ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ενα μεταφραστικό σχήμα μεταξύ δυο μεθόδων περιγραφής συστημάτων πραγματικού χρόνου

εργασία που υποβλήθηκε από την

Μαγδαληνή Β. Χατζάκη

σαν μερική απαίτηση για την απόκτηση
Μεταπτυχιακού Διπλώματος Εξειδίκευσης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ενα μεταφραστικό σχήμα μεταξύ δυο μεθόδων περιγραφής συστημάτων πραγματικού χρόνου

εργασία που υποβλήθηκε από την

Μαγδαληνή Β. Χατζάκη

σαν μερική απαίτηση για την απόκτηση
Μεταπτυχιακού Διπλώματος Εξειδίκευσης

24 Αυγούστου, 1995

Συγγραφέας:

Μαγδαληνή Β. Χατζάκη
Τμήμα Επιστήμης Υπολογιστών

Εισηγητική Επιτροπή:

Κουρκουμπέτης Κώστας, Αναπληρωτής Καθηγητής, Πρόεδρος

Νικολάου Χρήστος, Αναπληρωτής Καθηγητής, Μέλος

Γεωργακόπουλος Γιώργος, Επίκουρος Καθηγητής, Μέλος

Δεκτή:

Κωνσταντόπουλος Πάνος, Αναπληρωτής Καθηγητής
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Κώστα Κουρκουμπέτη, που υπήρξε εμπνευστής και σύμβουλός μου, από τα πρώτα χρόνια των σπουδών μου μέχρι και σήμερα. Ιδιαίτερα τον ευχαριστώ, γιατί μου πρόσφερε την ευκαιρία να επισκευθώ το Laboratoire de Génie Informatique IMAG de Grenoble, κατά την διάρκεια των μεταπτυχιακών μου σπουδών.

Θερμά ευχαριστώ τον καθηγητή κ. Σήφη Σηφάκη που η ουσιαστική υποστήριξη και καθοδήγησή του βοήθησε στην ολοκλήρωση αυτής της εργασίας. Τον ευχαριστώ ιδιαίτερα γιατί, σε συνεργασία με τον κ. Κουρκουμπέτη, μου έδωσε την ευκαιρία να επισκευθώ το Laboratoire de Génie Informatique IMAG de Grenoble και να συνεργαστώ με την ερευνητική του ομάδα για τέσσερις μήνες (Σεπτέμβριος-Δεκέμβριος 1993). Το ευχαριστώ τέλος για την ηθική αλλά και υλική υποστήριξή του, που έκαναν πιο εύκολη την διαμονή μου στην Grenoble.

Ευχαριστώ το Πανεμιστήμιο Κρήτης και το Ινστιτούτο Πληροφορικής του Ιδρύματος Τεχνολογίας και Ερευνας για την υλικοτεχνική και οικονομική υποδομή που μου παρείχαν κατά την διάρκεια των σπουδών μου.

Ιδιαίτερα ευχαριστώ τους φίλους και συνεργάτες, Παναγιώτη Τζουνάκη, που με βοήθησε να καταλάβω βασικές έννοιες στην περιοχή της αυτόματης επαλήθευσης όταν ξεκινούσα τις μεταπτυχιακές μου σπουδές, Μαρία Καραβασίλη και Γιώργο Γεωργιανάκη για την πολύτιμη φιλία τους.

Θέλω ακόμα να ευχαριστήσω τους φίλους και συναδέλφους Sergio Yovine και Alfredo Olivero για τον χρόνο που μου αφιέρωσαν και για την πολύτιμη γνώση που μοιράστηκαν μαζί μου. Ευχαριστώ ακόμα τους Alexandra Yovine, Noel-Vidart Olivero, Claire Loiseaux και Christoph για την φιλοξενία τους.

Τέλος ευχαριστώ τους φίλους μου Behnaz Djoharia και Lionel Oddos για τους υπέροχους περιπάτους στο Vercors.

Μα περισσότερο απ' όλους, ευχαριστώ την οικογένειά μου, τον πατέρα μου Βενιζέλο, την μάνα μου Αρτεμη και τον αδερφό μου Νίκο, που η αγάπη τους με κάνει να αισθάνομαι ασφαλής και ευτυχισμένη. Με όλο μου το σεβασμό, τους αφιερώνω την εργασία μου.

Περιεχόμενα

Κατάλογος Σχημάτων

Ενα μεταφραστικό σχήμα μεταξύ
δυο μεθόδων περιγραφής
συστημάτων πραγματικού χρόνου

Μαγδαληνή Χατζάκη
Μεταπτυχιακή εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Περίληψη

Οι υπολογιστές χρησιμοποιούνται όλο και περισσότερο για να υλοποιούν και να χειρίζονται πολύπλοκες εφαρμογές, όπως τηλεπικοινωνιακά συστήματα, τραπεζικά συστήματα κλπ. Η σχεδίαση λοιπόν "σωστών συστημάτων" έχει γίνει πολύ σημαντικό θέμα.

Κάποιος μπορεί να ισχυριστεί οτι ένα σύστημα είναι σωστό αν πληρεί ένα πλήθος από ιδιότητες. Η απόδειξη της ορθότητας ενός συστήματος είναι τις περισσότερες φορές μη τετριμμένη λόγω της πολύπλοκης και μη αιτιοκρατικής συμπεριφοράς του.

Τα τελευταία χρόνια έχουν αναπτυχθεί αυτόματες μέθοδοι επαλήθευσης με κύριο στόχο να μπορούν να απαντούν στο ερώτημα αν ένα σύστημα είναι σωστό ή όχι. Τέσσερα είναι τα απαραίτητα χαρακτηριστικά που μια μέθοδος αυτόματης επαλήθευσης πρέπει να διαθέτει. Πρώτα απ' όλα ένα μαθηματικό μοντέλο για να αντιλαμβανόμαστε την έννοια του χρόνου και των συμπεριφορών του συστήματος μέσα στο χρόνο, δεύτερον, μια γλώσσα περιγραφής του συστήματος, τρίτον, μια γλώσσα περιγραφής των επιθυμητών ιδιοτήτων και τέταρτον έναν αλγόριθμο επαλήθευσης.

Σ' αυτήν την εργασία ασχολούμαστε με δυο συγκεκριμένες μεθόδους επαλήθευσης. Η πρώτη χρησιμοποιεί χρονικούς γράφους για την περιγραφή των συστημάτων, εκφράσεις TCTL για την περιγραφή ιδιοτήτων και υιοθετεί έναν αλγόριθμο επαλήθευσης συμβολικού ελέγχου μοντέλου. Ενα εργαλείο επαλήθευσης που ονομάζεται KRONOS υλοποιεί την παραπάνω μέθοδο.

Η δεύτερη χρησιμοποιεί μια διαφορετική έκδοση των χρονικών γράφων που ονομάζονται *Set/Expire* χρονικοί γράφοι για την περιγραφή των συστημάτων και των ιδιοτήτων και υιοθετεί έναν αλγόριθμο που λύνει το πρόβλημα της επαλήθευσης, λύνοντας το πρόβλημα της

έγκλεισης γλωσσών αυτομάτων. Το εργαλείο που υλοποιεί την παραπάνω μέθοδο ονομάζεται RT-COSPAN.

Οι δυο παραπάνω μέθοδοι διαφέρουν ως προς τις γλώσσες περιγραφής των συστημάτων και ως προς τους αλγορίθμους επαλήθευσης. Ετσι κάθε μέθοδος αποδίδει καλύτερα σε διαφορετικές περιπτώσεις συστημάτων. Για να συγκρίνουμε τις παραπάνω μεθόδους κυρίως ως προς την απόδοσή τους πρέπει να βρούμε ένα τρόπο να τις ενοποιήσουμε.

Αυτή η εργασία ασχολείται με τα παρακάτω θέματα: Πρώτ' απ' όλα προτείνουμε μια επέκταση του μοντέλου των *Set/Expire* χρονικών γράφων που βελτιώνει την εκφραστικότητα του παραδοσιακού μοντέλου και έχει την ιδιότητα οτι περιγραφές σ' αυτό το νέο μοντέλο μπορούν να αναλυθούν από μια ελαφρά τροποποιημένη έκδοση του RT-COSPAN την οποία έχουμε υλοποιήσει. Επιπλέον προτείνουμε ένα συντακτικά κατευθυνόμενο μεταφραστικό σχήμα που μετατρέπει ένα χρονικό γράφο σε μια ισοδύναμη περιγραφή, δοσμένη στο δικό μας μοντέλο. Ετσι μπορούμε να χρησιμοποιούμε και τα δυο εργαλεία επαλήθευσης RT-COSPAN και KRONOS σε κοινή περιγραφή του συστήματος, ανάλογα με το τι θέλουμε να αποδείξουμε. Τελικά επιχειρούμε μια προσπάθεια σύγκρισης των μεθόδων βασισμένη σε πειραματικά αποτελέσματα που παίρνουμε από την εφαρμογή των δυο εργαλείων επαλήθευσης σε μη τετριμμένα παραδείγματα.

A translation scheme between two real-time formalisms

Magda Chatzaki
Master of Science thesis

Departement of Computer Science
University of Crete

Abstract

Computer programs are more and more used to manipulate real-time applications such as telecommunication applications, banking systems etc. The design of "correct systems" has become a very important issue.

One can state that a system is correct if it preserves a number of properties. Proving correctness is a non-trivial task because of the complex and non deterministic behavior of the systems.

Automatic verification methods have been developed in order to answer the question whether the system is correct or not. The four basic elements that an automatic verification method must assume are, a model for the time quantity, a language for modeling the system, a language for modeling the desired properties and a verification algorithm.

In this thesis, we deal with two particular automatic verification formalisms. The first formalism uses timed automata (timed graphs) for modeling the system, TCTL formulas for modeling the properties and adopts a symbolic model checking verification algorithm. The verification tool KRONOS implements the above formalism.

The second one uses a different version of timed automata, namely the *Set/Expire* timed automata for modeling real-time systems and their properties and adopts an automata oriented verification algorithm which solves the language inclusion problem. The verification tool RT-COSPAN implements this formalism.

The two methods differ in the system description languages and also in the verification algorithms. That is why each formalism is suited better to different cases of real-time systems. In order to compare their efficiency and performance we have to deal with the

unification problem of these two formalisms.

This thesis addresses the following issues: First of all we propose an extension of the *Set/Expire* timed automata which improves the expressiveness of the traditional *Set/Expire* timed automata and has the property that can be analyzed using a slightly different version of RT-COSPAN which we implement. Further more we propose a syntax directed translation scheme that transforms a timed automata specification to a specification written in our model. In this way we can use both verification tools RT-COSPAN and KRONOS on a common description of a real-time system depending on what we want to verify. Finally we try a comparison of the two methods through experimental results.

Κεφάλαιο 1

Εισαγωγή

Οι υπολογιστές χρησιμοποιούνται όλο και περισσότερο για να υλοποιούν και να χειρίζονται πολύπλοκες εφαρμογές, που η ορθή λειτουργία τους είναι πολύ κρίσιμη για την καθημερινή ζωή. Για παράδειγμα, εφαρμογές στις τηλεπικοινωνίες, τραπεζικά συστήματα, συστήματα ανάπτυξης βιομηχανικών εφαρμογών, όπως ενδεικτικά αναφέρουμε την κατασκευή οχημάτων, αεροσκαφών, συστήματα ελέγχου φαναριών κυκλοφορίας, πλοήγησης οχημάτων κλπ. Ολες οι παραπάνω εφαρμογές έχουν τα εξής χαρακτηριστικά:

Είναι συστήματα "πεπερασμένων καταστάσεων". Δηλαδή η εφαρμογή αποτελείται από ένα πεπερασμένο αριθμό βημάτων. Ετσι μπορούμε να περιγράψουμε τέτοιου είδους εφαρμογές, επιλέγοντας ένα μαθηματικό μοντέλο, με ένα πεπερασμένο αριθμό καταστάσεων.

Είναι συστήματα που "αντιδρούν στα ερεθίσματα που δέχονται από το περιβάλλον τους". Με άλλα λόγια, η συμπεριφορά της εφαρμογής μας εξαρτάται από το περιβάλλον στο οποίο μέσα έχει τοποθετηθεί.

Είναι συστήματα "πραγματικού χρόνου". Δηλαδή το σύστημά μας για να εκπληρώσει τις λειτουργικές ανάγκες για τις οποίες κατασκευάστηκε, πρέπει να σέβεται κάποιους χρονικούς περιορισμούς.

Λανθασμένη λειτουργία του συστήματος είναι πολλές φορές επικίνδυνη και μπορεί να αποβεί μοιραία. Σαν αποτέλεσμα, η σχεδίαση "σωστών συστημάτων", είναι πολύ σημαντική. Κάποιος μπορεί να ισχυρισθεί οτι ένα σύστημα λειτουργεί "σωστά" ως προς τις προδιαγραφές του, αν ικανοποιεί ένα πλήθος επιθυμητών ιδιοτήτων. Για παράδειγμα, το σύστημα ελέγχου σε μια φυλαγμένη διάβαση μιας σιδηροδρομικής γραμμής πρέπει να σέβεται την ιδιότητα: P = "η μπάρα πρέπει να κατέβει τουλάχιστο 3 λεπτά πριν το τραίνο μπει στη διάβαση".

Ο έλεγχος της ορθότητας συστημάτων, είναι γενικά ένα αρκετά δύσκολο πρόβλημα λόγω της πολύπλοκης και τις περισσότερες φορές μη αιτιοκρατικής συμπεριφοράς των συστημάτων. Η πολυπλοκότητα του προβλήματος της επαλήθευσης αυξάνεται τρομακτικά για συστήματα πραγματικού χρόνου. Αυτό οφείλεται γιατί στα συστήματα πραγματικού χρόνου δεν μας ενδιαφέρει μόνο μια ενέργεια του συστήματος αλλά και η χρονική στιγμή στην οποία συμβαίνει. Δεδομένου οτι ο χρόνος είναι ένας πραγματικός αριθμός καταλαβαίνουμε οτι τα συστήματα μας έχουν άπειρες το πλήθος δυνατές συμπεριφορές.

Η ανάγκη λοιπόν να βρούμε αποδοτικούς τρόπους επαλήθευσης συστημάτων γίνεται όλο και πιο επιτακτική.

Αλλά ποια είναι η καλύτερη μέθοδος για να συνάγουμε αν το σύστημά μας είναι σωστό ή όχι; Ενας μερικός τρόπος επίλυσης του προβλήματος είναι να προσομοιώσουμε τη λειτουργία του. Ενας άλλος τρόπος, που έχει κερδίσει το ερευνητικό ενδιαφέρον τα τελευταία χρόνια, είναι οι μέθοδοι "αυτόματης επαλήθευσης". Οι μέθοδοι αυτόματης επαλήθευσης βασίζονται στην εξής ιδέα: Περιγράφουμε το σύστημα μας με ένα πεπερασμένο τρόπο καθώς και την ιδιότητα που το σύστημα πρέπει να ικανοποιεί. Κατόπιν εφαρμόζουμε ένα αλγόριθμο ανάλυσης ο οποίος απαντά αν η επιθυμητή ιδιότητα ισχύει σ' αυτό το σύστημα.

Σ' ετούτη την εργασία θα μιλήσουμε κυρίως για "τυπικές μεθόδους" και για "εργαλεία περιγραφής" και "επαλήθευσης" συστημάτων "πραγματικού χρόνου πεπερασμένων καταστάσεων" και θα προτείνουμε μια θεωρία ενοποίησης για δυο τέτοιες μεθόδους.

1.1 Κίνητρα

Τα τελευταία χρόνια, πολλές διαφορετικές μέθοδοι έχουν αναπτυχθεί για την περιγραφή και επαλήθευση συστημάτων πραγματικού χρόνου, που καθεμιά χρησιμοποιεί διαφορετικό συντακτικό και σημασιολογικό συμβολισμό, για να φτάσουν όλες στον ίδιο σκοπό, που είναι η περιγραφή της συμεριφοράς και ακολούθως η επαλήθευση συστημάτων πραγματικού χρόνου.

Σαν αποτέλεσμα, είναι απαραίτητο να προσπαθήσουμε να συγκρίνουμε και να κατηγοριοποιήσουμε όλες αυτές τις διαφορετικές μεθόδους, να βρούμε τις ομοιότητες και τις διαφορές τους, να τις συγκρίνουμε ως προς την εκφραστικότητα και την απόδοσή τους, με άλλα λόγια να βρούμε τι έχει επιτευχθεί μέχρι τώρα.

Εχοντας πεισθεί για την αναγκαιότητα της παραπάνω σύγκρισης, εντοπίζουμε την προσοχή μας σε δυο διαφορετικές μεθόδους που έχουν αναπτυχθεί τα τελευταία χρόνια και έχουν ήδη εφαρμοσθεί στην υλοποίηση δυο εργαλείων. Η βασική διαφορά αυτών των δυο μεθόδων έγκειται στους διαφορετικούς αλγορίθμους που χρησιμοποιούν για την ανάλυση των συστημάτων.

Η πρώτη μέθοδος, [Dil89], προτείνει ένα τρόπο για την επέκταση ενός χρονικά ανεξάρτητου μοντέλου ενός συστήματος με χρονική πληροφορία. Αυτή η μέθοδος χρησιμοποιεί σημασιολογία ιχνών και επεκτείνει το παραδοσιακό μοντέλο των πεπερασμένων αυτομάτων με ένα σύνολο από συνθήκες, που ορίζουν τις χρονικές αποστάσεις μεταξύ γεγονότων του συστήματος. Το βασικό μειονέκτημα αυτής της μεθόδου είναι οτι αποτυγχάνει να περιγράψει αποδοτικά μη αιτιοκρατικές χρονικές συμπεριφορές, time-outs κλπ. Για την επαλήθευση χρησιμοποιεί έναν αλγόριθμο που ουσιαστικά λύνει το πρόβλημα της έγκλεισης γλωσσών πεπερασμένων αυτομάτων. Ενα εργαλείο επαλήθευσης που ονομάζεται RT-COSPAN [CDT92, CDCT92, Tzo92] έχει αναπτυχθεί χρησιμοποιώντας την παραπάνω μέθοδο.

