- Δεδομένου του συγχρονισμένου αυτομάτου τους συστήματος, κατασκευάζεται ο γράφος περιοχής όπως αναφέρθηκε παραπάνω.

- Αυτός ο γράφος είναι μεν πεπερασμένος αλλά "κληρονομεί" το πρόβλημα της έκρηξης καταστάσεων. Έτσι, το μέγεθός του αυξάνεται εκθετικά ως προς τον αριθμό των μεταβλητών του συστήματος αλλά επίσης και ως προς την μεγαλύτερη σταθερά του χρόνου και τον αριθμό των ρολογιών.

  Αυτό το πρόβλημα αντιμετωπίζεται με διάφορους αλγορίθμους οι οποίοι συχνά περιγράφονται ως *συμβολικός έλεγχος μοντέλων (symbolic model checking)*. Οι αλγόριθμοι αυτοί στοχεύουν στην ελαχιστοποίηση του αριθμού των καταστάσεων μέσω συμβολικής αναπαράστασης των καταστάσεων σε βελτιωμένες κατασκευές των γράφων περιοχής μέσω αμφιπροσομοιώσεων οι οποίες διατηρούν τις ιδιότητες που μας ενδιαφέρουν. Επίσης, άλλες τεχνικές επιτελούν ταυτόχρονα την επαλήθευση με την κατασκευή του γράφου καταστάσεων (on-the-fly verification) με προφανές όφελος την αποφυγή της κατασκευής ολόκληρου του γράφου. (Αναφορές για τέτοιους αλγορίθμους αποτελούν τα [Tr98], [HNSY94], [ACD90].)

- Το τελευταίο κομμάτι της διαδικασίας ελέγχου μοντέλων είναι βέβαια η διάσχιση του μοντέλου με κατάλληλο αλγόριθμο για την ύπαρξη ή απουσία καταστάσεων όπου ισχύει η ιδιότητα που μας ενδιαφέρει. Οι αλγόριθμοι αυτοί ποικίλλουν ανάλογα με τη λογική που χρησιμοποιείται και τον τρόπο κατασκευής του μοντέλου. (Οι παραπάνω αναφορές περιλαμβάνουν και αντίστοιχους αλγόριθμους.)

Σημαντικό είναι να τονίσουμε εδώ ότι τα παραπάνω βήματα δεν ανταποκρίνονται απαραίτητα σε αντίστοιχα βήματα και αλγορίθμους στις εφαρμογές του ελέγχου μοντέλων. Είδαμε για παράδειγμα ότι η κατασκευή του γράφου περιοχής μας αποτελεί ταυτόχρονα και ένα βήμα για τον αλγόριθμο της διάσχισης του γράφου. Οι αλγόριθμοι συνήθως αντιμετωπίζουν το πρόβλημα ολικά και η διάκριση εδώ αναφέρεται κυρίως για μεθοδολογικούς λόγους.

Ως προς την πολυπλοκότητα έχει δειχτεί ότι το πρόβλημα του ελέγχου μοντέλων για συστήματα πραγματικού χρόνου είναι $PSPACE$- δύσκολο. Πιο συγκεκριμένα, έχει δειχτεί ότι το πρόβλημα του ελέγχου μοντέλου για την TCTL με πυκνό χρόνο και με το μικρότερο σύνολο πράξεων για τους περιορισμούς σε χρόνου είναι $PSPACE$-πλήρες [ACD90].

3.3 Σχετικά με τις υλοποιήσεις

Όπως έχουμε ήδη αναφέρει, απώτερος στόχος των τυπικών μεθόδων είναι βέβαια η αυτοματοποίηση των λειτουργιών που περιλαμβάνουν. Η ύπαρξη αλγορίθμων για την τυπική επαλήθευση συστημάτων είναι αυτό που επιζητείται και βέβαια το επόμενο βήμα είναι η υλοποίηση αυτών των αλγορίθμων σε υπολογιστικά συστήματα. Ήδη, για περιπτώσεις τυπικής επαλήθευσης όχι απαραίτητα συστημάτων πραγματικού χρόνου έχουν γίνει πολλές προσπάθειες, τόσο σε ερευνητικό επίπεδο όσο και σε επίπεδο εφαρμογών.

Τα αποτελέσματα που είχαμε παραπάνω από τη θεωρία των μεθόδων τυπικής επαλήθευσης για συστήματα πραγματικού χρόνου μάλλον δεν είναι ενθαρρυντικά. Σε πολλές περιπτώσεις, το πρόβλημα μας δεν είναι υπολογίσιμο και σε αυτές που είναι, πολύ συχνά έχει απαγορευτική πολυπλοκότητα.

Παρόλα αυτά, έχουν γίνει προσπάθειες με τη χρήση διάφορων τεχνικών οι οποίες φαίνεται να αποδίδουν αρκετά καλά σε μικρής έκτασης παραδείγματα. Ενδεικτικά, να αναφέρουμε το εργαλείο KRONOS το οποίο είναι έχει αναπτυχθεί από το ερευνητικό ινστιτούτο VERIMAG στη Γαλλία από το 1992 και το οποίο χρησιμοποιεί τεχνικές ελέγχου μοντέλων. Δέχεται την περιγραφή του συστήματος ως ένα συγχρονισμένο αυτόματο και την ιδιότητα ως τύπο της λογικής TCTL.[2]

Η αλήθεια είναι ότι ακόμα και αν όλα τα θεωρητικά αποτελέσματα ήταν ιδιαίτερα θετικά, ο δρόμος προς την υλοποίηση τους είναι αρκετά μακρύς. Μερικά από τα ζητήματα στα οποία αξίζει να αναφερθούμε είναι τα εξής:

- Συνήθως οι υλοποιήσεις είναι ειδικές για κάθε περίπτωση ή πρόβλημα ή ακόμα και για κάθε λύση που προτείνεται. Το γεγονός αυτό απομακρύνει την προοπτική ομοιογενών συστημάτων για όλες τις εφαρμογές.

[2] Για περισσότερες πληροφορίες σχετικά με το KRONOS, δείτε το: http://www-verimag.imag.fr/TEMPORISE/kronos.