Η δεύτερη, [HNSY92], όπως και η πρώτη, χρησιμοποιεί μια επέκταση των πεπερασμένων αυτομάτων για την περιγραφή του συστήματος και εφαρμόζει έναν αλγόριθμο συμβολικού ελέγχου μοντέλου για την απόδειξη της ορθότητάς του. Το εργαλείο επαλήθευσης KRONOS έχει αναπτυχθεί σύμφωνα με την παραπάνω μέθοδο.

Οι διαφορές των μοντέλων περιγραφής μεταξύ της πρώτης μεθόδου και της δεύτερης είναι κυρίως συντακτικές. Τα εκτεταμένα αυτόματα που χρησιμοποιεί η δεύτερη μέθοδος ονομάζονται "χρονικά αυτόματα" ή "χρονικοί γράφοι", τα οποία είναι μη αιτιοκρατικά πεπερασμένα αυτόματα που έχουν επεκταθεί με ένα σύνολο μεταβλητών, που παίρνουν πραγματικές τιμές και ονομάζονται "ρολόγια", και καθορίζουν τις χρονικές αποστάσεις μεταξύ γεγονότων του συστήματος. Η ιδέα της κωδικοποίησης της χρονικής πληροφορίας είναι η ίδια και στην πρώτη μέθοδο μόνο που διαφέρουν συντακτικά οι δυο τρόποι περιγραφής. Η διαχείρηση της χρονικής πληροφορίας και στα δυο μοντέλα γίνεται με το να αναγκάζουμε τη λειτουργία του αυτομάτου να εξαρτάται και από τις τιμές των μεταβλητών που μετράνε το χρόνο.

Στα πλαίσια λοιπόν της γενικότερης σύγκρισης των μεθόδων αυτόματης επαλήθευσης ετούτη η εργασία έχει τρεις βασικούς στόχους:

Την δυνατότητα χρήσης των παραπάνω δυο μεθόδων επαλήθευσης σε κοινή περιγραφή του συστήματος.

Την συγκριτική μελέτη των αποτελεσμάτων ώστε να βρούμε τις περιπτώσεις των προβλημάτων στα οποία κάθε μέθοδος συμπεριφέρεται καλύτερα.

Την ενδεχόμενη βελτίωση των μεθόδων συνδυάζοντας τους αλγορίθμους ανάλυσης που χρησιμοποιούν.

Για να μπορέσουμε να φτάσουμε στους παραπάνω στόχους, στα πλαίσια αυτής της εργασίας, προτείνουμε πρώτ' απ' όλα μια επέκταση της πρώτης μεθόδου και ακολούθως ένα μεταφραστικό σχήμα που ενοποιεί τις παραπάνω δυο μεθόδους ως προς την περιγραφή του συστήματος. Τέλος τις συγκρίνουμε μέσα από πειραματικά αποτελέσματα που παίρνουμε από την εφαρμογή των δυο εργαλείων σε συγκεκριμένα παραδείγματα.

1.2 Ανασκόπηση

Μια τυπική μέθοδος για την περιγραφή και επαλήθευση συστημάτων πρέπει να διαθέτει:

1. Ενα μαθηματικό μοντέλο για να αντιλαμβανόμαστε τις συμπεριφορές του συστήματος. Ειδικότερα όταν τα συστήματά μας περιέχουν και χρονική πληροφορία, θα πρέπει να διαθέτει και ένα μοντέλο για τον "χρόνο".

2. Μια γλώσσα I για την περιγραφή του συστήματος που ονομάζεται "γλώσσα υλοποίησης".

3. Μια γλώσσα S για την περιγραφή των ιδιοτήτων που ονομάζεται "γλώσσα περιγραφής".

4. Ενα αλγόριθμο που είναι δύνατον να απαντήσει στο ερώτημα, αν ένα συγκεκριμένο σύστημα πληρεί μια δοσμένη ιδιότητα. Ο αλγόριθμος αυτός ονομάζεται "αλγόριθμος επαλήθευσης".

Αυτός είναι ο βασικός σκελετός για την περιγραφή συστημάτων. Για καθένα από τα παραπάνω χαρακτηριστικά έχουν προοταθεί πολλές διαφορετικές εναλλακτικές. Στις επόμενες παραγράφους θα κάνουμε μια μικρή ανασκόπηση των διαφόρων προτάσεων για το καθένα.

1.2.1 Γλώσσες υλοποίησης

Οι γλώσσες υλοποίησης μπορούν να χωριστούν σε δυο μεγάλες κατηγορίες: Σε γλώσσες περιγραφής συστημάτων με τη βοήθεια "αυτομάτων" και σε γλώσσες περιγραφής συστημάτων με τη βοήθεια μιας "άλγεβρας διεργασιών".

Στην πρώτη κατηγορία, χρησιμοποιούνται γράφοι για την περιγραφή του συστήματος. Κάθε μετάβαση του γράφου αντιστοιχεί σε ένα λογικό βήμα του συστήματος. Αυτοί οι γράφοι είναι στην πραγματικότητα επεκτάσεις των γνωστών πεπερασμένων αυτομάτων που χρησιμοποιούνται για να ορίζουν τις πεπερασμένες κανονικές γλώσσες. Διαφορετικά είδη πεπερασμένων αυτομάτων έχουν προταθεί. Τα πιο ενδιαφέροντα είναι τα *Büchi* [Buc62] ή τα *Müller* [Alu91] αυτόματα. Το σύστημα λοιπόν περιγράφεται σαν ένα μη αιτιοκρατικό πεπερασμένο αυτόματο. Οι συμπεριφορές του συστήματος εξηγούνται σαν ίχνη άπειρου μήκους πάνω στο αλφάβητο του αυτομάτου. Το αλφάβητο περιέχει στην ουσία τις λογικές ενέργειες του συστήματος. Η γλώσσα που δέχεται το αυτόματο αντιστοιχεί στο σύνολο των πιθανών συμπεριφορών του συστήματος. Πράξεις πάνω σε αυτόματα, χρήσιμες για την περιγραφή πιο πολύπλοκων συστημάτων, μπορούν να θεωρηθούν συνολοθεωρητικές πράξεις. Για παράδειγμα, η παράλληλη σύνθεση μπορεί να υλοποιηθεί με μια πράξη προβολής και κατόπιν τομής δυο συνόλων.

Στη δεύτερη κλάση γλωσσών υλοποίησης ανήκουν οι άλγεβρες διεργασιών. Οι άλγεβρες διεργασιών είναι γενικά γλώσσες που διαθέτουν ένα συντακτικό (όπως ακριβώς οι γλώσσες προγραμματισμού) και κατάλληλα ορισμένους τελεστές που εφαρμόζονται πάνω σε εκφράσεις της γλώσσας για να κατασκευαστούν μεγαλύτερες, πιο πολύπλοκες εκφράσεις. Διάφορες άλγεβρες διεργασιών έχουν προταθεί όπως η CCS [Mil80] ή η CSP [Hoa85].

Οι παραπάνω μέθοδοι έχουν επεκταθεί με σκοπό να είναι δυνατόν να υποστηρίζουν τις απαιτήσεις για περιγραφές συστημάτων πραγματικού χρόνου. Οπως αναφέραμε και προλίγου τα συστήματα πραγματικού χρόνου πρέπει να πληρούν κάποιες χρονικές απαιτήσεις. Για την περιγραφή συστημάτων πραγματικού χρόνου χρειάζεται να ορίσουμε ένα μοντέλο για την ποσότητα του χρόνου. Οι πρώτες προσπάθειες των ερευνητών ήταν να επεκτείνουν τα μέχρι τώρα γνωστά, χρονικά ανεξάρτητα μοντέλα, με την πληροφορία του χρόνου. Ετσι αρχικά πρόσθεσαν τον "χρόνο" στα μοντέλα σαν μια διακριτή ποσότητα. Το μοντέλο λοιπόν του χρόνου είναι οι ακέραιοι αριθμοί. Κάθε λογικό βήμα του συστήματος υποτείθεται οτι λαμβάνει χώρα σε ένα χρονικό βήμα (σύγχρονα μοντέλα).

Αλλά μιας και ο χρόνος είναι από τη φύση του μια συνεχής ποσότητα, τα παραπάνω μοντέλα ήταν ανακριβή για την περιγραφή ασύγχρονων συμπεριφορών.

Ετσι αναπτύχθηκαν μοντέλα για συνεχή χρόνο. Τα Büchi αυτόματα έχουν επεκταθεί σε χρονικά αυτόματα [AD90, Alu91, Dil89]. Τα χρονικά αυτόματα είναι μη αιτιοκρατικά αυτόματα πεπερασμένων καταστάσεων που έχουν επεκταθεί με ένα πεπερασμένο σύνολο ρολογιών που καταγράφουν το πέρασμα του χρόνου. Αλλες προτάσεις [Dil89, Lew90] επεκτείνουν τα παραδοσιακά πεπερασμένα αυτόματα, με ένα σύνολο από περιορισμούς που ορίζουν τις χρονικές αποστάσεις μεταξύ των γεγονότων του συστήματος.

Οι άλγεβρες διαδικασιών έχουν πάλι επεκταθεί για να περιγράφουν συστήματα πραγματικού χρόνου. Η ATP [NRSV90, NS90] είναι παράδειγμα μιας άλγεβρας διεργασιών για περιγραφή ασύγχρονων συστημάτων πραγματικού χρόνου. Η ATP υποστηρίζει ένα σύνολο από χρονικούς τελεστές που περιγράφουν συμπεριφορές όπως time-outs, watchdogs etc. Μια μέθοδος για να μεταφράζεται ένα ATP πρόγραμμα σε ένα συγκεκριμένο είδος χρονικών αυτομάτων προτείνεται στα [Yov92, Yov93].

1.2.2 Γλώσσες περιγραφής

Οι γλώσσες περιγραφής μπορούν επίσης να διαχωριστούν σε δυο μεγάλες κατηγορίες: Στις γλώσσες περιγραφής ιδιοτήτων με τη βοήθεια "αυτομάτων" και στις "χρονικές λογικές".

Οι γλώσσες περιγραφής με τη βοήθεια αυτομάτων χρησιμοποιούν μη αιτιοκρατικά πεπερασμένα αυτόματα για να περιγράφουν ιδιότητες που πρέπει να πληρεί το σύστημα.

Οι χρονική λογική προτάθηκε αρχικά από τον Pnuelli [Pnu85] σαν ένας τρόπος να εκφράζονται ιδιότητες συστημάτων. Οι χρονικές λογικές διαθέτουν ένα πεπερασμένο σύνολο από στοιχειώδεις προτάσεις, μαζί με ένα πεπερασμένο σύνολο από τροπικούς τελεστές που ερμηνεύονται με χρονικό τρόπο. Εχουν προταθεί πολλές διαφορετικές χρονικές λογικές όπως η PTL, CTL [EC82] ή CTL^* [CES86].

Ολες αυτές έχουν επεκταθεί κατάλληλα ώστε να μπορούν να περιγράφουν χρονικές ιδιότητες. Οι επεκτάσεις αυτές είναι οι TPTL (*Timed Propositional Temporal Logic*) ή η TCTL (*Timed Computation Tree Logic*) TCTL.

Για την περιγραφή των ιδιοτήτων ενός συστήματος χρησιμοποιούνται "εκφράσεις" χρονικής λογικής. Μια έκφραση, είναι μια συντακτική δομή που χρησιμοποιεί λογικούς και χρονικούς τελεστές για να εκφράσει την επιθυμητή ιδιότητα.

1.2.3 Αλγόριθμοι επαλήθευσης

Τελικά οι αλγόριθμοι επαλήθευσης μπορούν και αυτοί να διαχωριστούν σε δυο μεγάλες κατηγορίες:
σε αλγορίθμους που μετατρέπουν το πρόβλημα της επαλήθευσης στο πρόβλημα "έγκλεισης γλωσσών αυτομάτων" και σε αλγορίθμους "ελέγχου μοντέλου".

Οταν και το σύστημα και η ιδιότητα περιγράφονται σαν πεπερασμένα αυτόματα, το πρόβλημα της επαλήθευσης (αν το σύστημα ικανοποιεί την ιδιότητα) ανάγεται στο πρόβλημα της έγκλεισης γλωσσών (αν η γλώσσα του αυτομάτου που περιγράφει το σύστημα, είναι υποσύνολο της γλώσσας του αυτομάτου που περιγράφει την ιδιότητα). Αυτού του είδους οι αλγόριθμοι μπορούν να εφαρμοσθούν και για την επαλήθευση συστημάτων πραγματικού χρόνου. Τα εργαλεία επαλήθευσης COSPAN [HK90, KH87] και RT-COSPAN [CDT92, CDCT92, Tzo92] υιοθετούν κάποιον τέτοιο αλγόριθμο.

Ο έλεγχος μοντέλου από την άλλη πλευρά, είνα μια μέθοδος επαλήθευσης που μπορεί να εφαρμοσθεί όταν ένα σύστημα περιγράφεται σαν ένα σύστημα μεταβάσεων και η ιδιότητα περιγράφεται σαν μια έκφραση χρονικής λογικής. Η ιδέα του ελέγχου μοντέλου στηρίζεται στον υπολογισμό του συνόλου των καταστάσεων του συστήματος μεταβάσεων πάνω στις οποίες ισχύει η ιδιότητα. Κάμποσοι αλγόριθμοι που στηρίζονται στον έλεγχο μοντέλου έχουν προταθεί [QS81, CES86]. Κάποιοι από αυτούς, έχουν επεκταθεί για την επαλήθευση συστημάτων πραγματικού χρόνου.

Ο σημαντικότερος ανασταλτικός παράγοντας για την απόδοση αυτών των αλγορίθμων οφείλεται στο μεγάλο πλήθος καταστάσεων που ένας τέτοιος αλγόριθμος πρέπει να ελέγξει. Ο έλεγχος μοντέλου για την επαλήθευση συστημάτων πραγματικού χρόνου γίνεται ακόμα πιο πολύπλοκος, λόγω του οτι οι συμπεριφορές των συστήματων είναι άπειρες το πλήθος. Ετσι οι παραδοσιακοί αλγόριθμοι έλεγχου μοντέλου δεν μπορούν να χρησιμοποιηθούν για να επαληθεύσουν συστήματα πραγματικού χρόνου. Μόνο πρόσφατα κατασκευάστηκε ένας αλγόριθμος ελέγχου μοντέλου [ACD90] για συστήματα πραγματικού χρόνου. Αυτός ο αλγόριθμος βασίζεται στην κατασκευή ενός πεπερασμένου γράφου από το άπειρο μοντέλο που περιγράφει το σύστημα. Ο γράφος αυτός ονομάζεται "γράφος περιοχών" και οι καταστάσεις του είναι σύνολα καταστάσεων του αυθεντικού μοντέλου του συστήματος, που έχουν συγκεντρωθεί βάση μιας σχέσης ισοδυναμίας.

Ενας άλλος αλγόριθμος για "συμβολικό έλεγχο μοντέλου" έχει προταθεί στο [HNSY92]. Αυτός ο αλγόριθμος δεν κατασκευάζει τον γράφο περιοχών αλλά υπολογίζει το σύνολο των καταστάσεων που ικανοποιούν την ιδιότητα "συμβολικά", σαν ένα όριο μιας συνάρτησης λογικών κατηγορημάτων που αντιπροσωπεύουν τις καταστάσεις του συστήματος.

Το εργαλείο επαλήθευσης KRONOS [OY92] χρησιμοποιεί αυτόν τον αλγόριθμο. Δοσμένης δηλαδή μιας περιγραφής του συστήματος και μιας έκφρασης χρονικής λογικής, το εργαλείο μας απαντά αν η έκφραση είναι αληθής για το σύστημα αυτό ή όχι.

Βλέπουμε οτι αυτές οι δυο κατηγορίες αλγορίθμων διαφέρουν ως προς την φιλοσοφία που αντιμετωπίζουν την επίλυση του προβλήματος της επαλήθευσης. Αυτό τους διαφοροποιεί και ως προς την απόδοσή τους, ανάλογα με το σύστημα και τις ιδιότητες για τις οποίες εφαρμόζονται. Στα πλαίσια της γενικότερης σύγκρισης των μεθόδων επαλήθευσης θα θέλαμε να βρούμε τις περιπτώσεις που κάθε αλγόριθμος θα συμπεριφερόταν καλύτερα. Ετσι, ανάλογα με το τι θέλουμε να αποδείξουμε, θα μπορούσαμε να χρησιμοποιούμε ένα από τα δυο εργαλεία που αναφέραμε παραπάνω και που αντιπροσωπεύουν τις δυο κατηγορίες αλγορίθμων που προαναφέραμε. Οι τρόποι όμως που περιγράφουμε ένα σύστημα στο ένα εργαλείο ή στο άλλο διαφέρουν μεταξύ τους συντακτικά. Αυτό λοιπόν που λείπει είναι να ενοποιήσουμε τις δυο αυτές μεθόδους ως προς την περιγραφή του συστήματος, και έπειτα να εφαρμόσουμε τους διαφορετικούς αλγορίθμους για να δούμε την απόδοσή τους. Γι' αυτόν ακριβώς το λόγο σε αυτή την εργασία ασχολούμαστε με την ανάπτυξη μιας θεωρίας ενοποίησης των δυο μεθόδων ως προς την περιγραφή.

1.3 Οργάνωση της εργασίας

Στο **κεφάλαιο 2** παρουσιάζουμε μοντέλα για την περιγραφή συστημάτων πραγματικού χρόνου. Δίνουμε στοιχειώδεις ορισμούς που θα βοηθήσουν στην κατανόηση των μοντέλων που θα αναπτύξουμε στη συνέχεια. Παρουσιάζουμε δυο διαφορετικά μοντέλα που και τα δυο βασίζονται στην περιγραφή συστημάτων με τη βοήθεια χρονικών αυτομάτων.

Στο πρώτο μοντέλο η χρονική πληροφορία εκφράζεται με ένα σύνολο χρονικών περιορισμών. Τα αυτόματα που χρησιμοποιούνται ονομάζονται *Set/Expire* χρονικοί γράφοι. Το δεύτερο μοντέλο χρησιμοποιεί χρονικούς γράφους, για την περιγραφή των συστημάτων, που είναι στη βάση τους Büchi αυτόματα που έχουν επεκταθεί με ένα πεπερασμένο σύνολο ρολογιών. Οπως παρατηρούμε, τα παραπάνω μοντέλα χρησιμοποιούν πεπερασμένα αυτόματα για να περιγράφουν συστηματα. Ομως ο τρόπος έκφρασης της χρονικής πληροφορίας διαφέρει σημαντικά.

Στο **κεφάλαιο 3** αναφερόμαστε στους τρόπους με τους οποίους μπορούμε να περιγράφουμε ιδιότητες των συστημάτων. Ειδικότερα θα αναφερθούμε στην περιγραφή ιδιοτήτων με τη βοήθεια αυτομάτων και με χρονικές λογικές.

Στο **κεφάλαιο 4** αναπτύσσουμε δυο αλγορίθμους επαλήθευσης. Ο πρώτος είναι ένας αλγόριθμος που ανάγει το πρόβλημα της επαλήθευσης συστημάτων πραγματικού χρόνου στο πρόβλημα έγκλεισης γλωσσών αυτομάτων [Dil89, CDCT92], και ο δεύτερος είναι ένας αλγόριθμος συμβολικού ελέγχου μοντέλου [HNSY92].

Στο **κεφάλαιο 5** προτείνουμε μια επέκταση των *Set/Expire* χρονικών γράφων και ένα μεταφραστικό σχήμα για να μεταφράζουμε έναν χρονικό γράφο σε έναν *Set/Expire* χρονικό γράφο.

Στο **κεφάλαιο 6** παρουσιάζουμε συνοπτικά τα εργαλεία RT-COSPAN και KRONOS που υλοποιούν τις μεθόδους που αναπτύξαμε στα προηγούμενα κεφάλαια. Χρησιμοποιούμε αυτά τα δυο εργαλεία να να πετύχουμε τη σύγκριση των παραπάνω μεθόδων.

Στο **κεφάλαιο 7** δίνουμε ένα παράδειγμα, παρουσιάζουμε τα αποτελέσματα της επαλήθευσης που παίρνουμε εφαρμόζοντας τα δυο εργαλεία και προσπαθούμε να δώσουμε κάποια συμπεράσματα για την απόδοση των δυο μεθόδων. Στο **Παράρτημα Α** δίνουμε το πρωτόκολλο FDDI γραμμένο στην γλώσσα περιγραφής του KRONOS και του RT-COSPAN αντίστοιχα.

Τέλος στο **κεφάλαιο 8** συζητάμε τα συμπεράσματα και ιδέες για μελλοντική εργασία.

Κεφάλαιο 2

Μοντέλα συστημάτων πραγματικού χρόνου

Οπως αναφέραμε στο εισαγωγικό κεφάλαιο οι μέθοδοι περιγραφής και αυτόματης επαλήθευσης χρειάζονται τέσσερα βασικά στοιχεία: ένα μαθηματικό μοντέλο για να αντιλαμβανόμαστε τις συμπεριφορές του συστήματος, μια γλώσσα περιγραφής του συστήματος, μια γλώσσα περιγραφής των ιδιοτήτων και ένα αλγόριθμο επαλήθευσης.

Σε αυτό το κεφάλαιο θα παρουσιάσουμε εναλλακτικές προτάσεις για τα δυο πρώτα στοιχεία του παραπάνω σχήματος.

2.1 Συστήματα χωρίς χρονική πληροφορία

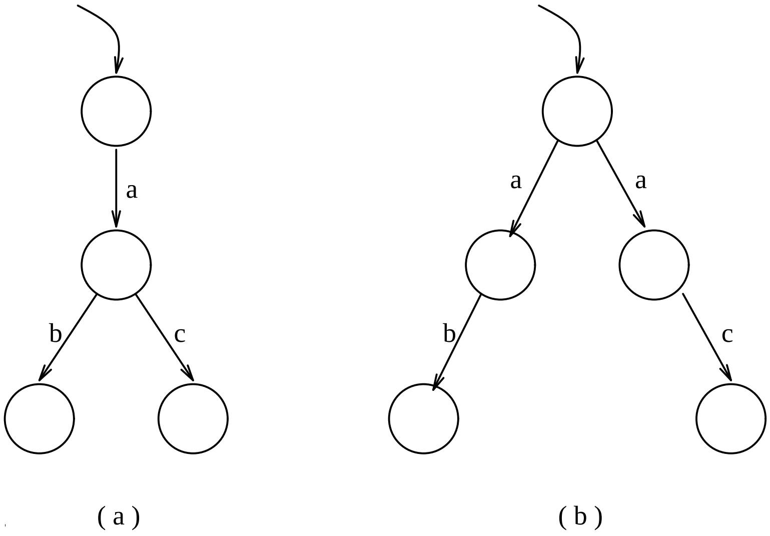
Σ' αυτή την ενότητα δίνουμε εισαγωγικούς ορισμούς και παραδείγματα, με σκοπό να θέσουμε το θεωρητικό υπόβαθρο ανάπτυξης μοντέλων για συστήματα πραγματικού χρόνου.

2.1.1 Συστήματα μεταβάσεων με ετικέττες

Ενα σύστημα μπορούμε να το αναπαραστήσουμε σαν ένα σύνολο από ακολουθίες καταστάσεων, κάθε μια από τις οποίες περιγράφει μια πιθανή εξέλιξη (συμπεριφορά) του συστήματος. Ενας γενικός τρόπος περιγραφής, όχι μόνο των συμπεριφορών, αλλά και της εσωτερικής δομής του συστήματος είναι τα "συστήματα μεταβάσεων με ετικέττες" (*labeled transition systems*).

Ενα "σύστημα μεταβάσεων με ετικέττες" ορίζεται σαν μια τετράδα $B = (\Sigma, S, \rightarrow, s_0)$, όπου

- Σ είναι ένα αλφάβητο,



Σχήμα 2.1: Περιγραφή συστημάτων από συστήματα μεταβάσεων με ετικέττες

- S είναι ένα πεπερασμένο σύνολο καταστάσεων,
- $\rightarrow \subseteq S \times \Sigma \times S$ είναι μια σχέση μετάβασης,
- s_0, είναι μια αρχική κατάσταση

Οι μεταβάσεις ενός συστήματος μεταβάσεων με ετικέττες είναι της μορφής $s \xrightarrow{\sigma} s'$, όπου $s, s' \in S$ και $\sigma \in \Sigma$. Το αλφάβητο Σ αποτελεί το σύνολο των "ενεργειών", (γεγονότων) του συστήματος. Οι ενέργειες του συστήματος θεωρούνται "ατομικές", (στοιχειώδεις) και "στιγμιαίες". Η εκτέλεση ενός γεγονότος του συστήματος σ αναπαριστάται με μια μετάβαση του συστήματος μεταβάσεων, από μια κατάσταση s σε μια κατάσταση s' και με ετικέττα σ.

Σαν παράδειγμα ας δούμε την εικόνα 2.1. Οπως είναι φανερό τα συστήματα μεταβάσεων με ετικέττες, δεν περιγράφουν μόνο ακολουθίες γεγονότων του συστήματος, αλλά και την εσωτερική δενδρική του δομή.

Συστήματα μεταβάσεων με ετικέττες που περιγράφουν απλά συστήματα, μπορούν να συνδυαστούν για να περιγράψουν πιο πολύπλοκα. Ορίζεται λοιπόν ένας λογισμός πάνω στα συστήματα μεταβάσεων, με τελεστές όπως της "παράλληλης σύνθεσης", "απόκρυψης", "προθέματος" κλπ. Το αποτέλεσμα αυτών των πράξεων είναι και πάλι ένα σύστημα μεταβάσεων με ετικέττες. Ετσι για ένα σύστημα που περιέχει πάνω από μια συνιστώσα, περιγράφουμε μια μια χωριστά με ένα σύστημα μεταβάσεων και έπειτα τις συνδιάζουμε με τους τελεστές που αναφέραμε μόλις πριν. Περισσότερες πληροφορίες γύρω από τα συστήματα μεταβάσεων με ετικέττες δίνονται στο [Sif82].

2.1.2 Ιχνη και διεργασίες χωρίς χρονική πληροφορία

Για να εξηγήσουμε τα συστήματα μεταβάσεων με ετικέττες μπορούμε να χρησιμοποιήσουμε "σημασιολογία ιχνών". Από το πρίσμα της σημασιολογίας ιχνών, οι υπολογισμοί ενός συστήματος μπορούν να θεωρηθούν σαν άπειρα ίχνη πάνω σε ένα σύνολο από παρατηρήσιμα γεγονότα. Ενα σύστημα λοιπόν περιγράφεται από το σύνολο όλων των ιχνών του.

Με σημασιολογία ιχνών δεν μπορούμε να δώσουμε καμία πληροφορία για την εσωτερική δομή του συστήματος. Ας δούμε για παράδειγμα τα δυο συστήματα της εικόνας 2.1. Τα συστήματα μετάβασης που περιγράφουν τα δυο συστήματα, έχουν διαφορετικές δενδρικές δομές και είναι πράγματι διαφορετικά αν τα συγκρίνουμε από την λειτουργική τους σκοπιά. Για παράδειγμα το σύστημα της εικόνας 2.1(a), έχει μια κατάσταση, απ' όπου μπορεί μη αιτιοκρατικά να διαλέξει να εκτελέσει είτε b είτε c, αφού πρώτα έχει εκτελέσει a. Δεν συμβαίνει όμως το ίδιο για το σύστημα της εικόνας 2.1(b). Αλλά αν συγκρίνουμε τα δυο συστήματα ως προς τα ίχνη τους πάνω στο σύνολο των παρατηρήσιμων γεγονότων $\{a, b, c\}$, μπορούμε εύκολα να δούμε οτι τα δυο συστήματα έχουν ακριβώς το ίδιο σύνολο ιχνών.

Ετσι λοιπόν, δοσμένου ενός πεπερασμένου συνόλου $\mathcal{E}$ από γεγονότα, ένα "ίχνος" ορίζεται ως $e = e_0e_1e_2\ldots$, όπου $e_i \in 2^{\mathcal{E}} - \emptyset$. Με άλλα λόγια, ένα ίχνος είναι μια άπειρη λέξη πάνω στο σύνολο των μη κενών υποσυνόλων του $\mathcal{E}$.

Μια "διεργασία" ορίζεται σαν το ζεύγος $(\mathcal{E}, \mathcal{T})$, όπου $\mathcal{E}$ είναι ένα σύνολο από γεγονότα και $\mathcal{T}$ είναι το σύνολο όλων των πιθανών ιχνών της. Ενα σύστημα μπορεί να περιγραφεί σαν μια διεργασία. Το σύνολο γεγονότων της διεργασίας αναπαριστά τις πιθανές ενέργειες του συστήματος και το σύνολο των ιχνών της αντιστοιχεί στις πεπερασμένες ή άπειρες συμπεριφορές του συστήματος. Διεργασίες που περιγράφουν απλά συστήματα μπορούν να συνδυαστούν για να περιγράψουν πιο πολύπλοκα.

Παράδειγμα 2.1 Σ' ένα φανάρι κυκλοφορίας το χρώμα του φαναριού μπορεί να είναι είτε κόκκινο είτε πράσινο. Η συμπεριφορά αυτού του συστήματος μπορεί να περιγραφεί από το ίχνος:

$$e_{TL} : \{\text{κόκκινο}\} \rightarrow \{\text{πράσινο}\} \rightarrow \{\text{κόκκινο}\} \rightarrow \{\text{πράσινο}\} \ldots$$

το οποίο είναι και το μοναδικό ίχνος του παραπάνω συστήματος. Η διεργασία η οποία περιγράφει το σύστημα του φαναριού κυκλοφορίας ορίζεται ως $P = (\{\text{κόκκινο}, \text{πράσινο}\}, (\text{κόκκινο πράσινο})^{\omega})$, όπου ω δηλώνει άπειρη επανάληψη.

Ενα σύστημα μπορεί να περιέχει περισσότερες από μια συνιστώσες, που η καθεμιά μπορεί να περιγραφεί χωριστά σαν μια διεργασία, και έτσι, ολόκληρο το σύστημα περιγράφεται βάζοντας όλες τις συνιστώσες να "τρέχουν" παράλληλα. Για να περιγράψουμε ακριβώς την συμπεριφορά, "βάζοντας όλες τις συνιστώσες να "τρέχουν" παράλληλα", ορίζουμε ένα λογισμό, έτσι ώστε αν οι τελεστές του λογισμού εφαρμοσθούν πάνω σε διεργασίες, το αποτέλεσμα είναι και πάλι διεργασία. Ενας από τους πιο σημαντικούς τελεστές, της "παράλληλης σύνθεσης", ορίζεται παρακάτω. Εστω $P_i = (\mathcal{E}_i, \mathcal{T}_i)$, $i = 1 \dots n$ διεργασίες.
Η "παράλληλη σύνθεση" P, που ορίζεται σαν $\|_i P_i = (\mathcal{E}, \mathcal{T})$, είναι μια διεργασία, τέτοια ώστε $\mathcal{E} = \cup_i \mathcal{E}_i$ είναι το σύνολο γεγονότων της και $\mathcal{T} = \{e : e \perp \mathcal{E}_i \in \mathcal{T}_i\}$ το σύνολο ιχνών της, όπου ο τελεστής $\perp$, που ονομάζεαι "τελεστής προβολής", ορίζεται όπως παρακάτω:
Η "προβολή" ένος ίχνους e σε ένα σύνολο $\mathcal{E}' \subseteq \mathcal{E}$, $(e \perp \mathcal{E}')$, είναι ένα ίχνος e', που κατασκευάζεται από την τομή καθενός συνόλου γεγονότων του e με το $\mathcal{E}'$, και την αφαίρεση από την ακολουθία όλων των κενών συνόλων γεγονότων που προκύπτουν. Για παράδειγμα, ας θεωρήσουμε το σύνολο γεγονότων $\{a, b, c\}$ και το ίχνος

$$\{a, c\} \rightarrow \{c, d\} \rightarrow \{a, b, c\} \rightarrow \{a\} \dots$$

Η προβολή του παραπάνω ίχνους στο σύνολο $\{a, b\}$, είναι το ίχνος που ακολουθεί:

$$\{a\} \rightarrow \{a, b\} \rightarrow \{a\} \dots$$

2.1.3 ω-αυτόματα

Οπως έχουμε ήδη πει, μπορούμε να αντιλαμβανόμαστε τους υπολογισμούς ενός συστήματος σαν άπειρα ίχνη πάνω σε ένα σύνολο παρατηρήσιμων γεγονότων. Τα ίχνη μπορεί να θεωρηθεί οτι είναι οι άπειρες λέξεις που παράγονται από ένα πεπερασμένο αυτόματο. Τα αυτόματα αυτού του είδους ονομάζονται "ω-αυτόματα" και είναι φυσική επέκταση των γνωστών σε όλους μας πεπερασμένων αυτομάτων πάνω σε πεπερασμένες λέξεις. Οι γλώσσες τις οποίες αποδέχονται τα ω-αυτόματα ονομάζονται "ω-γλώσσες". Μια ω-γλώσσα λοιπόν αποτελείται από το σύνολο των άπειρων λέξεων πάνω σε ένα αλφάβητο Σ. Ανάμεσα στους διαφορετικούς τύπους ω-αυτομάτων, που μπορεί κανείς να βρει στη βιβλιογραφία, τα *Büchi* αυτόματα [Buc62] εκλέχτηκαν σαν μοντέλο αναφοράς, επειδή έχουν μεγάλη εκφραστικότητα και υπάρχουν πολύ ενδιαφέροντα αποτελέσματα γι' αυτά.

Ενα Büchi αυτόματο B ορίζεται σαν μια πεντάδα: $B = (\Sigma, S, S_0, E, F)$, όπου

- Σ είναι ένα αλφάβητο,
- S είναι ένα πεπερασμένο σύνολο από καταστάσεις,
- $S_0 \subseteq S$ είναι ένα σύνολο από αρχικές καταστάσεις,
- $E \subseteq S \times \Sigma \times S$ είναι ένα σύνολο από μεταβάσεις (ακμές),
- $F \subseteq S$ είναι ένα σύνολο από καταστάσεις αποδοχής.

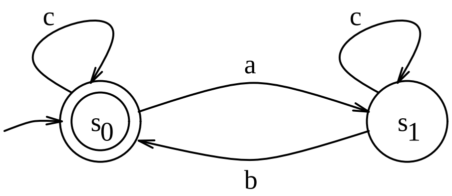
Το αυτόματο ξεκινά από μια από τις αρχικές του καταστάσεις και μπορεί να κινηθεί από μια κατάσταση s σε μια άλλη s' διαβάζοντας ένα σύμβολο εισόδου a αν $(s, a, s') \in E$.

Εστω $\sigma \in \Sigma^\omega$ μια άπειρη λέξη, $\sigma = \sigma_1\sigma_2\sigma_3\ldots$, ορίζουμε σαν ρ του B πάνω σε μια λέξη σ, την ακολουθία καταστάσεων $s_0 \xrightarrow{\sigma_1} s_1 \xrightarrow{\sigma_2} s_2 \xrightarrow{\sigma_3} \ldots$ τέτοια ώστε $s_0 \in S_0$ και για όλα τα $i \geq 0$, $(s_{i-1}, \sigma_i, s_i) \in E$.
Ενας "αποδεκτός δρόμος" είναι ένας δρόμος στον οποίο εμφανίζεται "άπειρα συχνά" ένα στοιχείο του συνόλου F.
Η γλώσσα $L(B)$ την οποία αποδέχεται το Büchi αυτόματο B, περιέχει όλες τις άπειρες λέξεις σ, τέτοιες ώστε το B έχει έναν αποδεκτό δρόμο πάνω στην σ.

Οπως ακριβώς στην περίπτωση των πεπερασμένων λέξεων, οι γλώσσες τις οποίες δέχονται τα πεπερασμένα αυτόματα ονομάζονται "κανονικές", έτσι και στην περίπτωση των άπειρων λέξεων, οι γλώσσες τις οποίες δέχονται τα Büchi αυτόματα ονομάζονται "ω-κανονικές γλώσσες". Πολύ ενδιαφέροντα αποτελέσματα ισχύουν για το σύνολο των ω-κανονικών γλωσσών:

- Η κλάση των ω-κανονικών γλωσσών είναι κλειστή κάτω από τις λογικές πράξεις της τομής, ένωσης και του συμπληρώματος.
- Μια ω-γλώσσα ονομάζεται ω-κανονική ανν υπάρχει κάποιο Büchi αυτόματο το οποίο την αναγνωρίζει.