- Αποσιωπήσαμε το γεγονός ότι η περιγραφή ενός συστήματος πραγματικού χρόνου είναι συχνά αρκετά περίπλοκη. Δεν είναι καθόλου προφανές ότι με βάση αυτή την περιγραφή μπορούμε να το μεταγράψουμε, για παράδειγμα, στη "γλώσσα" των συγχρονισμένων αυτομάτων. Απαιτούνται τυπικές γλώσσες ως ενδιάμεσοι μεταφραστές οι οποίες είναι άλλο ένα θέμα που πρέπει να λυθεί πριν προχωρήσουμε στην υλοποίηση.

- Τέλος, σημαντικό είναι το θέμα της αναπαράστασης της πληροφορίας στην υλοποίησή που επιλέγουμε κάθε φορά. Το θέμα αυτό έχει σημαντική επίδραση στην πολυπλοκότητα των υπολογισμών μας κι επομένως, αξίζει την προσοχή μας.

Κεφάλαιο 4

Μία τεχνική χρονοπρογραμματισμού

Ένα σύστημα πραγματικού χρόνου συχνά αποτελείται από ξεχωριστές εργασίες ή/και διαδικασίες οι οποίες αλληλεπιδρούν η μία με την άλλη ή/και με το εξωτερικό περιβάλλον (π. χ. τους χρήστες του συστήματος), μοιράζονται κοινούς υπολογιστικούς πόρους κ. λ. π. Ένα σύστημα λοιπόν πραγματικού χρόνου περιγράφεται όχι μόνο από τις εργασίες/διαδικασίες που το απαρτίζουν αλλά και από τους περιορισμούς από τους οποίους διέπεται η λειτουργία τους.

Όλοι αυτοί οι περιορισμοί οι οποίοι ουσιαστικά καθορίζουν την εκτέλεση των εργασιών/διαδικασιών εκφράζονται με τους περιορισμούς στο χρόνο. Το "πρόγραμμα εκτέλεσης" επομένως σε ένα τέτοιο σύστημα δεν είναι τίποτα άλλο από ένας προγραμματισμός μέσα στα πλαίσια του χρόνου - *χρονοπρογραμματισμός (scheduling)* - εργασιών/διαδικασιών ο οποίος θα σέβεται όλους τους περιορισμούς που θέτει το ίδιο το σύστημα.

Το πρόβλημα του χρονοπρογραμματισμού είναι ένα πρόβλημα παλιό όσο και η επιστήμη υπολογιστών. Ουσιαστικά, είναι ένα ακόμα πρόβλημα επαλήθευσης με παραπάνω απαιτήσεις ως προς τη λύση του. Εδώ θα παρουσιάσουμε μία τεχνική αντιμετώπισής του η οποία βασίζεται σε τεχνικές τυπικής επαλήθευσης συστημάτων πραγματικού χρόνου και ως τέτοια, αποτελεί μία νέα "ματιά" στο πρόβλημα. Η τεχνική αυτή περιγράφεται αναλυτικά στο [AGP99].

4.1 Το πρόβλημα

Το πρόβλημα του χρονοπρογραμματισμού για ένα σύστημα πραγματικού χρόνου μπορεί να εκφραστεί ως εξής:

> Δεδομένου ενός συστήματος S και μίας ιδιότητας P, να βρεθεί ένα πρόγραμμα εκτέλεσης του συστήματος ως προς το χρόνο ώστε να ικανοποιείται η ιδιότητα P.

Χωρίς να αναφέρεται ρητά, το πρόγραμμα αυτό πρέπει να πληροί δύο προϋποθέσεις: πρώτον, να σέβεται τους χρονικούς περιορισμούς των εργασιών του συστήματος και δεύτερον, η λειτουργία του συστήματος ως απόρροια του προγράμματος να ικανοποιεί την ιδιότητα P.

Κατ' ουσία, στο πρόβλημα του χρονοπρογραμματισμού κάνει την εμφάνισή της η παράμετρος "περιβάλλον" - το οποίο καλούμαστε να συμπεριλάβουμε ως ένα είδος επιπλέον περιορισμών. Γι' αυτό και η τεχνική που θα περιγράψουμε εδώ επισημαίνει το διαχωρισμό μεταξύ *ελέγξιμων (controllable)* και *μη ελέγξιμων (uncontrollable)* περιορισμών.

Η διαφοροποίηση είναι προφανής: μη ελέγξιμοι είναι οι περιορισμοί οι οποίοι επιβάλλονται από το περιβάλλον (π.χ. η χρονική διάρκεια αποπεράτωσης μίας εργασίας και η εξάρτησή της από το φόρτο του μηχανήματος στο οποίο εκτελείται) και τους οποίους δε μπορούμε να αλλάξουμε ή να επέμβουμε ενώ ελέγξιμοι είναι οι περιορισμοί στους οποίους μπορούμε να επέμβουμε (π.χ. η χρονική στιγμή εκκίνησης μίας εργασίας). Παρακάτω θα δούμε πως η τεχνική που θα αναπτύξουμε εδώ χειρίζεται αυτή τη διαφοροποίηση.

Παράδειγμα. Για να γίνουμε πιο σαφείς, θα χρησιμοποιήσουμε ένα παράδειγμα το οποίο θα μας συνοδεύσει καθ' όλη τη διάρκεια της παρουσίασης της προσέγγισής μας. Ένα "απλό" σύστημα πραγματικού χρόνου είναι ένα έγγραφο πολυμέσων το οποίο περιλαμβάνει διάφορες αυτόνομες διαδικασίες/εργασίες όπως μουσική, βίντεο, ηχητικό μέρος, κείμενο, πρόγραμμα Java και εικόνα. Όταν καλείται το σύστημα αυτό από ένα χρήστη, προφανώς κινητοποιεί τις εργασίες του με τέτοιο τρόπο ώστε να δοθεί ένα ολοκληρωμένο αποτέλεσμα όπως αυτό έχει καθοριστεί από τον κατασκευαστή του.

Η καθεμία από τις εργασίες αυτές έχει μία συγκεκριμένη χρονική διάρκεια η οποία στην περίπτωσή μας χαρακτηρίζεται από ένα ελάχιστο και ένα μέγιστο αριθμό μονάδων χρόνου. Η χρονική διάρκεια αυτή καθορίζεται από τον κατασκευαστή για την κάθε εργασία χωριστά και χωρίς να έχει ληφθεί υπ' όψιν η ύπαρξη των άλλων. Τα χρονικά διαστήματα εκτέλεσης στο παράδειγμά μας φαίνονται στον πίνακα 4.1.