Δοσμένων των παραπάνω δυο ιδιοτήτων, οι λογικές πράξεις της τομής, ένωσης και συμπληρώματος πάνω σε ω-κανονικές γλώσσες, μπορούν να υλοποιηθούν με την κατασκευή αυτομάτου, το οποίο να δέχεται την γλώσσα που προκύπτει από την τομή, ένωση ή συμπλήρωμα



Σχήμα 2.2: Büchi αυτόματο που αναγνωρίζει την γλώσσα $c^\omega + (c^* a c^* b)^\omega$

ω-κανονικών γλωσσών [Buc62]. Για την ακρίβεια υπάρχουν αλγόριθμοι, τέτοιοι ώστε, δοσμένων δυο Büchi αυτομάτων, που αποδέχονται δυο ω-κανονικές γλώσσες, κατασκευάζουν ένα Büchi που αποδέχεται την τομή ή την ένωση ή το συμπλήρωμα των παραπάνω δυο γλωσσών. Ετσι αντί να μιλάμε για λογικές πράξεις των ω-κανονικών γλωσσών, μπορούμε ισοδύναμα να μιλάμε για λογικές πράξεις των αντίστοιχων αυτομάτων που δέχονται αυτές τις γλώσσες.

Παράδειγμα 2.2 Ας θεωρήσουμε το αυτόματο της εικόνας 2.2. Περιγράφεται σαν (Σ, S, S_0, E, F), όπου $\Sigma = \{a, b, c\}$, $S = \{s_0, s_1\}$, $S_0 = \{s_0\}$, $E = \{(s_0, c, s_0), (s_0, a, s_1), (s_1, c, s_1), (s_1, b, s_0)\}$, $F = \{s_0\}$.

Η γλώσσα L του αυτομάτου είναι η ακόλουθη: $L = c^\omega + (c^* a c^* b)^\omega$.

Τα Büchi αυτόματα έχουν χρησιμοποιηθεί για να περιγράφουν συστήματα που αντιδρούν σε ερεθίσματα που λαμβάνουν από το περιβάλλον τους (*reactive systems*). Συνήθως τα συστήματα που μας ενδιαφέρει να περιγράψουμε είναι μια σύνθεση επιμέρους συνιστωσών. Εστω οτι κάθε επιμέρους συνιστώσα ονομάζεται C_i. Κάθε λοιπόν C_i περιγράφεται με ένα Büchi αυτόματο B_{C_i}. Ολόκληρο το σύστημα περιγράφεται με το αυτόματο B_S, το οποίο προκύπτει από την τομή των B_{C_i}.

2.2 Συστήματα πραγματικού χρόνου

Στην ενότητα 2.1 δώσαμε ένα γενικό σχήμα, τα συστήματα μεταβάσεων με ετικέττες, για την περιγραφή σύγχρονων συστημάτων τα οποία δεν περιέχουν χρονική πληροφορία. Σ' αυτήν την ενότητα θα προσπαθήσουμε να δώσουμε ένα γενικό σχήμα για την περιγραφή συστημάτων πραγματικού χρόνου.

Ενα σύστημα πραγματικού χρόνου μπορούμε να το δούμε σαν ένα σύνολο από χρονικές ακολουθίες καταστάσεων, κάθε μια από τις οποίες παριστάνει μια πιθανή εξέλιξη (συμπεριφορά) του συστήματος. Για να μπορέσουμε να περιγράψουμε ένα σύστημα πραγματικού χρόνου, χρειάζεται να προσθέσουμε στο μοντέλο μας την έννοια του "χρόνου". Σε οτι ακολουθεί, ο "χρόνος" θεωρείται σαν μια πλασματική ολική μεταβλητή του συστήματος t, η οποία παίρνει τιμές στο σύνολο των μη αρνητικών πραγματικών αριθμών R^+.

Ο "χρόνος" χρησιμοποιείται για να εκφράζει περιορισμούς για τις χρονικές στιγμές στις οποίες συμβαίνουν τα γεγονότα του συστήματος. Μιας και το πεδίο του χρόνου είναι πυκνό σύνολο, τα γεγονότα του συστήματος μπορούν να συμβούν αυθαίρετα, οσοδήποτε κοντά το ένα στο άλλο. Ο χρόνος ικανοποιεί τις παρακάτω δυο ιδιότητες:

- **συνέχειας**: ο χρόνος είναι μια συνεχής ποσότητα. Με άλλα λόγια: $\forall t, t' \in R^+, \exists t'' \in R^+$, such that $t \leq t'' \leq t'$,
- **προόδου**: $\forall t \in R^+$, υπάρχει πάντα κάποιο $t' \in R^+$ τέτοιο ώστε $t < t'$.

Ενα σύστημα πραγματικού χρόνου αποτελείται από τους παρακάτω συντελεστές: ένα σύνολο από καταστάσεις, ένα σύνολο παρατηρήσεων (observables), μια συνάρτηση χαρακτηρισμού (labeling function) και μια σχέση μετάβασης.

Οι καταστάσεις ενός συστήματος πραγματικού χρόνου περιγράφονται σαν ζεύγη (s, t), όπου s είναι μια θέση ελέγχου του συστήματος (control location), (κωδικοποιεί την ιστορία (εξέλιξη) του συστήματος), και t είναι η χρονική στιγμή στην οποία το σύστημα βρίσκεται σ' αυτήν την θέση ελέγχου.

Το σύνολο των παρατηρήσεων αντιστοιχεί είτε σε ένα σύνολο από γεγονότα του συστήματος, τα οποία αντιστοιχούν στις ενέργειες του συστήματος, είτε σε ένα σύνολο προτάσεων, οι οποίες μπορούν να ισχύουν στις καταστάσεις του συστήματος.

Η συνάρτηση χαρακτηρισμού αναθέτει σε κάθε κατάσταση του συστήματος ένα σύνολο είτε από γεγονότα, είτε από προτάσεις. Αν οι παρατηρήσεις είναι γεγονότα, τότε σε κάθε κατάσταση του συστήματος ανατείθεται ένα σύνολο γεγονότων που έχουν γίνει ή μπορούν να γίνουν στο σύστημα. Αν οι παρατηρήσεις είναι προτάσεις, τότε σε κάθε κατάσταση αντιστοιχεί το σύνολο προτάσεων που είναι αληθείς σ' αυτήν την κατάσταση.

Η σχέση μετάβασης ορίζει πώς το σύστημα αλλάζει κατάσταση, εξαρτώμενο βέβαια από την ενέργεια την οποία εκτελεί και από τους χρονικούς περιορισμούς τους οποίους πρέπει να ικανοποιεί.

Για την περιγραφή συστημάτων πραγματικού χρόνου έχουν προταθεί διάφορες εκδόσεις του γενικού σχήματος. Στις επόμενες ενότητες θα περιγράψουμε δυο από αυτές. Για κάθε μια από αυτές θα ορίσουμε μαθηματικά το υποκείμενο μοντέλο των συμπεριφορών του συστήματος, το συντακτικό και την σημασιολογία του μοντέλου και θα δώσουμε διευκρινηστικά παραδείγματα.

2.2.1 Συστήματα μεταβάσεων με ετικέττες που περιέχουν χρονική πληροφορία

Οπως αναφέραμε και παραπάνω, ένα σύστημα πραγματικού χρόνου μπορεί να θεωρηθεί σαν ένα σύνολο χρονικών ακολουθιών καταστάσεων. Κάθε κατάσταση του συστήματος ορίζεται σαν ένα ζευγάρι (s, t) όπου s είναι μια θέση ελέγχου του συστήματος και t είναι μια χρονική στιγμή. Μιας και ο χρόνος θεωρείται συνεχής ποσότητα, είναι προφανές οτι οι καταστάσεις ενός συστήματος πραγματικού χρόνου είναι άπειρες το πλήθος.

Για να μπορέσουμε λοιπόν να περιγράψουμε ένα σύστημα πραγματικού χρόνου με πεπερασμένο τρόπο, εισάγουμε την έννοια των συστημάτων μεταβάσεων με ετικέττες που περιέχουν χρονική πληροφορία. Στην ενότητα 2.1.1 παρουσιάσαμε τα συτήματα μεταβάσεων με ετικέττες σαν μοντέλα συστημάτων που δεν περιέχουν χρονική πληροφορία. Σ' ετούτη την ενότητα εισάγουμε λοιπόν την έννοια του χρόνου στα συστήματα μεταβάσεων με ετικέττες. Ο "χρόνος" θεωρείται σαν μια πλασματική ολική μεταβλητή του συστήματος που παίρνει τις τιμές της στο σύνολο των μη αρνητικών πραγματικών αριθμών R^+. Ενα σύστημα μπορεί να εξελίσσεται είτε με το "να εκτελεί μια ενέργεια" είτε με το "να αφήνει το χρόνο να περνά".

Στην πραγματικότητα η εκτέλεση ενός συστήματος αποτελείται από βήματα δυο φάσεων. Στην πρώτη φάση ο χρόνος αυξάνει κατά μια είτε πεπερασμένη είτε άπειρη ποσότητα. Αυτή η ποσότητα χρόνου είναι η ίδια για όλες τις συνιστώσες του συστήματος.

Στη δεύτερη φάση, οι συνιστώσες μπορεί να εκτελούν, είτε ανεξάρτητα είτε σε συνεργασία, μια πεπερασμένη αλλά αυθαίρετη ακολουθία από στοιχειώδης ενέργειες.

Ετσι λοιπόν, ένα σύστημα μεταβάσεων με ετικέττες που περιέχει χρονική πληροφορία μπορεί να παρασταθεί όπως ακολουθεί: Εστω $B = (\Sigma, S, \rightarrow, s_0)$ ένα σύστημα μεταβάσεων με ετικέττες όπως το ορίσαμε στην ενότητα 2.1.1. Η σχέση μετάβασης επεκτείνεται για να περιέχει και "χρονικές μεταβάσεις". Το σύνολο Σ ορίζεται σαν $Act \cup R^+$, όπου Act είναι το σύνολο των ενεργειών του συστήματος. Οι καταστάσεις του B είναι της μορφής (s, ν) όπου s είναι μια θέση ελέγχου και ν αντιστοιχεί σε ένα σύνολο από χρονικές στιγμές στις οποίες το σύστημα μπορεί να βρίσκεται στη θέση ελέγχου s. Οι χρονικές μεταβάσεις είναι της μορφής $s \xrightarrow{t} s'$, όπου s, s' είναι καταστάσεις του συστήματος μεταβάσεων B και $t \in R^+$. Η εκτέλεση ενός συστήματος πραγματικού χρόνου δεν μπορεί να παρασταθεί με μια διακριτή ακολουθία καταστάσεων όπως στην περίπτωση συστημάτων χωρίς χρονική πληροφορία, γιατί όπως αναφέρθηκε και πιο πριν ο αριθμός των καταστάσεων είναι άπειρος. Σαν αποτέλεσμα, μια χρονική μετάβαση της μορφής $s \xrightarrow{t} s'$ περιγράφει το πέρασμα του συστήματος απ' όλες τις καταστάσεις της μορφής $s + t'$, όπουτ $t' \leq t$. Οι χρονικές μεταβάσεις πρέπει να ακολουθούν τους παρακάτω κανόνες:

- **αιτιοκρατίας**: Το σύστημα εξελίσσεται με αιτιοκρατικό τρόπο. Δηλαδή,

$$\forall s, s', s'' \in S, \forall t \in R^+, s \xrightarrow{t} s' \wedge s \xrightarrow{t} s'' \Rightarrow s' = s''$$

- **προσθετικότητας**: Οι χρονικές μεταβάσεις σέβονται την συνέχεια του χρόνου.

$$\forall s, s' \in S, \forall t, t' \in R^+, \exists s'' \in S, s \xrightarrow{t} s'' \wedge s'' \xrightarrow{t'} s' \Rightarrow s \xrightarrow{t+t'} s'$$

Ενα "σύστημα μεταβάσεων με ετικέττες που περιέχει χρονική πληροφορία" είναι ένα σύστημα μεταβάσεων με ετικέττες B που ικανοποιεί τις ιδιότητες της αιτιοκρατίας και της προσθετικότητας για τις χρονικές μεταβάσεις.

2.3 Προσθέτοντας χρονική πληροφορία σε ένα σύστημα χωρίς χρόνο

2.3.1 Ιχνη και διεργασίες που περιέχουν χρονική πληροφορία

Στην ενότητα 2.1.2 παρουσιάσαμε οτι ένα σύστημα χωρίς χρονική πληροφορία μπορεί να περιγραφεί από το σύνολο όλων των ιχνών του. Σ'αυτήν εδώ την ενότητα θα εισάγουμε την έννοια του χρόνου στην σημασία ενός ίχνους. Οπως έχουμε ήδη αναφέρει στην ενότητα 2.2, ο χρόνος θεωρείτε σαν μια μεταβλητή t που παίρνει τις τιμές της στο σύνολο των μη αρνητικών πραγματικών αριθμών, R^+. Μπορούμε, με κάθε γεγονός, να συσχετίσουμε έναν πραγματικό αριθμό που είναι ο χρόνος εκτέλεσης αυτού του γεγονότος.

Ορίζουμε μια "χρονική ακολουθία" $\tau = \tau_0\tau_1\tau_2\ldots$ να είναι μια άπειρη ακολουθία από μη αρνητικούς πραγματικούς αριθμούς, $\tau_i \in R^+$.
Κάθε χρονική ακολουθία πρέπει να ικανοποιεί τους παρακάτω κανόνες:

- **μονοτονίας:** Η χρονική ακολουθία είναι αύξουσα,

- **προόδου:** $\forall t \in R^+$, υπάρχει πάντα $i \geq 0$ τέτοιο ώστε $t < \tau_i$.

Μπορούμε να προσθέσουμε χρονική πληροφορία σε ένα ίχνος με το να του αντιστοιχίσουμε μια χρονική ακολουθία.

Ετσι ένα "χρονικό ίχνος" πάνω σε ένα πεπερασμένο σύνολο γεγονότων $\mathcal{E}$ είναι ένα ζεύγος (e, τ), τέτοιο ώστε e είναι ένα ίχνος από σύνολα γεγονότων του $\mathcal{E}$ και τ είναι μια χρονική ακολουθία. Εχουμε λοιπόν το παρακάτω σχήμα:

$$\begin{array}{ccccccc} e & = & e_0 & \rightarrow & e_1 & \rightarrow & e_2 \ldots \\ & & \downarrow & & \downarrow & & \downarrow \\ \tau & = & \tau_0 & \rightarrow & \tau_1 & \rightarrow & \tau_2 \ldots \end{array}$$

το οποίο ερμηνεύεται διαισθητικά ως εξής: Σε κάθε σύνολο γεγονότων του ίχνους αντιστοιχούμε έναν πραγματικό αριθμό, ο οποίος δηλώνει την χρονική στιγμή που συμβαίνει το συγκεκριμένο σύνολο γεγονότων. Προσέξτε οτι το τ_i δηλώνει τον πραγματικό χρόνο στον οποίο συμβαίνουν τα γεγονότα του e_i, ενώ ο δείκτης i δηλώνει τον λογικό χρόνο εκτέλεσης του συνόλου e_i. Για παράδειγμα, ας θεωρήσουμε οτι το e_3, που είναι το τρίτο στοιχείο μιας ακολουθίας γεγονότων συμβαίνει στο χρόνο $\tau_3 = 4.7$. Προσέξτε οτι από τον κανόνα προόδου συνεπάγεται οτι μόνο πεπερασμένο πλήθος από γεγονότα μπορούν να συμβούν σε ένα φραγμένο διάστημα χρόνου και ακόμα, οτι δεν υπάρχει άνω φράγμα για το χρόνο.

Ορίζουμε επίσης μια "χρονική διεργασία" P_T να είναι ένα ζεύγος $(\mathcal{E}, \mathcal{T}_T)$, τέτοιο ώστε $\mathcal{E}$ είναι ένα πεπερασμένο σύνολο γεγονότων και $\mathcal{T}_T$ είναι το σύνολο των άπειρων χρονικών ιχνών της διεργασίας.

Η επέκταση των ορισμών των πράξεων προβολής και παράλληλης σύνθεσης για την περίπτωση των χρονικών διεργασιών είναι τετριμμένη.

Παράδειγμα 2.3 Ας θεωρήσουμε και πάλι το φανάρι κυκλοφορίας που αναφέραμε στο παράδειγμα 2.1. Πρώτα απ' όλα, εκτός από κόκκινο και πράσινο, το φανάρι ανάβει και πορτοκαλί. Εστω λοιπόν οτι η διάρκεια του κόκκινου είναι 2 λεπτά, του πράσινου 1 λεπτό και του πορτοκαλί 3 δευτερόλεπτα. Μας ενδιαφέρουν οι χρονικές στιγμές όπου το φανάρι αλλάζει χρώμα. Το ακόλουθο ίχνος περιγράφει την συμπεριφορά του φαναριού.

$$\begin{aligned}(e_{TL}, \tau) = (\{\text{κόκκινο}\}, 0) \rightarrow (\{\text{πράσινο}\}, 2) \rightarrow (\{\text{πορτοκαλί}\}, 3) \rightarrow (\{\text{κόκκινο}\}, 3.3) \rightarrow \\ (\{\text{πράσινο}\}, 5.3) \rightarrow (\{\text{πορτοκαλί}\}, 6.3) \rightarrow (\{\text{κόκκινο}\}, 6.6) \ldots\end{aligned}$$

Η χρονική διεργασία που περιγράφει την συμπεριφορά του συστήματος περιγράφεται όπως παρακάτω:

$P_T = [\{\text{κόκκινο}, \text{πράσινο}, \text{πορτοκαλί}\}, \{(e_{TL}, \tau) \mid \forall i(\tau_{2i+1} - \tau_{2i} = 2) \wedge (\tau_{2i+2} - \tau_{2i+1} = 1) \wedge (\tau_{2i+3} - \tau_{2i+2} = 0.3)\}]$.