Εργασία	**Ελάχιστη διάρκεια**	**Μέγιστη διάρκεια**
Μουσική	30	40
Βίντεο	15	20
Ηχητικό μέρος	20	30
Κείμενο	5	10
Πρόγραμμα Java	20	30
Εικόνα	20	∞

Πίνακας 4.1 Χρονική διάρκεια εργασιών του εγγράφου πολυμέσων

Παράλληλα, οι εργασίες αυτές υπόκεινται και σε κάποιους περιορισμούς συγχρονισμού ώστε να δημιουργείται το επιθυμητό αποτέλεσμα - για παράδειγμα, το ηχητικό μέρος ενδεχομένως να αποτελεί τις συνομιλίες των ανθρώπων που εμφανίζονται στο βίντεο κι επομένως, βίντεο και ηχητικό μέρος θα πρέπει να εκτελούνται ταυτόχρονα. Οι περιορισμοί αυτοί είναι:

1. Κατά την έναρξη εκτέλεσης του εγγράφου, οι εργασίες μουσική, βίντεο, ηχητικό μέρος και πρόγραμμα Java ξεκινούν παράλληλα.

2. Οι εργασίες βίντεο και ηχητικό μέρος είναι στενά δεμένες μεταξύ τους - τερματίζουν και οι δύο μόλις τελειώσει οποιαδήποτε από τις δύο.

3. Η εργασία κείμενο ακολουθεί αμέσως μετά από το τερματισμό των εργασιών βίντεο και ηχητικό μέρος.

4. Οι εργασίες μουσική και κείμενο τελειώνουν την ίδια στιγμή.

5. Η εργασία πρόγραμμα Java ακολουθείται από την εργασία εικόνα.

6. Η εκτέλεση του εγγράφου τερματίζει μόλις τερματίσουν η εικόνα και οι εργασίες μουσική και κείμενο.

7. Τέλος, οι χρόνοι εκτέλεσης των εργασιών ηχητικό μέρος και πρόγραμμα Java εξαρτώνται από το φόρτο του μηχανήματος στο οποίο τα εκτελούμε κι επομένως, δεν είναι ελέγξιμοι.

Όλα τα παραπάνω αποτελούν ουσιαστικά την περιγραφή ενός συστήματος πραγματικού χρόνου - του εγγράφου πολυμέσων. Ας υποθέσουμε ότι η ιδιότητα την οποία θέλουμε να ικανοποιεί είναι: "το έγγραφο πολυμέσων εκτελείται επιτυχώς" - δηλαδή όταν το έγγραφο καλείται, εκτελεί όλες τις εργασίες του και τερματίζει. Το πρόβλημα, λοιπόν σε αυτή την περίπτωση είναι:

> Δεδομένου ενός εγγράφου πολυμέσων με τους παραπάνω περιορισμούς και της ιδιότητας "επιτυχής εκτέλεση", να βρεθεί ένα πρόγραμμα εκτέλεσης των εργασιών του ώστε να ικανοποιείται η ιδιότητα αυτή.

Θα δούμε στη συνέχεια πως θα προσεγγίσουμε αυτό το πρόβλημα μέσω των μεθόδων της τυπικής επαλήθευσης.

4.2 Τυπική περιγραφή του προβλήματος

Το επόμενο βήμα προς τη λύση του προβλήματος είναι η επιλογή ενός τυπικού εργαλείου για την περιγραφή του. Η χρησιμοποιήση της τεχνικής του ελέγχου μοντέλων ουσιαστικά μας υποδεικνύει ότι το εργαλείο αυτό θα είναι τα συγχρονισμένα αυτόματα.

Πριν φτάσουμε όμως σε αυτά θα εκμεταλλευτούμε την εκφραστικότητα των Petri Nets, από τα οποία στη συνέχεια θα μεταβούμε στα συγχρονισμένα αυτόματα. Πιο συγκεκριμένα, θα χρησιμοποιήσουμε τα *Petri Nets με προθεσμίες - PNΠ (Petri Nets with Deadlines - PND)* - τα οποία αποτελούν επέκταση των 1-safe Petri Nets στην περίπτωση που θέλουμε να περιγράψουμε ιδιότητες χρόνο. Πριν δώσουμε τον ορισμό τους, ας θυμηθούμε τον ορισμό των 1-safe Petri Nets.

Ορισμός 4.2.1 *Ένα 1-safe Petri Net $(\mathcal{P}, \mathcal{T}, A)$ αποτελείται από τα εξής τρία μέρη:*

1. *Ένα πεπερασμένο σύνολο θέσεων $\mathcal{P}$.*
2. *Ενα πεπερασμένο λεξιλόγιο πράξεων A.*
3. *Μία συνάρτηση μετάβασης $\mathcal{T} \subseteq 2^{\mathcal{P}} \times A \times 2^{\mathcal{P}}$.*

Τώρα μπορούμε να δώσουμε τον ορισμό ενός ΡΝΠ.

Ορισμός 4.2.2 *Ένα ΡΝΠ $(\mathcal{P}, \mathcal{T}, A, X, h)$ αποτελείται από τα εξής τρία μέρη:*

1. *Ένα 1-safe Petri net $(\mathcal{P}, \mathcal{T}, A)$*
2. *Ενα σύνολο $X = \{x_1, x_3, \ldots, x_m\}$ μεταβλητών που τις ονομάζουμε ρολόγια και παίρνουν τιμές στο R_+*
3. *Μία συνάρτηση h η οποία αντιστοιχεί τις μεταβάσεις του $\mathcal{T}$ που δεν έχουν στοιχεία χρόνου σε μεταβάσεις οι οποίες έχουν πληροφορία χρόνου, δηλαδή: $h(P, a, P') = (P, (a, g, r), P')$.*
   Το g είναι ένα κατηγόρημα-φρουρός το οποίο έχει το ακόλουθο συντακτικό $g ::= x \leq c | x < c | x - y \leq c | x - y < c | g \wedge g | \neg g$ όπου $x, y \in X$ και c είναι ακέραιος και $r \subseteq X$.