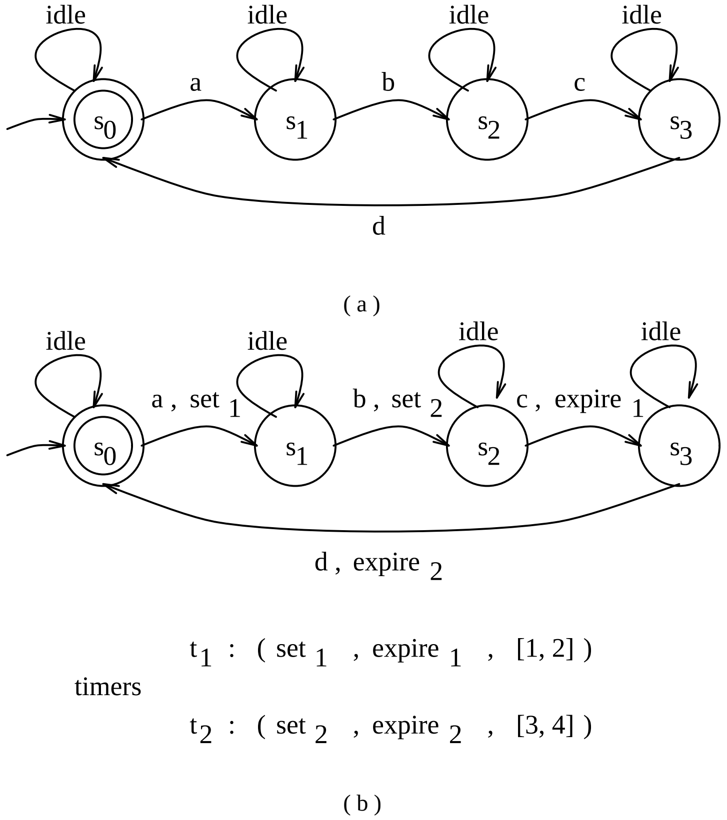
2.3.2 Set/Expire χρονικοί γράφοι

Η βασική ιδέα αυτού του μοντέλου είναι οτι ένα σύστημα πραγματικού χρόνου αποτελείται από δύο μέρη. Το "χρονικά ανεξάρτητο" μέρος και το "χρονικά εξαρτημένο". Το χρονικά ανεξάρτητο μπορούμε να το μοντελοποιήσουμε με ένα Büchi αυτόματο που περιγράφει μόνο τις λογικές ακολουθίες από γεγονότα του συστήματος και το χρονικά εξαρτημένο περιορίζει τις παραπάνω, μόνο σε εκείνες που είναι συνεπείς με ένα σύνολο χρονικών περιορισμών.

Σύμφωνα με αυτό το μοντέλο, ένα μοντέλο ενός συστήματος, χωρίς χρονική πληροφορία, επεκτείνεται για να αποκτήσει πληροφορία πραγματικού χρόνου, με ένα πεπερασμένο σύνολο από "μετρητές". Με άλλα λόγια, ένα σύστημα πραγματικού χρόνου μπορεί να περιγραφεί σαν ένα ζεύγος (B, T), όπου B είναι ένα Büchi αυτόματο και $T = \{t_1, t_2 \ldots t_n\}$ είναι ένα πεπερασμένο σύνολο από μετρητές που θέτουν χρονικούς περιορισμούς στις λογικές συμπεριφορές που περιγράφει το αυτόματο B. Οι "μετρητές" είναι πραγματικές μεταβλητές. Κάθε μετρητής σχετίζεται με δυο διαφορετικά γεγονότα μετρητών, ονομαστικά το γεγονός "θέσης" *(set)* και το γεγονός "εκπνοής" *(expire)*. Κάθε μετρητής έχει τα δικά του γεγονότα. Το αλφάβητο του B ορίζεται κατάλληλα ώστε να περιέχει και τα γεγονότα του συστήματος και τα γεγονότα των μετρητών.

Μαζί με κάθε μετάβαση του B, οι μετρητές μπορούν να "τεθούν" ή να "εκπνεύσουν". Οταν ένας μετρητής "τίθεται", παίρνει την τιμή του σε ένα φραγμένο διάστημα χρόνου και "μετράει" από πάνω προς τα κάτω μέχρι να εκπνεύσει το οποίο σημαίνει οτι η τιμή του είναι μηδέν. Οι μετρητές θέτουν περιορισμούς στις συμπεριφορές ενός συστήματος χωρίς χρόνο, της μορφής, "ο χρόνος που περνά ανάμεσα στη χρονική στιγμή που εκτελείται το γεγονός a και την χρονική στιγμή που εκτελείται το γεγονός b είναι μικρότερος από 2 λεπτά". Για να εκφράσουμε τον παραπάνω περιορισμό, ορίζουμε ένα μετρητή, του οποίου το γεγονός "θέσης" εκτελείται μαζί με το γεγονός του συστήματος a και το γεγονός "εκπνοής" μαζί με το γεγονός b. Η τιμή t στην οποία τίθεται ο μετρητής βρίσκεται στο διάστημα χρόνων $0 \leq t \leq 2$. Πριν προχωρήσουμε σε περισσότερες λεπτομέρειες, ας δούμε ένα παράδειγμα.

Παράδειγμα 2.4 Ας θεωρήσουμε το Büchi αυτόματο της εικόνας 2.3(a). Το σύστημα που περιγράφεται από αυτό το αυτόματο έχει μια πολύ απλή συμπεριφορά. Το σύνολο γεγονότων του είναι το $\{idle,\ a,\ b,\ c,\ d\}$ και η κατάσταση αποδοχής του είναι η s_0. Το σύστημα ξεκινά από την αρχική του κατάσταση που είναι η s_0. Μπορεί να μείνει για πάντα στην s_0 εκτελώντας συνεχώς *idle*. Αλλά όποτε αυτό το σύστημα εκτελεί a, τελικά θα εκτελέσει και d. Θέλουμε ακόμα το σύστημά μας να σέβεται τις παρακάτω συνθήκες:



Σχήμα 2.3: (a) Büchi αυτόματο που περιγράφει το σύστημα χωρίς χρονική πληροφορία. (b) Η επέκταση του Büchi αυτομάτου μαζί με ένα σύνολο από μετρητές, η οποία περιγράφει το σύστημα του παραδείγματος 2.4

"Ο χρόνος που περνά μεταξύ της εκτέλεσης του a και της εκτέλεσης του c είναι $1 \leq t \leq 2$".
"Ο χρόνος που περνά μεταξύ της εκτέλεσης του b και της εκτέλεσης του d είναι $3 \leq t \leq 4$".

Για να προσθέσουμε αυτήν την πληροφορία στο σύστημα, χρειαζόμαστε δυο μετρητές. Η αναπαράστασή τους φαίνεται στην εικόνα 2.3(b). Διαισθητικά, μπορούμε να φανταστούμε τους μετρητές σαν "ξυπνητήρια" που μπορούν να πάρουν μια οποιαδήποτε τιμή μέσα σε ένα φραγμένο διάστημα πραγματικών αριθμών. Αυτή η τιμή εκφράζει τον χρόνο τον οποίο πρέπει να μετρήσουν. Οι μετρητές τίθενται και αρχίζουν να μετρούν ανάποδα. Εκπνέουν δε, όταν ο χρόνος που είχαν να μετρήσουν έχει περάσει. Τα *set* και *expire* είναι γεγονότα. Κάθε μετρητής έχει τα δικά του *set* και *expire* γεγονότα, τα οποία συσχετίζονται κατάλληλα με τα υπόλοιπα γεγονότα του συστήματος. Η εμφάνιση του γεγονότος *set* ενός μετρητή δηλώνει την χρονική στιγμή της θέσης του μετρητή, ενώ η εμφάνιση του *expire* δηλώνει την στιγμή εκπνοής. Το Büchi αυτόματο της εικόνας 2.3(a) έχει επεκταθεί κατάλληλα, ώστε να περιλαμβάνει και τα γεγονότα μετρητών, όπως φαίνεται στην εικόνα 2.3(b).

Προσέξτε, στην εικόνα 2.3(b), οτι ο μετρητής t_1 τίθεται μαζί με το γεγονός του συστήματος a και εκπνέει μαζί με το c. Ομοια ο t_2 τίθεται μαζί με το γεγονός του συστήματος b και εκπνέει μαζί με το d. Προσέξτε ακόμα, οτι το διάστημα χρόνου για τον μετρητή t_1 είναι $[1, 2]$ και για τον t_2 είναι $[3, 4]$. Η λειτουργία των μετρητών μπορεί να εξηγηθεί διαισθητικά όπως παρακάτω: Ας δούμε για παράδειγμα τον μετρητή t_1 στην εικόνα 2.3(b). Οταν ενεργοποιείται η μετάβαση από την κατάσταση s_0 στην κατάσταση s_1, εκτελούνται δυο γεγονότα, το a και το set_1. Το a είναι ένα γεγονός του συστήματος και το set_1 είναι το γεγονός θέσης του μετρητή t_1. Η εκτέλεση του set_1 δηλώνει οτι η τιμή του μετρητή γίνεται t_1, όπου $1 \leq t_1 \leq 2$. Οι μετρητές μετρούν ανάποδα. Σε κάθε χρονική στιγμή η τιμή του μετρητή είναι ίση με το υπόλοιπο του χρόνου που του μένει να μετρήσει. Οταν εκτελείται η μετάβαση από την κατάσταση s_2 στην s_3, δυο άλλα γεγονότα εκτελούνται, c και $expire_1$. Το $expire_1$ δηλώνει την εκπνοή του μετρητή t_1. Αυτή τη χρονική στιγμή η τιμή του μετρητή ισούται με μηδέν.

Σ'αυτό το σημείο μπορούμε να κάνουμε μια παρατήρηση: Σ' αυτό το μοντέλο, η χρονική πληροφορία προστίθεται στο σύστημα με το να καθορίζουμε τις χρονικές αποστάσεις μεταξύ των γεγονότων του συστήματος. Οι μετρητές, που καθ' ένας από αυτούς κωδικοποιεί και μια χρονική απόσταση μεταξύ δυο γεγονότων παίρνουν τις τιμές τους στο σύνολο των μη αρνητικών πραγματικών αριθμών. Παρ' όλα αυτά, δεν μπορούμε να ξέρουμε την τιμή ενός μετρητή εκτός και αν συμβαίνει ένα γεγονός μετρητή. Οταν συμβαίνει ένα set, ξέρουμε οτι η τιμή του μετρητή γίνεται μια αυθαίρετη τιμή μέσα σε ένα συγκεκριμένο διάστημα χρόνου. Οσο δεν βλέπουμε κάποιο γεγονός μετρητή, δεν είναι δυνατόν να γνωρίζουμε την ακριβή τιμή του μετρητή. Σε ένα $expire$ ξέρουμε οτι η τιμή του μετρητή έχει φτάσει στο μηδέν. Με άλλα λόγια, μπορούμε να πούμε οτι αντί να κρατάμε συνεχώς πληροφορία για τις τιμές των μετρητών, παίρνουμε μόνο στιγμιότυπα των τιμών τους όταν συμβαίνουν σημαντικά γεγονότα, δηλαδή όταν συμβαίνουν τα γεγονότα μετρητών.

Οι μετρητές

Ενας μετρητής ορίζεται σαν μια τριάδα της μορφής: $t_i = (set_i, expire_i, interval_i)$ όπου $set_i, expire_i$ είναι τα γεγονότα του και $interval_i$ είναι το διάστημα τιμών του. Το $interval_i$ μπορεί να είναι κλειστό, ανοιχτό ή ημίκλειστο (από αριστερά ή από δεξιά), αλλά με όρια ακεραίους αριθμούς. Ετσι, σωστοί ορισμοί για διαστήματα είναι για παράδειγμα $[2, 4]$ ή $[2, \infty)$ ή $[1, 2)$ ή $(1, 3]$ ή $(3, 5)$

αλλά όχι και το $[2.5, 5.8]$. Ας ονομάσουμε l_i το κάτω άκρο και u_i το πάνω άκρο του διαστήματος $interval_i$. Οταν ένας μετρητής τίθεται, παίρνει μια τιμή $l_i \prec t_i \prec u_i$ όπου $\prec$ είναι $\leq$ ή $<$ ανάλογα με το αν το διάστημα είναι ανοιχτό ή κλειστό αντίστοιχα. Εστω T ένα πεπερασμένο σύνολο από μετρητές, που παίρνουν τις τιμές τους στο σύνολο των μη αρνητικών πραγματικών αριθμών R^+. Ενας μετρητής θεωρείται "ενεργός" αν έχει τεθεί και δεν έχει ακόμα εκπνεύσει. Μιας και τα γεγονότα μετρητών μπορούν να εμφανιστούν στα ίχνη του συστήματος, το σύστημα παράγει ακολουθίες από γεγονότα μετρητών, καθώς εξελίσσεται. Τα γεγονότα μετρητών δεν μπορούν να εμφανιστούν αυθαίρετα σε μια ακολουθία.

Μια ακολουθία από γεγονότα μετρητών ονομάζεται "καλά ορισμένη" αν:

1. Οι ενεργοί μετρητές δεν τίθενται αν δεν εκπνέουν στο ίδιο σύνολο γεγονότων.

2. Οι μετρητές εκπνέουν μόνο όταν είναι ενεργοί.

3. Κάθε ενεργός μετρητής τελικά εκπνέει.

Εστω $\nu : T \rightarrow R^+$ μια συνάρτηση που ονομάζεται "αποτίμηση μετρητών". Μια "αποτίμηση μετρητών" αναθέτει σε κάθε μετρητή $t_i \in T$, μια τιμή μέσα στο σύνολο R^+. Το σύνολο V είναι το σύνολο $[T \rightarrow R^+]$ όλων των δυνατών αποτιμήσεων μετρητών. Για $t \in R^+$ και $\nu \in V$, $\nu - t$ δηλώνει την αποτίμηση $\nu'(t_i) = \nu(t_i) - t, \forall t_i \in T$.

Οπως αναφέραμε στην προηγούμενη ενότητα, οι τιμές των μετρητών είναι γνωστές μόνο όταν συμβαίνουν γεγονότα μετρητών. Εστω $A \subseteq T$ ένα σύνολο από ενεργούς μετρητές, ν μια αποτίμηση μετρητών και $\mathcal{B}$ ένα σύνολο από γεγονότα μετρητών. Επίσης, έστω $E = \{t_i \mid expire_i \in \mathcal{B}\}$ και $S = \{t_j \mid set_j \in \mathcal{B}\}$. $nexttv(\nu)$ δηλώνει την αποτίμηση ν' τέτοια ώστε:

$$\begin{array}{lll} \exists t \in R^+ : & \forall t_i \in E : & \nu(t_i) = t \\ & \wedge \ \ \forall t_k \in A - E : & \nu(t_k) - t > 0 \wedge \nu'(t_k) = \nu(t_k) - t \\ & \wedge \ \ \forall t_j \in S : & l_j \prec \nu'(t_j) \prec u_j, \prec \in \{\leq, <\} \end{array}$$

Το καινούριο σύνολο από ενεργούς μετρητές είναι $A' = (A - E) \cup S$.

Μια αποτίμηση μετρητών που υπολογίζεται με τον παραπάνω τρόπο μπορεί να είναι "κενή". Αυτό σημαίνει οτι δεν υπάρχει πραγματικός αριθμός t που να ικανοποιεί τους παραπάνω κανόνες. Μια κενή αποτίμηση, σημαίνει διαισθητικά, οτι το σύνολο $\mathcal{B}$ των γεγονότων μετρητών δεν μπορεί να συμβεί κάτω από τις τρέχουσες τιμές των μετρητών. Αυτό επιβάλλει κανόνες διάταξης μεταξύ των γεγονότων εκπνοής των διαφόρων μετρητών. Η ιδιότητα που εξασφαλίζεται με τους παραπάνω κανόνες είναι οι ακόλουθη:

“Για κάθε μετρητή $t_i \in T$ ο χρόνος που περνά μεταξύ της χρονικής στιγμής που συμβαίνει το set_i και αυτής που συμβαίνει το πρώτο $expire_i$, μετά το set_i, είναι $l_i \prec t_{expire_i} - t_{set_i} \prec u_i, \prec \in \{\leq, <\}$”.

Συντακτικό και σημασιολογία των Set/Expire χρονικών γράφων

Τυπικά, ένας *Set/Expire* χρονικός γράφος ορίζεται σαν ένα ζεύγος (B, T) όπου:

1. $B = (\Sigma, S, S_0, E, F)$ είναι ένα Büchi αυτόματο, όπου:

    - $\Sigma = Act \cup 2^{\{set_i, expire_i | \forall t_i \in T\}}$. Σ περιέχει όλα τα γεγονότα του συστήματος (Act) και όλα τα γεγονότα μετρητών,
    - S είναι ένα πεπερασμένο σύνολο από καταστάσεις,
    - $S_0 \subseteq S$ είναι ένα σύνολο από αρχικές καταστάσεις,
    - $E \subseteq S \times Act \times 2^{\{set_i, expire_i | \forall t_i \in T\}} \times S$ είναι ένα σύνολο από μεταβάσεις,
    - F είναι ένα σύνολο από καταστάσεις αποδοχής

2. $T = \{t_1, t_2 \ldots t_n\}$ είναι ένα πεπερασμένο σύνολο από μετρητές.

Τώρα θα ορίσουμε το σύνολο των συμπεριφορών ενός ζεύγους (B, T). Ας θυμηθούμε τον ορισμό των χρονικών ιχνών από την προηγούμενη ενότητα. Ας δηλώνουμε με $\tau^{set_i}(k)$ την χρονική στιγμή που συμβαίνει το e_j, με $set_i \in e_j$ και είναι η $k-$οστη εμφάνιση του set_i στο ίχνος. Παρόμοια, ορίζουμε και το $\tau^{expire_i}(k)$. Το σύνολο των συμπεριφορών του ζεύγους (B, T) είναι το σύνολο όλων των χρονικών ιχνών (e, τ) που ικανοποιούν τις ακόλουθες συνθήκες:

1. e είναι ένα ίχνος του Büchi αυτομάτου B,

2. τ είναι μια χρονική ακολουθία,

3. $\tau^{set_i}(k) < \propto$ συνεπάγεται $\tau^{expire_i}(k) < \propto$,

    Αυτό σημαίνει οτι αν ένας μετρητής τίθεται, πρέπει τελικά να εκπνέει.

4. $\tau^{set_i}(k) < \tau^{expire_i}(k) \leq \tau^{set_i}(k+1)$,

    Αυτό διαισθητικά σημαίνει οτι οι ενεργοί μετρητές δεν τίθενται, εκτός αν εκπνέουν στο ίδιο σύνολο γεγονότων.

5. $\tau^{expire_i}(k) - \tau^{set_i}(k) \in interval_i$

   Αυτό είναι ο χρονικός περιορισμός που θέτει ο μετρητής.

Τώρα που δώσαμε την σημασιολογία του (B, T), είναι προφανές πως να ορίσουμε την τομή δυο τέτοιων ζευγών. Η συμπεριφορές της τομής δυο τέτοιων ζευγών (B_1, T_1), (B_2, T_2) είναι ακριβώς αυτές του ζεύγους $(B_1 \cap B_2, T_1 \cup T_2)$.

Το μοντέλο

Τα ζεύγη (B, T) μπορούν να χρησιμοποιηθούν για να περιγράφουν συστήματα πραγματικού χρόνου. Οι καταστάσεις του συστήματος μπορούν να παρασταθούν με ένα ζεύγος (s, ν), όπου s είναι μια θέση ελέγχου του συστήματος και ν είναι μια συνάρτηση αποτίμησης των μετρητών. Οι θέσεις ελέγχου αντιστοιχούν στις καταστάσεις του αυτομάτου B.

Το σύστημα ξεκινά από την αρχική κατάσταση $(s_0, \nu_{undefined})$ όπου s_0 είναι μια αρχική κατάσταση του αυτομάτου B και όλοι οι μετρητές του συστήματος είναι ανενεργοί. Αυτό σημαίνει οτι η τιμή τους δεν είναι ορισμένη. Το σύστημα κινείται από μια κατάσταση $(s, \nu) \stackrel{a,\mathcal{B}}{\to} (s', \nu')$ εκτελώντας μια ενέργεια a και ένα σύνολο γεγονότων μετρητών $\mathcal{B}$, αν υπάρχει ακμή του αυτομάτου τέτοια ώστε $s \xrightarrow{a,\mathcal{B}} s'$. Η αποτίμηση ν' είναι η $nexttv(\nu)$ δεδομένου του συνόλου γεγονότων μετρητών $\mathcal{B}$.

Ετσι, η συμπεριφορά του συστήματος μπορεί να περιγραφεί με τον παρακάτω κανόνα μετάβασης:

$$\frac{s \xrightarrow{a,\mathcal{B}} s'}{(s, \nu) \xrightarrow{a,\mathcal{B}} (s', nexttv(\nu))}$$

2.4 Μοντελοποίηση συστημάτων πραγματικού χρόνου με χρονικούς γράφους

2.4.1 Ακολουθίες καταστάσεων

Οι συμπεριφορές ενός συστήματος πραγματικού χρόνου θεωρούνται ακολουθίες καταστάσεων που γεννιόνται από ένα σύστημα μεταβάσεων με ετικέττες με χρονική πληροφορία. Παρακάτω, θα παρουσιάσουμε τους απαραίτητους ορισμούς.

Ας θεωρήσουμε ένα σύστημα μεταβάσεων με ετικέττες $B = (\Sigma, S, \to, s_0)$ όπως έχει ήδη ορισθεί.

Μια "ακολουθία καταστάσεων" $\sigma = s_0 s_1 s_2 \ldots$ είναι μια άπειρη ακολουθία καταστάσεων $s_i \in S$, τέτοιες ώστε για όλα τα $i \geq 0$ είτε:

1. (s_i, s_{i+1}) είναι μια "μετάβαση" το οποίο σημαίνει οτι $s_i \xrightarrow{a} s_{i+1}$, όπου a είναι μια ενέργεια του συστήματος. Σ' αυτήν την περίπτωση λέμε οτι η s_{i+1} είναι μια "διάδοχη κατάσταση μετάβασης" της s_i. Ο χρόνος εκτέλεσης μιας τέτοιας μετάβασης είναι ίσος με μηδέν.

2. (s_i, s_{i+1}) είναι μια "χρονική μετάβαση" το οποίο σημαίνει οτι $s_i \xrightarrow{t} s_{i+1}$, όπου $t \in R^+$. Σ' αυτή την περίπτωση λέμε οτι s_{i+1} είναι μια "διάδοχη κατάσταση χρονικής μετάβασης" της s_i. Αν ο χρόνος, στον οποίο το σύστημα είναι στην κατάσταση s_i, είναι t_i, τότε ο χρόνος που το σύστημα είναι στην s_{i+1} είναι $t_i + t$. Σημειώνουμε την s_{i+1} σαν $s_i + t$.

Για $s, s' \in S$ και $t \in R^+$ ορίζουμε:

$$s \overset{t}{\triangleright} s' \text{ if } s \xrightarrow{t} s + t \wedge \exists a \in Act, s + t \xrightarrow{a} s'.$$

Μια "ακολουθία βήματος" ορίζεται όπως ακολουθεί:

$$\sigma_s = s_0 \overset{t_0}{\triangleright} s_1 \overset{t_1}{\triangleright} s_2 \ldots$$

Το σύνολο όλων των ακολουθιών βήματος ενός συστήματος μεταβάσεων με ετικέττες που περιέχει χρονική πληροφορία B σημειώνεται με Σ_B^{step}.

Για κάθε κατάσταση $s \in S$, $\Sigma_B^{step}(s) = \{s_0 \overset{t_0}{\triangleright} s_1 \overset{t_1}{\triangleright} s_2 \ldots \mid s_0 = s\}$ είναι το σύνολο των ακολουθιών βήματος που ξεκινούν από το s.

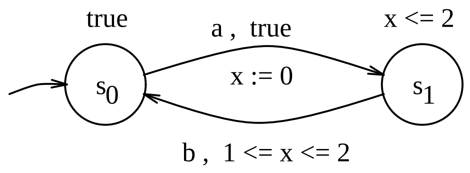
Δοσμένης μιας ακολουθίας βήματος $\sigma_s = s_0 \overset{t_0}{\triangleright} s_1 \overset{t_1}{\triangleright} s_2 \ldots$, για κάθε $i, j \in N$, $s_i, s_j \in \sigma_s$ ορίζουμε, $succ_t(s_i) = s_j$ ανν:

1. $i < j$, και

2. $i = j$ και $s_j = s_i + t$

2.4.2 Χρονικοί Γράφοι

Σ' αυτήν την ενότητα θα περιγράψουμε μια διαφορετική προσέγγιση πρόσθεσης χρονικής πληροφορίας σε πεπερασμένα αυτόματα. Οι "χρονικοί γράφοι" ή αλλιώς "χρονικά αυτόματα" [AD90, ACD90, Alu91], είναι ω-αυτόματα που έχουν επεκταθεί με ένα πεπερασμένο σύνολο από πραγματικές μεταβλητές που τις ονομάζουμε "ρολόγια". Οι τιμές των ρολογιών αυξάνουν ομοιόμορφα με το χρόνο. Ενα ρολόι μπορεί να τεθεί στο μηδέν μαζί με μια μετάβαση

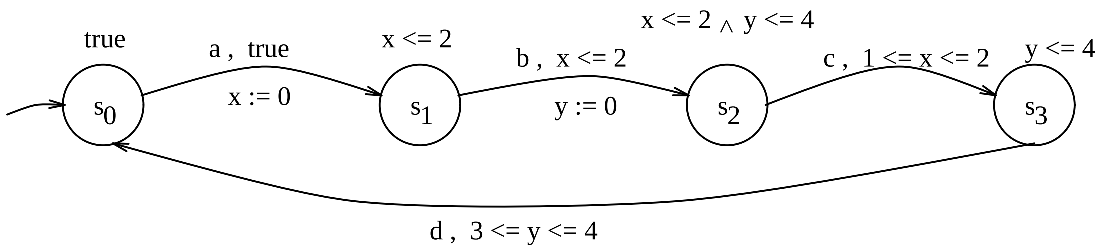


Σχήμα 2.4: Ενας χρονικός γράφος με ένα ρολόι

του αυτομάτου. Αυτή η πράξη ονομάζεται "λειτουργία επανάθεσης" ενός ρολογιού. Σε κάθε χρονική στιγμή, η τιμή ενός ρολογιού είναι ίση με το χρόνο που έχει περάσει από την τελευταία επανάθεσή του. Ενας "φύλακας" είναι ένας χρονικός περιορισμός που συσχετίζεται με κάθε μετάβαση του γράφου. Αυτός ο περιορισμός μπορεί να θεωρηθεί και σαν συνθήκη επίτρεψης. Με άλλα λόγια η μετάβαση μπορεί να πραγματοποιηθεί μόνο αν οι τρέχουσες τιμές των ρολογιών ικανοποιούν τον "φύλακα" της ακμής. Στην παραλλαγή χρονικών γράφων που θα εξηγήσουμε εδώ, αντιστοιχίζεται ένας περιορισμός και με κάθε κατάσταση του συστήματος που ονομάζεται "αναλλοίωτο". Αυτός ο περιορισμός έχει την έννοια οτι το σύστημα μπορεί να μείνει σε αυτήν την κατάσταση όσο ο περιορισμός ικανοποιείται από τις τιμές των ρολογιών. Τα "αναλλοίωτα" εκφράζουν ιδιότητες της μορφής: "Το σύστημα πρέπει τελικά να φύγει από αυτήν την κατάσταση". Στην πραγματικότητα, τα αναλλοίωτα εκφράζουν πιο ισχυρές ιδιότητες της μορφής: "Το σύστημα πρέπει να φύγει από αυτήν την κατάσταση μέσα σε ένα καθορισμένο διάστημα χρόνου". Ο συνδυασμός των περιορισμών που θέτουν τα αναλλοίωτα και αυτών που θέτουν οι φύλακες, ορίζουν το πώς το σύστημα εξελίσσεται μέσα στο χρόνο.

Τα αναλλοίωτα παίζουν ένα παρόμοιο ρόλο με τις συνθήκες αποδοχής που ορίσαμε στα Büchi αυτόματα. Αυτοί οι δυο τρόποι έκφρασης συνθηκών αποδοχής είναι παρόμοιοι αλλά όχι ισοδύναμοι. Πριν μπούμε σε περισσότερες λεπτομέρειες για τους χρονικούς γράφους, ας δούμε μερικά παραδείγματα.

Παράδειγμα 2.5 Σε αυτό το εισαγωγικό παράδειγμα, θα περιγράψουμε διαισθητικά την έννοια ενός χρονικού γράφου. Ας θεωρήσουμε τον χρονικό γράφο της εικόνας 2.4. Το σύστημα ξεκινά από την αρχική κατάσταση s_0 με το ρολόι x αρχικοποιημένο στην τιμή μηδέν. Στην αρχική κατάσταση το αναλλοίωτο είναι $true$ το οποίο σημαίνει οτι το σύστημα μπορεί να μείνει σ' αυτήν την κατάσταση για πάντα. Ακόμα μπορεί να διαλέξει να εκτελέσει την μετάβαση από την κατάσταση s_0 στην s_1. Η συνθήκη επίτρεψης αυτής της μετάβασης είναι επίσης $true$ το οποίο σημαίνει οτι το σύστημα μπορεί να αποφασίσει να την πάρει ή όχι, μη αιτιοκρατικά. Αν την πάρει, τότε εκτελεί a και επαναθέτει το ρολόι x στην τιμή μηδέν.



Σχήμα 2.5: Ενας χρονικός γράφος με δυο ρολόγια

Το αναλλοίωτο της κατάστασης s_1, αναγκάζει το σύστημα να φύγει από την κατάσταση s_1 σε χρόνο μικρότερο ή ίσο 2 χρονικών μονάδων από την στιγμή που μπήκε στην κατάσταση αυτή. Η μετάβαση από την s_1 πίσω στην s_0 επιτρέπεται όσο η τιμή του ρολογιού x είναι μικρότερη ή ίση 2 και μεγαλύτερη ή ίση του 1. Το σύστημα παίρνει αυτήν την μετάβαση εκτελώντας b.

Η τιμή του ρολογιού x στην s_1 είναι ο χρόνος που έχει περάσει από τη στιγμή που εκτελέσθηκε η μετάβαση από την s_0 στην s_1. Προσέξτε οτι όταν το σύστημα μπαίνει στην s_1, η τιμή του x ισούται με μηδέν. Για όλες τις τιμές του ρολογιού x, τέτοιες ώστε $0 \leq x < 1$ το σύστημα μπορεί να μένει στην s_1 χωρίς να κάνει τίποτα, απλώς να αφήνει το χρόνο να περνά. Για όλες τις τιμές του ρολογιού x τέτοιες ώστε $1 \leq x \leq 2$, το σύστημα μπορεί να διαλέξει να μείνει στην s_1 αφήνοντας το χρόνο να περνά ή να εκτελέσει την μετάβαση προς την s_0. Αλλά σίγουρα θα φύγει από την s_1 πριν η τιμή του ρολογιού ξεπεράσει την τιμή 2.

Παράδειγμα 2.6 Ας θεωρήσουμε το χρονικό γράφο της εικόνας 2.5. Το σύστημα ξεκινά από την κατάσταση s_0 με τα δυο ρολόγια του αρχικοποιημένα στην τιμή μηδέν. Το αναλλοίωτο της αρχικής κατάστασης είναι *true*, που σημαίνει οτι το σύστημα μπορεί να μείνει για πάντα εκεί. Μπορεί ακόμα να διαλέξει να εκτελέσει την μετάβαση από την s_0 στην s_1. Η συνθήκη για να εκτελέσει αυτή τη μετάβαση είναι επίσης *true* που σημαίνει οτι το σύστημα μπορεί να αποφασίσει μη αιτιοκρατικά. Εκτελώντας την παραπάνω μετάβαση, το σύστημα εκτελεί a και επαναθέτει το x στην τιμή μηδέν.

Το αναλλοίωτο της s_1, αναγκάζει το σύστημα να φύγει από αυτήν την κατάσταση σε χρόνο μικρότερο ή ίσο 2 χρονικών μονάδων από τη στιγμή που μπήκε στην s_1. Η μετάβαση από την s_1 στην s_2 επιτρέπεται όσο η τιμή του ρολογιού x είναι μικρότερη ή ίση του 2. Το σύστημα παίρνει αυτή την μετάβαση, εκτελώντας b και επαναθέτοντασς το y στο μηδέν.

Το αναλλοίωτο της s_2 δηλώνει οτι το σύστημα μπορεί να μείνει εκεί μόνο για το χρονικό διάστημα που $x \leq 2$ και $y \leq 4$. Η μετάβαση προςς την s_3 μπορεί να εκτελεσθεί όταν η τιμή του x είναι μεταξύ του 1 και του 2. Το σύστημα εκτελεί c και πάει στην s_3, όπου μπορεί να μείνει μέχρι η τιμή του y να φτάσει το 4. Η μετάβαση από την s_4 πίσω στην s_0 επιτρέπεται όταν $3 \leq y \leq 4$.

Προσέξτε οτι αυτός ο χρονικός γράφος περιγράφει το ίδιο σύστημα που περιγράψαμε στην εικόνα 2.3(b) χρησιμοποιώντας ένα Büchi αυτόματο και ένα σύνολο μετρητών.

Χρονικοί περιορισμοί

Και οι φύλακες και τα αναλλοίωτα ενός χρονικού γράφου είναι χρονικοί περιορισμοί που δεν διαφέρουν ούτε στο συντακτικό τους, ούτε στη σημασιολογία τους.

Παρακάτω θα δώσουμε τη γραμματική που γεννά τους παραπάνω χρονικούς περιορισμούς. Ας θεωρήσουμε C ένα πεπερασμένο σύνολο από ρολόγια που παίρνουν τις τιμές τους στο σύνολο R^+, δηλαδή το σύνολο των μη αρνητικών πραγματικών αριθμών. Το σύνολο Φ των χρονικών περιορισμών γεννιέται από την ακόλουθη γραμματική:

$$
\begin{array}{rcl}
\phi & ::= & true \\
 & | & c \leq m \\
 & | & c < m \\
 & | & c - c_1 \leq m \\
 & | & c - c_1 < m \\
 & | & \neg \phi \\
 & | & \phi_1 \wedge \phi_2
\end{array}
$$

όπου $c, c_1 \in C$ και $m \in N$. Οι περιορισμοί όπως $c = 1$, $c > 1$, $\phi_1 \vee \phi_2$ or $2 \leq c - c_1 \leq 3$ εκφράζονται σε συνάρτηση των παραπάνω.

Εστω $\nu : C \rightarrow R^+$ μια συνάρτηση που ονομάζεται . Μια αναθέτει σε κάθε ρολόι $c \in C$, μια τιμή του συνόλου R^+. Το σύνολο V είναι το σύνολο $[C \rightarrow R^+]$ όλων των αποτιμήσεων μετρητών. $R \subseteq C$, $\nu[R]$ δηλώνει την συνάρτηση αποτίμησης ν' τέτοια ώστε:

$$
\nu' = \begin{cases} 0 & for \quad c \in R \\ \nu(c) & otherwise \end{cases}
$$

Για $t \in R^+$ και $\nu \in V$, $\nu + t$ δηλώνει την αποτίμηση $\nu'(c) = \nu(c) + t, \forall c \in C$.

Εστω $\phi \in \Phi$ ένας χρονικός περιορισμός. $\phi(\nu)$ δηλώνει την τιμή του περιορισμού, αν όλα τα ρολόγια του ϕ παίρνουν τις τιμές τους μέσω της ν. Για παράδειγμα $\phi(\nu[R])$ είναι η τιμή του χρονικού περιορισμού όπου όλες οι εμφανίσεις του ρολογιού $c \in R$ έχουν αντικατασταθεί με μηδέν. Λέμε λοιπόν οτι μια αποτίμηση ν ικανοποιεί ένα περιορισμό ϕ αν $\phi(\nu)$ είναι αληθής.