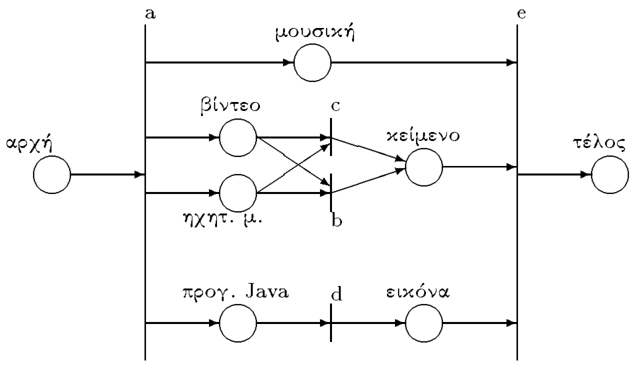
Το g περιγράφει με τυπικό τρόπο τους περιορισμούς σε χρόνο που καθορίζουν την πράξη a ενώ το r είναι ένα σύνολο ρολογιών τα οποία μηδενίζονται κατά την εκτέλεση της πράξης a.

Παράδειγμα. Ας δούμε πως αναπαριστάνεται το σύστημα του εγγράφου πολυμέσων με βάση τον παραπάνω φορμαλισμό. Πρώτα θα προσπαθήσουμε να περιγράψουμε το PN χωρίς να βάλουμε την πληροφορία του χρόνου.
Το σύνολο των θέσεων είναι: $\mathcal{P} = \{$ α, μ, β, ημ, κ, πJ, ε, τ $\}$[1].
Το λεξιλόγιο των πράξεων είναι: $A = \{a, b, c, d, e\}$.
Η συνάρτηση μετάβασης είναι: $\mathcal{T} = \{$ ({α},a, {μ, β, ηχ, πJ}), ({β, ηχ}, b,{κ}),

[1] Οι συμβολισμοί είναι προφανείς, δηλαδή: α: αρχή, μ: μουσική, β: βίντεο, ημ: ηχητικό μέρος, κ: κείμενο, πJ: πρόγραμμα Java, ε: εικόνα, τ: τέλος.

({β, ηχ,},c,{κ}), ({πJ}, d, {κ}), ({μ, κ, ε}, e, {τ}) }.

Η εκφραστικότητα αυτού του εργαλείου είναι προφανώς περιορισμένη. Για παράδειγμα, είμαστε σε θέση να εκφράσουμε τον περιορισμό 5 (με την ύπαρξη της πράξης d και του στοιχείου της συνάρτησης μετάβασης ({πρόγραμμα Java}, d, {εικόνα})) αλλά δεν είμαστε σε θέση να εκφράσουμε οποιοδήποτε περιορισμό στη χρονική διάρκεια των εργασιών. Δείτε το σχήμα 4.1.



Σχήμα 4.1 PN που αναπαριστά το έγγραφο πολυμέσων

Από αυτό το PN θα περάσουμε σε ένα PNΠ το οποίο θα μας επιτρέψει να εκφράσουμε όλες τις πληροφορίες σχετικά με τους περιορισμούς σε χρόνο για το σύστημά μας.

Το σύνολο των ρολογιών είναι: $X = \{x_\mu, x_\beta, x_{\eta\mu}, x_{\pi J}, x_\varkappa, x_\varepsilon\}$. Θα μπορούσαμε να χρησιμοποιήσουμε λιγότερα ρολόγια - συγκεκριμένα τρία ρολόγια μας αρκούν - αλλά για λόγους σαφήνειας χρησιμοποιούμε ένα ξεχωριστό ρολόι για κάθε εργασία.

Τα κατηγορήματα που περιγράφουν τυπικά τους περιορισμούς στη διάρκεια της κάθε εργασίας και σχετίζονται με τις μεταβάσεις στο PN είναι[2] :

$$\begin{array}{lll} g^\varepsilon_\mu = 30 \leq x_\mu \leq 40 & g^\varepsilon_\beta = 15 \leq x_\beta \leq 20 & g^\mu_{\eta\mu} = 20 \leq x_{\eta\mu} \leq 30 \\ g^\varepsilon_\varkappa = 5 \leq x_\varkappa \leq 10 & g^\mu_{\pi J} = 20 \leq x_{\pi J} \leq 30 & g^\varepsilon_\varepsilon = 20 \leq x_\varepsilon \end{array}$$

[2] Οι δείκτες "ε" και "μ" προσδιορίζουν ελέγξιμους και μη ελέγξιμους περιορισμούς αντίστοιχα.

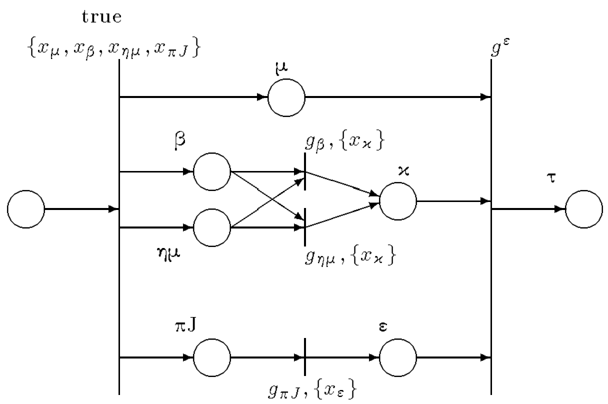
Η συνάρτηση μετάβασης h στην προκειμένη περίπτωση είναι:

$$(\{\alpha\}, a, \{\mu, \beta, \eta\mu, \pi J\}) \to (\{\alpha\}, (\alpha, true, \{x_\mu, x_\beta, x_{\eta\mu}, x_{\pi J}\}), \{\mu, \beta, \eta\mu, \pi J\}),$$
$$(\{\beta, \eta\mu\}, b, \{\varkappa\}) \to (\{\beta, \eta\mu\}, (b, g_{\eta\mu}, \{x_\beta, x_{\eta\mu}\}), \{\varkappa\}),$$
$$(\{\beta, \eta\mu, \}, c, \{\varkappa\}) \to (\{\beta, \eta\mu, \}, (c, g_\beta, \{x_\beta, x_{\eta\mu}\}), \{\varkappa\}),$$
$$(\{\pi J\}, d, \{\varepsilon\}) \to (\{\pi J\}, (d, g_{\pi J}, \{x_\varepsilon\}), \{\varepsilon\}),$$
$$(\{\mu, \varkappa, \varepsilon\}, e, \{\tau\}) \to (\{\mu, \varkappa, \varepsilon\}, (e, g^\varepsilon, \emptyset), \{\tau\})$$

Ο περιορισμός g^ε "εκφράζει" τους περιορισμούς 4 και 6 και υπολογίζεται μέσω πράξεων μεταξύ των αντίστοιχων περιορισμών των εργασιών. Η επιλογή των πράξεων βασίζεται στον τρόπο ερμηνείας των μεταβάσεων συγχρονισμού των ΡΝΠ. Στη συγκεκριμένη περίπτωση, έχουμε ότι:

$$\begin{aligned} g^\varepsilon &:= MAX(AND(g_\mu, g_\varkappa), g_\varepsilon) \\ &= (30 \leq x_\mu) \wedge (5 \leq x_\varkappa) \wedge (20 \leq x_\varepsilon) \wedge (20 \leq x_\mu - x_\varkappa \leq 35) \wedge \\ &\quad (x_\mu - x_\varepsilon \leq 40) \wedge (x_\varkappa - x_\varepsilon \leq 10) \end{aligned}$$

Το ΡΝΠ που περιγράφει το σύστημά μας φαίνεται στο σχήμα 4.2.



Σχήμα 4.2 ΡΝΠ που αναπαριστά το έγγραφο πολυμέσων

Το επόμενο βήμα στην τυπική περιγραφή του προβλήματός μας θα είναι η υιοθέτηση σημασιολογίας για το ΡΝΠ. Αυτό θα το κάνουμε με τη βοήθεια των συγχρονισμένων αυτομάτων με προθεσμίες. (Δείτε την αντίστοιχη παράγραφο στην ενότητα 2.2.)

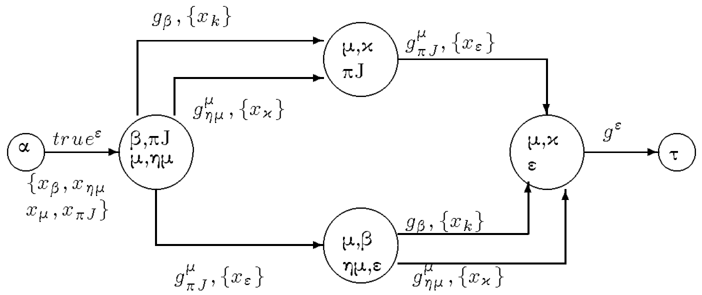
Ορισμός 4.2.3 *Δοσμένου ενός ΡΝΠ* $(\mathcal{P}, \mathcal{T}, A, X, h)$ *ορίζεται ένα ΣΑΠ* $\mathcal{A}$ $(S, \rightarrow, A, X, h')$ *ως εξής:*

1. *Το σύνολο των διακριτών καταστάσεων του* $\mathcal{A}$ *είναι:* $S \subseteq 2^{\mathcal{P}}$.

2. $(s, a, s') \in \rightarrow$ *αν και μόνο αν* $\exists P_1, P_2 \subseteq \mathcal{P}$ *τέτοια ώστε (a)* $(P_1, a, P_2) \in \mathcal{T}$, *(b)* $P_1 \subseteq s$ *και (c)* $s' = (s - P_1) \cup P_2$.

3. *Η* h' *αντιστοιχεί τις μεταβάσεις της* $\rightarrow$ *που δεν έχουν πληροφορία για το χρόνο σε μεταβάσεις οι οποίες έχουν πληροφορία για το χρόνο. Έτσι,* $h'(s, a, s') = h(P_1, a, P_2) = (s, (a, g, r), s')$, *όπου τα* P_1, P_2 *είναι αυτά στα οποία οφείλεται η μετάβαση* (s, a, s') *όπως ορίσαμε στο (2).*

Παράδειγμα.Ας δούμε πως θα εφαρμόσουμε τους παραπάνω ορισμούς για το συγκεκριμένο παράδειγμα του εγγράφου πολυμέσων. Στόχος μας είναι από το ΡΝΠ το οποίο φαίνεται στο σχήμα 4.2 και το οποίο περιγράφει με τυπικό τρόπο το έγγραφο μας να περάσουμε στο αντίστοιχο ΣΑΠ. Το συγχρονισμένο αυτόματο που παράγεται φαίνεται στο σχήμα 4.3.



Σχήμα 4.3 ΣΑΠ που αναπαριστά το έγγραφο πολυμέσων

Επομένως, το πρόβλημα του χρονοπρογραμματισμού τώρα μπορεί να περιγραφεί με τη βοήθεια των εργαλείων αυτών με τυπικό τρόπο. Το ίδιο το σύστημα πραγματικού χρόνου θα εκφραστεί μέσω ενός ΣΑΠ ενώ η ιδιότητα την οποία επιθυμούμε να ικανοποιεί θα είναι ένα κατηγόρημα. Το πρόβλημα που αναφέραμε παραπάνω γίνεται τώρα:

> Δοσμένου ενός ΣΑΠ $\mathcal{A}$ και μίας ιδιότητας P', να βρεθεί ένα ΣΑΠ $\mathcal{A}_{P'}$ το οποίο "εκφράζει" ένα χρονικό πρόγραμμα εκτέλεσης του συστήματος ώστε να ικανοποιείται η ιδιότητα P'.

Προφανώς, θα πρέπει να καθορίσουμε τι εννοούμε με την λέξη "εκφράζει". Στη προκειμένη περίπτωση, το ζητούμενο ΣΑΠ $\mathcal{A}_{P'}$ χαρακτηρίζει όλες τις καταστάσεις του $\mathcal{A}$, δηλαδή του συστήματος πραγματικού χρόνου, οι οποίες ικανοποιούν την ιδιότητα P'. Οι πιθανές ακολουθίες ελέγξιμων μεταβάσεων του αυτομάτου αυτού είναι αυτές και μόνο αυτές οι οποίες περνούν από καταστάσεις του συστήματος στις οποίες το σύστημα

4.3 Σύνθεση του εργαλείου χρονοπρογραμματισμού

Πριν δώσουμε την προτεινόμενη λύση στο πρόβλημα, θα ήταν σκόπιμο να αναλύσουμε την επιλογή της λέξης "σύνθεσης" η οποία εμφανίστηκε για πρώτη φορά στη επικεφαλίδα αυτής της ενότητας. Η προσέγγιση που περιγράφουμε εδώ χαρακτηρίζεται ως "σύνθεση" υπό την έννοια ότι η λύση που θα δώσουμε δεν είναι στατική, δεν αποτελείται από ένα χρονικό πρόγραμμα όπως περιγράψαμε στην πρώτη ενότητα αυτού του κεφαλαίου. Η λύση ουσιαστικά είναι το σύνολο των χρονικών προγραμμάτων που ικανοποιούν τις υποθέσεις μας.