Συντακτικό και σημασιολογία των χρονικών γράφων

Ο οριομός των χρονικών γράφων είναι μια επέκταση αυτού των ω-αυτομάτων.

Ενας χρονικός γράφος ορίζεται σαν μια εξάδα: $G = (\Sigma, S, s_0, C, E, f)$, όπου

- Σ είναι ένα πεπερασμένο σύνολο από γεγονότα (ένα αλφάβητο),
- S είναι ένα πεπερασμένο σύνολο καταστάσεων,
- s_0 είναι μια αρχική κατάσταση,
- C είναι ένα πεπερασμένο σύνολο από ρολόγια,
- $E \subseteq S \times \Sigma \times \Phi \times 2^C \times S$ είναι ένα σύνολο μεταβάσεων,
- $f : S \to \Phi$ είναι μια συνάρτηση που αντιστοιχεί ένα χρονικό περιορισμό (το αναλλοίωτο) σε κάθε κατάσταση του αυτομάτου.

Η έννοια της συνάρτησης f είναι οτι το σύστημα μπορεί να μένει σε μια δοσμένη κατάσταση s του γράφου μόνο όσο οι τιμές των ρολογιών ικανοποιούν την συνθήκη $f(s)$. Με άλλα λόγια $f(s)$ καθορίζει το αναλλοίωτο της κατάστασης s. Απαιτείται το παρακάτω:

$$\forall s \in S, \nu \in V, t \in R^+, \text{ if } f(s)(\nu + t) \text{ είναι αληθής then } f(s)(\nu) \text{ είναι επίσης αληθής.}$$

Η παραπάνω ιδιότητα σημαίνει διαισθητικά οτι αν το αναλλοίωτο μιας κατάστασης ικανοποιείται από τις τιμές των ρολογιών σε κάποια χρονική στιγμή στο μέλλον τότε πρέπει να ικανοποιείται και για όλες τις προηγούμενές της.

Οι χρονικοί γράφοι χρησιμοποιούνται για την περιγραφή συστημάτων πραγματικού χρόνου. Οι καταστάσεις του γράφου αντιστοιχούν σε θέσεις ελέγχου του συστήματος. Μια κατάσταση του συστήματος μπορεί να παρασταθεί με ένα ζευγάρι (s, ν) όπου s είναι μια θέση ελέγχου και ν είναι μια αποτίμηση των ρολογιών.

Το σύστημα ξεκινά από την αρχική κατάσταση (s_0, ν_0) όπου s_0 είναι η αρχική κατάσταση του χρονικού γράφου, με όλα τα ρολόγια μηδενισμένα. Το σύστημα κινείται από μια κατάσταση $(s, \nu) \xrightarrow{a} (s', \nu')$ εκτελώντας ένα γεγονός a, αν υπάρχει ακμή του γράφου τέτοια ώστε $s \xrightarrow{a,\phi,R} s'$. Οι κινήσεις του γράφου γίνονται στιγμιαία. Μια τέτοια κίνηση μπορεί να εκτελεσθεί μόνο αν οι τιμές των ρολογιών ικανοποιούν τον περιορισμό ϕ που είναι συσχετισμένος με αυτήν την ακμή του γράφου. Μαζί με μια τέτοια κίνηση όλα τα ρολόγια του συνόλου R μηδενίζονται. Λέμε λοιπόν οτι τα ρολόγια του R "επανατίθενται". Οι τιμές των ρολογιών αυξάνουν ομοιόμορφα και σε κάθε χρονική στιγμή, η τιμή του ρολογιού είναι ίση με τον χρόνο που έχει περάσει από την τελευταία επανάθεσή του. Ετσι ένα σύστημα πραγματικού χρόνου μπορεί να περιγραφεί με τους παρακάτω κανόνες μετάβασης:

$$\frac{s \xrightarrow{a,\phi,R} s' \wedge \phi(\nu)}{(s,\nu) \xrightarrow{a} (s',\nu[R])} \qquad \frac{f(s)(\nu+t)}{(s,\nu) \xrightarrow{t} (s,\nu+t)}$$

Κεφάλαιο 3

Γλώσσες περιγραφής ιδιοτήτων

Οπως έχουμε αναφέρει ήδη, η απόδειξη της ορθότητας συστημάτων, είναι ένας από τους βασικούς στόχους της αυτόματης επαλήθευσης. Χρειαζόμαστε λοιπόν ένα "τρόπο" να περιγράφουμε τις επιθυμητές ιδιότητες τις οποίες πρέπει να πληρεί το σύστημα. Αυτός ο "τρόπος" ονομάζεται γλώσσα περιγραφής ιδιοτήτων. Η γλώσσα περιγραφής ιδιοτήτων είναι το τρίτο στοιχείο του σχήματος που δώσαμε στην ενότητα 1.2. Στο κεφάλαιο 2 περιγράψαμε δυο διαφορετικά μοντέλα για την περιγραφή συστημάτων πραγματικού χρόνου. Σ' αυτό το κεφάλαιο παρουσιάζουμε δυο διαφορετικές γλώσσες περιγραφής ιδιοτήτων.

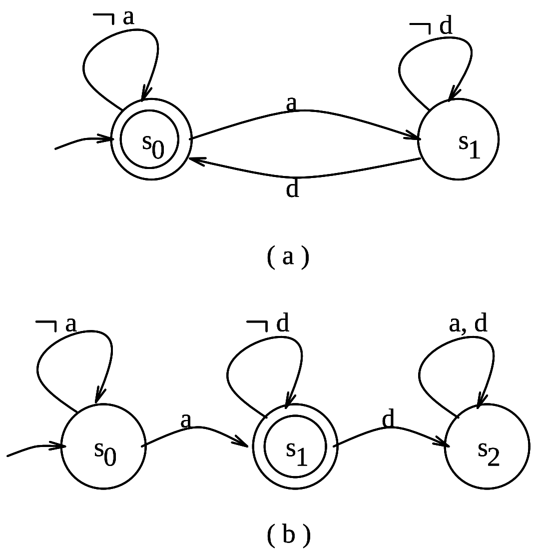
3.1 Περιγραφή ιδιοτήτων με αυτόματα

Σ' αυτή την ενότητα θα εξηγήσουμε πώς περιγράφουμε ιδιότητες χρησιμοποιώντας αυτόματα. Πρώτα θα εξηγήσουμε πώς εκφράζονται ιδιότητες που δεν περιέχουν χρονική πληροφορία και έπειτα θα επεκτείνουμε αυτή τη μεθοδολογία για την περίπτωση που οι ιδιότητες περιέχουν χρονική πληροφορία.

3.1.1 Περιγραφή χρονικά ανεξάρτητων ιδιοτήτων

Ας θυμηθούμε από την ενότητα 2.1.3 οτι ένα σύστημα μπορεί να περιγραφεί με ένα Büchi αυτόματο. Αυτό το αυτόματο ονομάζεται "υλοποίηση" του συστήματος.

Παρόμοια μια "περιγραφή" (ιδιότητα) του συστήματος μπορεί επίσης να εκφραστεί με ένα Büchi αυτόματο.



Σχήμα 3.1: (a)Büchi αυτόματο που δέχεται όλα τα ίχνη στα οποία όποτε συμβαίνει το a τελικά ακολουθείται από ένα d. (b)Büchi αυτόματο που δέχεται όλα εκείνα τα ίχνη στα οποία το a συμβαίνει πάντα και δεν ακολουθείται ποτέ από d.

Ενα σύστημα ικανοποιεί μια ιδιότητα ανν η γλώσσα που αναγνωρίζει η "υλοποίηση" του συστήματος είναι υποσύνολο της γλώσσας που αναγνωρίζει η "περιγραφή".

Παράδειγμα 3.1 Ας θεωρήσουμε το απλό σύστημα του παραδείγματος 2.4 που δίνεται στην εικόνα 2.3(a). Είναι φανερό οτι όλα τα αποδεκτά ίχνη που γεννιούνται από αυτό το αυτόματο ικανοποιούν την ιδιότητα: "Κάθε a τελικά ακολουθείται από ένα d".

Η ιδιότητα εκφράζεται σαν το Büchi της εικόνας 3.1(a). Αυτό το αυτόματο μένει στην κατάσταση s_0 η οποία είναι και κατάσταση αποδοχής. Μόλις συμβεί ένα a, το αυτόματο μεταβαίνει στην s_1 όπου περιμένει για το γεγονός d για να γυρίσει πίσω στην s_0.

Ας ονομάσουμε τις ιδιότητες που το σύστημα πρέπει να πληρεί, το "καθήκον" του συστήματος. Υπάρχουν περιπτώσεις που η κατασκευή του αυτομάτου του καθήκοντος του συστήματος είναι δύσκολη και καθόλου τετριμμένη υπόθεση. Σε τέτοιες περιπτώσεις είναι καμιά φορά πιο εύκολο να εκφράσουμε απ' ευθείας το "συμπλήρωμα του καθήκοντος" του συστήματος. Το "συμπλήρωμα του καθήκοντος" είναι η ιδιότητα που περιγράφει το σύνολο όλων των ανεπιθύμητων συμπεριφορών του συστήματος [ACW90]. Για παράδειγμα αντί να εκφράσουμε την ιδιότητα του παραδείγματος 3.1, εκφράζουμε το συμπλήρωμα της ιδιότητας: "Ενα a δεν ακολουθείται ποτέ από ένα d", και κατασκευάζουμε το αυτόματο που αντιστοιχεί σ' αυτήν και το οποίο φαίνεται στην εικόνα 3.1(b).

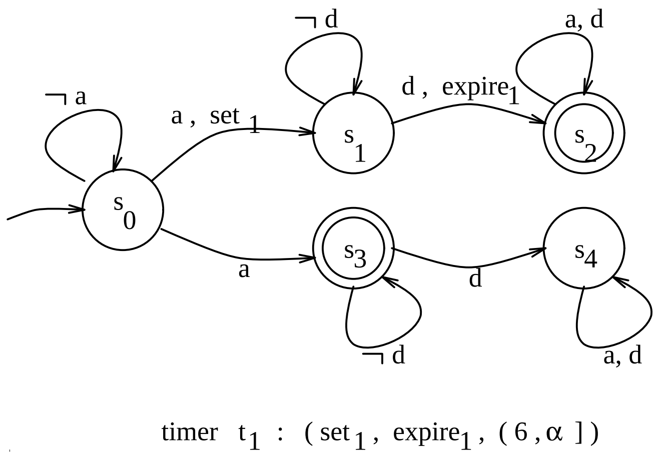
Σε αυτή την περίπτωση, αντί να ελέγχουμε αν η γλώσσα της υλοποίησης είναι υποσύνολο της γλώσσας της περιγραφής του συστήματος, ελέγχουμε αν η τομή της γλώσσας της υλοποίησης με την γλώσσα του συμπληρώματος του καθήκοντος ισούται με το κενό σύνολο. Ενα σύστημα μπορεί να πρέπει να πληρεί κάμποσες ιδιότητες. Το καθήκον του συστήματος αποτελείται από τη σύζευξη όλων αυτών των ιδιοτήτων. Αυτές οι ιδιότητες μπορούν όλες μαζί να εκφραστούν με ένα αυτόματο (που προφανώς είναι η τομή των επιμέρους αυτομάτων). Αλλά αυτός δεν είναι ο ιδανικός τρόπος μιας και το αυτόματο που προκύπτει μπορεί να είναι πάρα πολύ μεγάλο. Η τεχνική που ακολουθείται λοιπόν είναι η παρακάτω: Το συμπλήρωμα καθεμιάς ιδιότητας εκφράζεται αυτόνομα σαν ένα αυτόματο. Για να ελέγξουμε αν το σύστημα πληρεί το καθήκον του αρκεί να ελέγξουμε Χωριστά, αν πληρεί κάθε επιμέρους ιδιότητα. Αυτή η τεχνική έχει καλύτερη απόδοση μιας και ελαττώνει τον αριθμό των καταστάσεων που πρέπει να αναλυθούν.

3.1.2 Περιγραφή ιδιοτήτων πραγματικού χρόνου

Ας θυμηθούμε από την ενότητα 2.3.2 οτι ένα σύστημα πραγματικού χρόνου μπορεί να περιγραφεί με ένα Büchi αυτόματο B μαζί με ένα σύνολο από μετρητές T που κωδικοποιούν τους χρονικούς περιορισμούς που το σύστημα πρέπει να πληρεί. Σημειώναμε το σύστημα σαν (B, T). Το (B, T) δέχεται όλα τα ίχνη του συστήματος που είναι συμβατά με τους χρονικούς περιορισμούς. Οι ιδιότητες που μπορούν να εκφραστούν για ένα τέτοιο σύστημα μπορούν να είναι είτε χρονικά ανεξάρτητες, είτε χρονικά εξαρτημένες.

Οι χρονικά ανεξάρτητες ιδιότητες περιγράφουν μόνο τη λογική ακολουθία των γεγονότων (δηλαδή ποιό συμβαίνει πρώτο, ποιό δεύτερο κλπ.) αλλά όχι και τη χρονική στιγμή στην οποία συμβαίνουν. Για τις χρονικά ανεξάρτητες ιδιότητες η μεθοδολογία είναι ακριβώς η παραπάνω. Μιας και δεν μας ενδιαφέρει σε ποιές χρονικές στιγμές συμβαίνουν τα γεγονότα, κατασκευάζεται το Büchi αυτόματο για το συμπλήρωμα του καθήκοντος και ελέγχεται αν η τομή της γλώσσας του (B, T) με τη γλώσσα του συμπληρώματος του καθήκοντος, είναι το κενό σύνολο.

Οι χρονικά εξαρτημένες ιδιότητες περιέχουν και χρονική πληροφορία. Για παράδειγμα, ας θεωρήσουμε την ιδιότητα: "Ενα a ακολουθείται πάντα από ένα d σε χρόνο μικρότερο ή ίσο των 6 χρονικών μονάδων". Η ιδέα του συμπληρώματος της ιδιότητας εφαρμόζεται και σε αυτήν την περίπτωση. Το "χρονικά εξαρτημένο συμπλήρωμα καθήκοντος" είναι η ιδιότητα που περιγράφει το σύνολο όλων των ανεπιθύμητων χρονικών συμπεριφορών του συστήματος και εκφράζεται σαν (B', T'). Σ' αυτήν την περίπτωση λέμε οτι: Ενα σύστημα ικανοποιεί μια ιδιότητα ανν η τομή



Σχήμα 3.2: Büchi αυτόματο μαζί με ένα μετρητή που δέχεται όλα τα ίχνη στα οποία ένα a δεν ακολουθείται ποτέ από ένα d σε χρόνο μικρότερο ή ίσο των 6 χρονικών μονάδων μετά το a.

της γλώσσας του (B, T) με τη γλώσσα του (B', T'), του συμπληρώματος του καθήκοντος, είναι το κενό σύνολο.

Παράδειγμα 3.2 Ας θεωρήσουμε το σύστημα πραγματικού χρόνου του παραδείγματος 2.4 που περιγράφεται στην εικόνα 2.3(b). Ολα τα αποδεκτά χρονικά ίχνη που γεννιόνται από το παραπάνω σύστημα ικανοποιούν την ιδιότητα: "Το d συμβαίνει το πολύ 6 χρονικές μονάδες μετά το a". Το συμπλήρωμα της παραπάνω ιδιότητας μπορεί να εκφραστεί σαν το ζεύγος (B', T') που φαίνεται στην εικόνα 3.2.

Εδώ πρέπει να τονίσουμε κάτι: Υπάρχουν κάποιες χρονικές ιδιότητες που το συμπλήρωμά τους δεν μπορεί να εκφραστεί σαν ένα ζεύγος (B, T) [Dil89, AD90]. Γι' αυτές τις περιπτώσεις, η παραπάνω μέθοδος δεν μπορεί να εφαρμοσθεί. Παρ' όλα αυτά, στις περισσότερες περιπτώσεις το να εκφράσουμε το συμπλήρωμα μιας ιδιότητας είναι σχετικά απλή διαδικασία.

3.2 Χρονικές Λογικές

Η χρονική λογική αρχικά προτάθηκε από τον Pnueli [Pnu85] σαν ένας τρόπος περιγραφής συστημάτων που αντιδρούν σε ερεθίσματα που λαμβάνουν από το περιβάλλον τους (reactive systems). Είναι μια τροπική λογική με ένα σύνολο από στοιχειώδεις προτάσεις P και οι τελεστές της ερμηνεύονται ως εξής: Ο $\diamond$ σημαίνει οτι κάτι θα γίνει "τελικά", ο $\Box$ σημαίνει οτι κάτι ισχύει "πάντα" ή ο $\mathcal{U}$ που σημαίνει οτι κάτι ισχύει "μέχρι" να ισχύσει κάτι άλλο. Η χρονική λογική προσφέρει ένα φυσικό τρόπο να εκφράζουμε ιδιότητες των συστημάτων. Με άλλα λόγια χρησιμοποιείται κυρίως για γλώσσα περιγραφής ιδιοτήτων.

Οι χρονικές λογικές χωρίζονται σε δυο κατηγορίες: Σε "λογικές γραμμικού χρόνου" και σε "λογικές δενδρικού χρόνου". Πολλές παραλλαγές έχουν οριοθεί σε καθεμιά από τις παραπάνω κατηγορίες. Ενα χαρακτηριστικό παράδειγμα για τις λογικές γραμμικού χρόνου είναι η Προτασιακή Χρονική Λογική ΠΧΛ,(Propositional Temporal Logic, PTL) και για τις λογικές δενδρικού χρόνου είναι η Λογική Δενδρικού Υπολογισμού ΛΔΥ, (Computation Tree Logic, CTL) [EC82], [CES86]. Αυτές οι εναλλακτικές έχουν επεκταθεί κατάλληλα για να εισάγουν την έννοια του πραγματικού χρόνου στα μοντέλα των συστημάτων και έχουν προκύψει η Προτασιακή Χρονική Λογική με Χρόνο (Timed Propositional Temporal Logic, TPTL) και η Λογική Δενδρικού Υπολογισμού με Χρόνο (Timed Computation Tree Logic, TCTL).