Αυτό ουσιαστικά κάναμε θεωρώντας ότι η απάντηση στο πρόβλημά μας είναι ένα συγχρονισμένο αυτόματο. Ένα συγχρονισμένο αυτόματο, όπως και ένα αυτόματο στη γενική του μορφή, ορίζει στην πραγματικότητα ένα σύνολο ακολουθιών μεταβάσεων οι οποίες υπόκεινται σε κάποιους κανόνες - κανόνες που προκύπτουν από τον ορισμό του αυτομάτου. Υπό αυτή την έννοια, το συγχρονισμένο αυτόματο που είναι η λύση μας, θα ορίζει όλες τις δυνατές ακολουθίες ελέγξιμων μεταβάσεων οι οποίες σέβονται τους περιορισμούς του συστήματος και ικανοποιούν και τη ζητούμενη ιδιότητα.

Δεδομένου λοιπόν ενός ΣΑΠ $\mathcal{A}$, $\mathcal{A} = (S, \rightarrow, A, X, h)$ και μιας ιδιότητας P', η λύση μας θα είναι ένα ΣΑΠ $\mathcal{A}_{P'}$, $\mathcal{A}_{P'} = (S_{P'}, \rightarrow_{P'}, A_{P'}, X_{P'}, h_{P'})$ το οποίο ορίζεται ως εξής:

- $S_{P'} = S$, δηλαδή το σύνολο των διακριτών καταστάσεων του $\mathcal{A}_{P'}$ είναι ίδιο με αυτό του $\mathcal{A}$.

- $X_{P'} = X$, δηλαδή το σύνολο των ρολογιών του $\mathcal{A}_{P'}$ είναι ίδιο με αυτό του $\mathcal{A}$.

- $A_{P'} = A$.

- Η συνάρτηση $h_{P'}$ θα είναι διαφορετική από την h μόνο ως προς τις αντιστοιχήσεις που περιλαμβάνουν ελέγξιμους περιορισμούς. Πιο συγκεκριμένα, ο περιορισμός g κάθε ελέγξιμης μετάβασης $(s,(a,g,r),s')$ αντικαθίσταται στο αυτόματο $\mathcal{A}_{P'}$ από τον $g' = g \wedge Q_s^* \wedge pre_\alpha(Q_{s'}^*)$.

Είναι φυσικό η λύση μας να επηρεάζει μόνο τους ελέγξιμους περιορισμούς - είναι ακριβώς οι παράμετροι του συστήματος στους οποίους μπορούμε να επέμβουμε.

Ο αλγόριθμος που θα παρουσιάσουμε εδώ αντιμετωπίζει το πρόβλημα μόνο στις περιπτώσεις που η ιδιότητα P' είναι της μορφής $\Box P$ (δηλαδή "πάντα P") ή της μορφής $\Diamond P$ (δηλαδή "τελικά P") όπου P είναι ένα κατηγόρημα που ισχύει ή δεν ισχύει σε κάθε κατάσταση του αυτομάτου. Υπό αυτή την έννοια, $Q(P)$ είναι όλες οι καταστάσεις του αυτομάτου όπου ισχύει το κατηγόρημα P.

Για να βρούμε τις καταστάσεις του αυτομάτου στις οποίες ισχύει, για παράδειγμα η ιδιότητα $\Diamond P$, υπολογίζουμε τις καταστάσεις μέσω ενός τελεστή π έως ότου φτάσουμε σε ένα σταθερό σημείο, όπως φαίνεται παρακάτω. (Η διαδικασία για την ιδιότητα $\Box P$ είναι ακριβώς η ίδια μόνο που αντί της ένωσης παίρνουμε την τομή των συνόλων των καταστάσεων σε κάθε βήμα.)

$$\begin{array}{l} Q^0 = Q(P) \\ Repeat \\ \quad Q^{i+1} = Q^i \cup \pi(Q^i) \\ Until\ Q^i = Q^{i+1} \end{array}$$

Ο τελεστής π χαρακτηρίζει τις καταστάσεις του αυτομάτου (S) και ορίζεται ως προς ένα κατηγόρημα P. Ο τελεστής αυτός χαρακτηρίζει ουσιαστικά όλες τις καταστάσεις του ΣΑΠ από τις οποίες μπορούμε να φτάσουμε σε μία κατάσταση όπου ισχύει το P μέσω ελέγξιμης μετάβασης και από τις οποίες δεν μπορούμε να φτάσουμε σε κατάσταση όπου ισχύει το $\neg P$ μέσω μη ελέγξιμης μετάβασης. Ορίζεται ως εξής:

$$\begin{array}{lll} \pi_P((s,v)) & \alpha\nu\nu & \exists t \in R_+.(s,v) \rightarrow^t (s, v+t) \\ & & \wedge\; pre^{\varepsilon}_P(s, v+t) \;\wedge\; \neg\exists t' \in [0,t].pre^{\mu}_{\neg P}(s, v+t') \end{array}$$

Όπου ορίζουμε τα εξής σύνολα:

$$\begin{array}{lll} pre^{\varepsilon}_P(s,v) & \alpha\nu\nu & \exists a,g,s' \text{ τ.ω.}\{\; h(s,a,s') = (s,(a,g,r),s') \;\wedge (s', v[r])(P) \\ & & \wedge((a \in A^{\varepsilon} \wedge g(v)) \vee (a \in A^{\mu} \wedge d^{\varepsilon}_g(v))\} \end{array}$$

$$\begin{array}{lll} pre^{\mu}_P(s,v) & \alpha\nu\nu & \exists a,g,s' \text{ τ.ω.}\{\; h(s,a,s') = (s,(a,g,r),s') \;\wedge (s', v[r])(P) \\ & & \wedge(a \in A^{\mu} \wedge g(v))\} \end{array}$$

Τα σύνολα pre^{ε}_P και pre^{μ}_P αντιπροσωπεύουν τις καταστάσεις από τις οποίες μπορούμε να φτάσουμε σε μία κατάσταση στην οποία ισχύει το κατηγόρημα P μέσω ελέγξιμων και μη ελέγξιμων μεταβάσεων αντίστοιχα. Η προθεσμία $d^{\varepsilon}_g(v)$ ορίζεται ως: $d^{\varepsilon}_g(v) = g(v) \wedge \exists t > 0, \forall t' \in (0, t\,], \neg g(v+t')$.

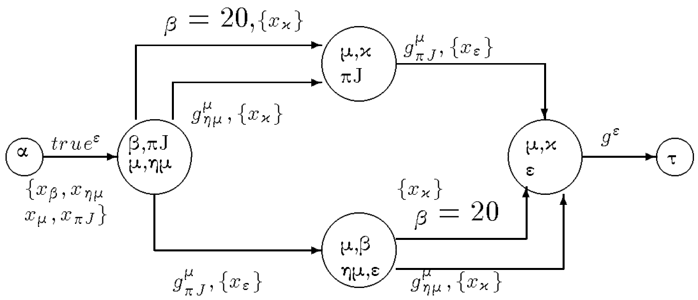
Έστω ότι το σταθερό σημείο του τελεστή π είναι ένα σύνολο καταστάσεων Q^*. Συμβολίζουμε με Q^*_s τις καταστάσεις που ανήκουν στο σταθερό σημείο και το διακριτό τους μέρος είναι η κατάσταση s.

Ας προσπαθήσουμε τώρα να ανασυντάξουμε το συνολικό αλγόριθμο:

Υπολογισμός σταθερού σημείου Q^* του τελεστή π
for all ελέγξιμες μεταβάσεις $(s,(a,g,r),s')$ **do**
  $Q^*_s := \{\langle s,v\rangle : \langle s,v\rangle \in Q^*\}$
  $Q^*_{s'} := \{\langle s',v\rangle : \langle s',v\rangle \in Q^*\}$
  $pre_{\alpha}(Q^*_{s'})$ $\alpha\nu\nu$ $\forall s' \in Q^*_{s'} \exists a, g, s''$ τ.ω.
    $\{h(s',a,s'') = (s',(a,g,r),s'') \wedge (s',v[r])(P) \wedge g(v)\}$
  $g' = g \wedge Q^*_s \wedge pre_{\alpha}(Q^*_{s'})$
  return $(s,(a,g',r),s')$

Παράδειγμα. Θα χρησιμοποιήσουμε τον παραπάνω αλγόριθμο με είσοδο το συγχρονισμένο αυτόματο που αναπαριστά το έγγραφο πολυμέσων και την ιδιότητα $\diamond$"τέλος". Αυτό που θέλουμε δηλαδή είναι ουσιαστικά η επιτυχής εκτέλεση του εγγράφου σεβόμενη τους περιορισμούς των εργασιών του.

Το αποτέλεσμα του αλγορίθμου είναι το συγχρονισμένο αυτόματο που φαίνεται στο σχήμα 4.4. Όλοι οι "υπολογισμοί" αυτού του αυτομάτου αποτελούν λύσεις του προβλήματος. Η λύση επεμβαίνει σε ένα ελέγξιμο περιορισμό - αυτόν του τερματισμού της εργασίας βίντεο. Κάθε επιτυχές χρονοπρόγραμμα του εγγράφου θα πρέπει να τερματίζει την εργασία βίντεο τη χρονική στιγμή που το ρολόι της είναι 20 ($x_\beta = 20$).



Σχήμα 4.4 Χρονοπρογραμματιστής του εγγράφου πολυμέσων

Να κλείσουμε λέγοντας ότι σε αυτό το πλαίσιο τεχνικών γίνονται προσπάθειες ανάπτυξης εφαρμογών στο χώρο του λογισμικού τηλεπικοινωνιών, όπως το εργαλείο TAXYS [BPPS99].

Κεφάλαιο 5

Επίλογος

Στο κέντρο της ενασχόλησής μας υπήρξαν στο παρόν οι τυπικές μέθοδοι και οι τεχνικές τους. Είδαμε πως αυτές οι τεχνικές μπορούν να ειδωθούν και να αναπτυχθούν για συστήματα πραγματικού χρόνου - συστήματα στα οποία η χρονική αλληλουχία των ενεργειών, των εργασιών είναι κρίσιμη. Ήρθαμε σε επαφή με ενδιαφέρουσες κατασκευές και ιδέες που απορρέουν από τον ιδιαίτερο χαρακτήρα των συστημάτων αυτών - ουσιαστικά της σχέσης τους με το χρόνο - αλλά ταυτόχρονα δεν ξεφύγαμε από το γενικότερο πλαίσιο προβληματισμού των τυπικών μεθόδων. Τόσο τα προβλήματα που αντιμετωπίσαμε όσο και οι τεχνικές που χρησιμοποιήσαμε για την αντιμετώπισή τους δε ξεφεύγουν σε καμία περίπτωση από αυτό το πλαίσιο.

Η συζήτηση είναι λοιπόν περί τυπικών μεθόδων. Ένα θέμα που έχει κατακτήσει το ενδιαφέρον των ερευνητών τα τελευταία χρόνια· ένα θέμα που "διαβάζεται" πολύ διαφορετικά από διαφορετικούς ανθρώπους. Είναι διαφορετική η ματιά ενός μαθηματικού σε αυτές, διαφορετική η ματιά ενός επιστήμονα πληροφορικής, διαφορετική η ματιά ενός μηχανικού λογισμικού και σίγουρα διαφορετική η ματιά ενός υπεύθυνου ανάπτυξης νέων προϊόντων μίας μεγάλης εταιρείας υπολογιστών.

Αν διαβάσει κανείς τις συζητήσεις που γίνονται για τις τυπικές μεθόδους θα δει τις παρανοήσεις και τις διαφωνίες που υπάρχουν μεταξύ των εμπλεκομένων σε αυτές οι οποίοι προέρχονται από ίδιους ή διαφορετικούς επιστημονικούς ή άλλους χώρους. (Ενδιαφέρον ως προς αυτή την κατεύθυνση είναι το [BH94].)

Θα προσπαθήσουμε εδώ να αναφερθούμε σε μερικά σημεία που αποτελούν κοινό τόπο και ταυτόχρονα, συνοψίζουν τις σχετικές συζητήσεις.

Δεν υπάρχει σίγουρα κανείς που να διαφωνεί ότι οι τυπικές μέθοδοι έχουν ενδιαφέρον. Ενδιαφέρον επιστημονικό που αποδεικνύεται από το πλήθος των ερευνητών που ασχολούνται με αυτές καθώς και την πληθώρα θεωρητικών αποτελεσμάτων που προκύπτουν· ενδιαφέρον διεπιστημονικό λόγω της συνάφειάς τους με διαφορετικές επιστήμες· ενδιαφέρον από την άποψη των εφαρμογών με αδιάψευστο μάρτυρα την πληθώρα εργαλείων που έχουν αναπτυχθεί.

Εξίσου αδιαμφισβήτητο είναι το γεγονός ότι οι τυπικές μέθοδοι μπορούν να δώσουν αποτελέσματα. Παρά τη νεαρή ηλικία τους, τα αποτελέσματα τόσο στη θεωρία όσο και στην πράξη είναι ιδιαίτερα αξιόλογα. Είδαμε ότι ακόμα και για συστήματα πραγματικού χρόνου, λίγα χρόνια ήταν αρκετά για να έχουμε μία καλή εικόνα της λειτουργίας τους και να φτάσουμε μέχρι και σε επίπεδο εφαρμογών. Αλλά ακόμα και για τα προβλήματα που υπάρχουν, κυρίως στην τεχνική ελέγχου μοντέλων που είναι και η πιο αξιόλογη, όλοι συμφωνούν ότι υπάρχουν πολλά περιθώρια έρευνας και λύσεων.

Από την άλλη μεριά, οι τυπικές μέθοδοι έχουν φτάσει σε μία φάση ωριμότητας η οποία μας επιτρέπει να δούμε πιο ξεκάθαρα τη φύση τους και κατ' επέκταση τη μελλοντική πορεία τους. Έχει γίνει πλέον κατανοητό ότι οι τυπικές μέθοδοι δεν εξασφαλίζουν εκ των προτέρων την ορθότητα ενός συστήματος. Δεν αποτελούν απόλυτη εγγύηση κανενός είδους ορθότητας. Ωστόσο, μας βοηθούν να κατανοήσουμε καλύτερα το υπό εξέταση σύστημα και να ανακαλύψουμε ασάφειες και ασυνέπειες.

Αρχίζει να γίνεται συνείδηση ότι ο ρόλος των τυπικών μεθόδων θα πρέπει να είναι παράλληλος με τις πιο “παραδοσιακές” μεθόδους ελέγχου της ορθότητας των συστημάτων. Παράλληλα, κρίνεται σκόπιμη η ανάγκη συνεργασίας μεταξύ των διαφορετικών τεχνικών και μεθόδων - πολλές από τις οποίες μέχρι τώρα λειτουργούσαν ανεξάρτητα.

Κλείνοντας, να εκφράσουμε για μια ακόμα φορά τη σιγουριά ότι ο χώρος των τυπικών μεθόδων γενικότερα αλλά και ειδικότερα σε ότι αφορά τα συστήματα πραγματικού χρόνου έχει να αποδώσει πολλά ακόμα ενδιαφέροντα αποτελέσματα. Όχι μόνο εφαρμογές και εργαλεία αλλά και κομψούς θεωρητικούς φορμαλισμούς· γεγονός που ίσως να αποτελεί και την πιο σημαντική συμβολή των τυπικών μεθόδων: ότι αποτελούν παράδειγμα της επιτυχούς σύμπραξης διαφορετικών επιστημών και κατευθύνσεων η οποία αποβαίνει προς όφελος όλων.

Βιβλιογραφία

[ACD90] R. Alur, C. Courcoubetis, D. Dill, Model-Checking in Dense Real-Time, *Proceedings of the 5th Anual Symposium on Logic in Computer Science*, 414-425, IEEE Computer Society Press, 1990.

[AD94] R. Alur, D. L. Dill, A Theory of Timed Automata, *TCS 126*, σ. 183-235,1994.

[AGP99] K.Altisen, G. Gößler, A. Pnueli, J. Sifakis, S. Tripakis, S. Yovine, A Framework for Scheduler Synthesis, *IEEE Real-Time Systems Symposium*, RTSS'99, 1999.

[AH91] R. Alur, T. A. Henzinger, Logics and Models of Real Time: A Survey, *Real-Time: Theory in Practice, REX Workshop, LNCS 600*, σελ. 74-106, 1991.

[BH94] J. P. Bowen, M. G. Hichey, Sever More Myths of Formal Methods, *PRG-TR-7-94*, Oxford University Computing Laboratory, 1994.

[BPPS99] V. Bertin, M. Poize, J. Pulou, J. Sifakis, Towards Validated Real-Time Software, *παραχώρηση της εργασίας από τον J. Sifakis*, 1999.

[BST98] S. Bornot, J. Sifakis, S. Tripakis, Modeling Urgency in Timed Systems, *COMPOS '97*, LNCS 1536, 1998.

[CW96] E.M. Clarke, J.M.Wing, Formal Methods: State of the Art and Future Directions, *CMU Computer Science Technical Report*, CMU-CS-96-178, 1996.

[E90] E. A. Emerson, Temporal and Modal Logic, *Handbook of Theoretical Computer Science*, Volume B, 995-1072, Elsevier Science Publisheres NH, 1990.

[HNSY94] T. A. Henzinger, X. Nicollin, J. Sifakis, S. Yovine, Symbolic Model Checking for Real-Time Systems, *Proc.* 7^{th} *Annual IEEE Symposium for Logic in Computer Science* (LICS 1992), σελ. 394-406 και *Information and Computation 111*, σελ. 193-244, 1994.

[MPS95] O. Maler, A. Pnueli, J. Sifakis, On the Synthesis of Discrete Controllers for Timed Systems, *Proc STACS '95*, LNCS 900, σελ. 229-242, 1995.

[P90] D. Perrin, Finite Automata, *Handbook of theoretical computer science*, 3-57, Elsevier Science Publishers B.V., 1990.

[Th90] W. Thomas, Automata on Infinite Objects, *Handbook of theoretical computer science*, 133-191, Elsevier Science Publishers B.V., 1990.

[Tr98] S. Tripakis, *L'Analyse Formelle des Systèmes Temporisés en Pratique*, PhD Thesis, Université Joseph Fourier-Grenoble I, 1998.