Πώς όμως οι χρονικές λογικές χρησιμοποιούνται σαν γλώσσες περιγραφής ιδιοτήτων; Ας θυμηθούμε από την ενότητα 2.1.1 οτι ένα σύστημα μπορεί να περιγραφεί σαν ένα σύστημα μεταβάσεων με εττικέτες. Μια ιδιότητα εκφράζεται σαν μια "έκφραση χρονικής λογικής". Μια έκφραση f είναι μια συντακτική κατασκευή που γεννιέται από μια γραμματική χρησιμοποιώντας συνδυασμούς λογικών και χρονικών τελεστών. Για παράδειγμα η έκφραση $\forall\Box\exists\Diamond(p\vee q)$ σημαίνει οτι πάντα είναι δυνατόν να φτάσουμε σε μια κατάσταση του συστήματος όπου είτε ισχύει το p είτε ισχύει το q. Οι εκφράσεις χρονικής λογικής ερμηνεύονται πάνω σε "ακολουθίες καταστάσεων" που γεννιόνται από ένα σύστημα μεταβάσεων. Οι ακολουθίες καταστάσεων είναι το υποκείμενο μοντέλο για την περιγραφή των συμπεριφορών συστημάτων. Ετσι μια έκφραση χρονικής λογικής στην ουσία ορίζει ένα σύνολο από ακολουθίες καταστάσεων οι οποίες ικανοποιούν την κείμενη ιδιότητα.

3.2.1 TCTL: Περιγραφή ιδιοτήτων πραγματικού χρόνου

Σ' αυτήν την ενότητα θα παρουσιάσουμε μια γλώσσα περιγραφής ιδιοτήτων πραγματικού χρόνου. Η CTL (Λογική Δενδρικού Υπολογισμού) προτάθηκε αρχικά από τους Emerson και Clarke [EC82] σαν γλώσσα περιγραφής χρονικά ανεξάρτητων ιδιοτήτων. Εχει επεκταθεί στην TCTL (Λογική Δενδρικού Υπολογισμού με Χρόνο) [ACD90, Alu91] για την περιγραφή ιδιοτήτων πραγματικού χρόνου. Η επέκταση αυτή βασίζεται στο γεγονός οτι εισάγεται η έννοια του πραγματικού χρόνου στους τελεστές. Για παράδειγμα μπορούμε να γράψουμε $\forall\Diamond_{\leq 3}p$, το οποίο διαισθητικά σημαίνει οτι για όλους τους υπολογισμούς του συστήματος, η ιδιότητα p είναι αληθής σε χρόνο μικρότερο ή ίσο των 3 χρονικών μονάδων.

Οι εκφράσεις TCTL ερμηνεύονται πάνω σε "ακολουθίες καταστάσεων" που γεννιόνται από ένα χρονικό σύστημα μεταβάσεων με ετικέττες. Ας θυμηθούμε τους ορισμούς για τις ακολουθίες καταστάσεων από την ενότητα 2.2.1.

Συντακτικό και σημασιολογία

Εστω P ένα σύνολο από στοιχειώδεις προτάσεις. Οι εκφράσεις TCTL ορίζονται από την ακόλουθη γραμματική:

$$\begin{array}{rcl} \phi & ::= & p \\ & \mid & \neg\phi \\ & \mid & \phi_1 \vee \phi_2 \\ & \mid & \phi_1 \exists \mathcal{U}_{\prec n} \phi_2 \\ & \mid & \phi_1 \forall \mathcal{U}_{\prec n} \phi_2 \end{array}$$

όπου $p \in P$, $n \in N$ και $\prec$ είναι ένα από τα σύμβολα $<, \leq, >, \geq, =$.

Διαισθητικά, $\phi_1 \exists \mathcal{U}_{\prec n} \phi_2$ σημαίνει οτι υπάρχει μια συμπεριφορά του συστήματος τέτοια ώστε ϕ_1 ισχύει μέχρι μια κατάσταση σε χρόνο $t \prec n$ όπου ισχύει η ϕ_2.

$\phi_1 \forall \mathcal{U}_{\prec n} \phi_2$ σημαίνει οτι για όλες τις συμπεριφορές του συστήματος η ϕ_1 ισχύει μέχρι μια κατάσταση σε χρόνο $t \prec n$ όπου ισχύει η ϕ_2.

Αλλοι χρονικοί τελεστές εκφράζονται σε συνάρτηση με τους παραπάνω. Για παράδειγμα:

- Οι $\exists \mathcal{U}_{\geq 0}$, $\forall \mathcal{U}_{\geq 0}$ αντιστοιχούν στους μη φραγμένους τελεστές $\exists \mathcal{U}$ και $\forall \mathcal{U}$ αντίστοιχα.
- $true \exists \mathcal{U}_{\prec n} \phi$ αντιστοιχεί στην έκφραση $\exists \Diamond_{\prec n} \phi$.
- $true \forall \mathcal{U}_{\prec n} \phi$ αντιστοιχεί στην $\forall \Diamond_{\prec n} \phi$.
- $\neg \forall \Diamond_{\prec n} \neg \phi$ αντιστοιχεί στην $\exists \Box_{\prec n} \phi$
- $\neg \exists \Diamond_{\prec n} \neg \phi$ αντιστοιχεί στην $\forall \Box_{\prec n} \phi$

Αλλοι τελεστές όπως ο $\wedge$ ή $\Rightarrow$ εκφράζονται όπως φαίνεται παρακάτω:

- $\neg(\neg\phi_1 \vee \neg\phi_2)$ εκφράζεται ως $\phi_1 \wedge \phi_2$
- $\neg\phi_1 \vee \phi_2$ εκφράζεται ως $\phi_1 \Rightarrow \phi_2$

Οι εκφράσεις TCTL ερμηνεύονται πάνω στις καταστάσεις ενός συστήματος μεταβάσεων με ετικέττες. Ας θεωρήσουμε ένα σύστημα μεταβάσεων με ετικέττες $B = (\Sigma, S, \rightarrow, s_0)$, ένα σύνολο από ακολουθίες βήματος $\Sigma^{step} \subseteq \Sigma_B^{step}$, μια κατάσταση $s \in S$, και μια έκφραση TCTL ϕ. Ορίζουμε μια συνάρτηση $\mu : S \rightarrow 2^P$, η οποία αντιστοιχεί σε κάθε κατάσταση του συστήματος το

σύνολο των στοιχειωδών προτάσεων που είναι αληθείς σ' αυτήν την κατάσταση. Ορίζουμε την σχέση $s \models_{\Sigma^{step}} \phi$: $s \models_{\Sigma^{step}} p$ ανν $p \in \mu(s)$
$s \models_{\Sigma^{step}} \neg\phi$ ανν $s \not\models_{\Sigma^{step}} \phi$
$s \models_{\Sigma^{step}} \phi_1 \vee \phi_2$ ανν $s \models_{\Sigma^{step}} \phi_1$ or $s \models_{\Sigma^{step}} \phi_2$
$s \models_{\Sigma^{step}} \phi_1 \exists\mathcal{U}_{\prec n}\phi_2$ ανν $\exists\sigma_s \in \Sigma^{step}(s).\exists t \in R^+ : succ_t(s) \models_{\Sigma^{step}} \phi_2 \wedge \forall t', 0 \leq t' \leq t$ $succ_{t'}(s) \models_{\Sigma^{step}} \phi_1 \vee \phi_2$
$s \models_{\Sigma^{step}} \phi_1 \forall\mathcal{U}_{\prec n}\phi_2$ ανν $\forall\sigma_s \in \Sigma^{step}(s).\exists t \in R^+ : succ_t(s) \models_{\Sigma^{step}} \phi_2 \wedge \forall t', 0 \leq t' \leq t$ $succ_{t'}(s) \models_{\Sigma^{step}} \phi_1 \vee \phi_2$

Παράδειγμα 3.3 Ας θεωρήσουμε τον χρονικό γράφο της εικόνας 2.5. Αυτός παριστάνει ένα σύστημα πραγματικού χρόνου που πληρεί την ιδιότητα: "Οποτε συμβαίνει το a, τελικά συμβαίνει και το d σε χρόνο μικρότερο ή ίσο των 6 χρονικών μονάδων μετά το a". Αυτή η ιδιότητα μπορεί να εκφραστεί με την ακόλουθη έκφραση TCTL:

$$\forall\Box(a \Rightarrow \forall\Diamond_{\leq 6} d)$$

Η παραπάνω έκφραση δηλώνει οτι δοσμένης μιας κατάστασης s του συστήματός μας, για όλες τις ακολουθίες βήματος $\sigma_s \in \Sigma^{step}(s)$ και για όλες τις καταστάσεις s' μιας τέτοιας ακολουθίας είτε a είναι ψευδής στην s', είτε, αν a είναι αληθής στην s' τότε υπάρχει μια κατάσταση s'' τέτοια ώστε $s'' = succ_t(s')$, $t \leq 6$, στην οποία ισχύει το d.

Κεφάλαιο 4

Αλγόριθμοι Επαλήθευσης

Στα δυο προηγούμενα κεφάλαια μιλήσαμε για μεθόδους περιγραφής συστημάτων και ιδιοτήτων. Σε αυτό το κεφάλαιο θα παρουσιάσουμε δυο αλγόριθμους επαλήθευσης συστημάτων πραγματικού χρόνου.

Δοσμένου ενός συστήματος και μιας ιδιότητας, ο αλγόριθμος επαλήθευσης απαντά στο ερώτημα, αν το σύστημα ικανοποιεί την ιδιότητα.

Ο πρώτος αλγόριθμος ανάγει το πρόβλημα της επαλήθευσης στο πρόβλημα της έγκλεισης γλωσσών αυτομάτων. Ο δεύτερος, είναι ένας αλγόριθμος συμβολικού ελέγχου μοντέλου (symbolic model checking) για μια έκφραση της TCTL. Απαντά το πρόβλημα της επαλήθευσης με το να υπολογίζει το σύνολο των καταστάσεων ενός συστήματος μεταβάσεων που ικανοποιούν την έκφραση TCTL.

4.1 Εγκλειση γλωσσών αυτομάτων

Σύγχρονα συστήματα μπορούν να περιγραφούν και αναλυθούν με Büchi αυτόματα. Ας θυμηθούμε από την ενότητα 2.1.3 οτι ένα σύστημα μπορεί να περιγραφεί σαν ένα Büchi αυτόματο B. Ας θυμηθούμε ακόμα από την ενότητα 3.1.1 οτι μια ιδιότητα μπορεί επίσης να περιγραφεί σαν ένα Büchi αυτόματο B'. Οπως έχουμε ήδη αναφέρει, ένα σύστημα ικανοποιεί μια ιδιότητα ανν η γλώσσα του αυτομάτου που περιγράφει το σύστημα (υλοποίηση του συστήματος), είναι υποσύνολο της γλώσσας του αυτομάτου που περιγράφει την ιδιότητα (περιγραφή του συστήματος).

Κατά συνέπεια, μπορούμε να ελέγχουμε αν ένα σύστημα ικανοποιεί μια ιδιότητα με το να ελέγχουμε αν $L(B) \subseteq L(B')$. Ετσι το πρόβλημα της επαλήθευσης ανάγεται στο πρόβλημα έγκλεισης γλωσσών πεπερασμένων αυτομάτων.

Ελέγχουμε $L(B) \subseteq L(B')$ με το να ελέγχουμε αν $L(B \cap \overline{B'}) = \emptyset$, όπου $\overline{B'}$ είναι το αυτόματο που περιγράφει το συμπλήρωμα της ιδιότητας όπως έχουμε περιγράψει στην ενότητα 3.1.2. Το αν η γλώσσα ενός αυτομάτου είναι η κενή ελέγχεται χρησιμποποιώντας μια απλή γραφοθεωρητική συνθήκη:

Η γλώσσα ενός αυτομάτου είναι κενή ανν δεν υπάρχει μια ισχυρά συνεκτική συνιστώσα που να περιέχει μια κατάσταση αποδοχής προσβάσιμη από μια από τις αρχικές καταστάσεις.

4.1.1 Επαλήθευση συστημάτων πραγματικού χρόνου

Ας θεωρήσουμε οτι ένα σύστημα πραγματικού χρόνου περιγράφεται σαν (B, T) χρησιμοποιώντας το μοντέλο που περιγράψαμε στην ενότητα 2.3.2. Ας θυμηθούμε ακόμα οτι το συμπλήρωμα του καθήκοντος ενός συστήματος περιγράφεται και αυτό σαν ένα ζεύγος $(\overline{B'}, T_{\overline{B'}})$, ενότητα 3.1.2. Το σύστημα που προκύπτει από την τομή των παραπάνω ζευγών είναι το $(B \cap \overline{B'}, T \cup T_{\overline{B'}})$.

Ο παραπάνω αλγόριθμος μπορεί να εφαρμοσθεί άμεσα για την επαλήθευση συστημάτων πραγματικού χρόνου τα οποία περιγράφονται σύμφωνα με το παραπάνω μοντέλο. Η ιδέα είναι να κατασκευάσουμε ένα αυτόματο που να δέχεται τις ίδιες συμπεριφορές με εκείνες του $(B \cap \overline{B'}, T \cup T_{\overline{B'}})$ και να ελέγξουμε την κενότητα της γλώσσας αυτού του αυτομάτου.

Μια τέτοια μέθοδος περιγράφεται στο [Dil89] και θα την εξηγήσουμε εδώ συνοπτικά.

Αναπαράσταση των χρονικών περιορισμών

Σ' αυτή την ενότητα, θα παρουσιάσουμε συνοπτικά μια μέθοδο αναπαράστασης των χρονικών περιορισμών. Η κεντρική ιδέα της μεθόδου είναι οτι οι χρονικοί περιορισμοί μπορούν να εκφραστούν πλήρως με γραμμικές ανισότητες πάνω στις τιμές μεμονωμένων μετρητών και στις διαφορές τους ανά δύο.

Ας θεωρήσουμε για παράδειγμα τον περιορισμό:

$$1 \leq t_1 \leq 2 \wedge 3 \leq t_2 \leq 4.$$

Το σύνολο των τιμών μετρητών που ικανοποιεί τον παραπάνω περιορισμό ορίζεται ακριβώς με το παραπάνω σύνολο γραμμικών ανισοτήτων:

$$\begin{array}{rcccl} & & t_1 - t_2 & \leq & -1 \\ & & t_2 - t_1 & \leq & 3 \\ 1 & \leq & t_1 & \leq & 2 \\ 3 & \leq & t_2 & \leq & 4 \end{array}$$

Αυτές οι ανισότητες μπορούν να περιγραφούν από τον ακόλουθο πίνακα:

$$\begin{array}{cccc} & t_0 & t_1 & t_2 \\ t_0 & (0, \leq) & (-1, \leq) & (-3, \leq) \\ t_1 & (2, \leq) & (0, \leq) & (-1, \leq) \\ t_2 & (4, \leq) & (3, \leq) & (0, \leq) \end{array}$$

Το (i, j) – στό στοιχείο του πίνακα δίνει ένα άνω φράγμα d_{ij} της διαφοράς μεταξύ των τιμών των μετρητών t_i, t_j. Ο μετρητής t_0 είναι ένας πρόσθετος μετρητής με την τιμή του να ισούται πάντα με μηδέν. Χρησιμοποιείται για να περιγράφουμε άνω φράγματα των τιμών μεμονωμένων μετρητών. Αυτοί οι πίνακες ονομάζονται "Πίνακες Φραγμάτων Διαφορών", ΠΦΔ (*Difference Bounds Matrices*).

Τα στοιχεία ενός ΠΦΔ είναι "φράγματα". Τα "φράγματα" ορίζονται να είναι διατεταγμένα ζεύγη που παίρνουν τις τιμές τους από το πεδίο $\mathcal{B}ounds$: $Z \times (\{<, \leq\} \cup \{(\infty, <), (-\infty, <)\})$, όπου Z είναι το σύνολο των ακεραίων. Τα σύμβολα $<, \leq$ είναι ολικά διατεταγμένα, $<$ είναι αυστηρά μικρότερο του $\leq$. Μια μερική διάταξη ορίζεται σαν: $(z, b) \preceq (z', b')$ αν $(z < z')$ ή αν $(z = z'$ και $b \leq b')$ (λεξικογραφική διάταξη).

Ετσι, για παράδειγμα, ο περιορισμός $1 \leq t_1 \leq 2$ μπορεί να παρασταθεί με το σύνολο των ανισοτήτων:

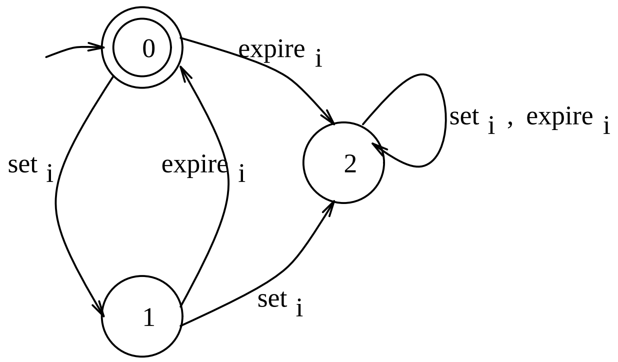
$$\begin{aligned} t_1 - t_0 &\leq 2 \\ t_0 - t_1 &\leq -1 \end{aligned}$$

Η (t_0, t_1) είσοδος του πίνακα είναι το φράγμα $(-1, \leq)$ και η είσοδος (t_1, t_0) θα είναι $(2, \leq)$.

Το κύριο πρόβλημα αυτής της αναπαράστασης είναι οτι δεν είναι μοναδική για δοσμένο σύνολο περιορισμών. Ας δούμε για παράδειγμα τα σύνολα των ανισοτήτων:

$$\begin{array}{ll} t_1 - t_2 < 1 & t_1 - t_2 < 1 \\ t_2 - t_3 < 1 & t_2 - t_3 < 1 \\ t_1 - t_3 < 100 & t_1 - t_3 < 2 \end{array}$$

Είναι προφανές οτι αυτά τα δυο σύνολα περιγράφουν το ίδιο σύνολο περιορισμών. Αυτό συμβαίνει γιατί οι δυο πρώτες ανισότητες συνεπάγονται την $t_1 - t_3 < 2$. Ετσι αν η ανισότητα $t_1 - t_3 < 2$ αντικατασταθεί με οποιαδήποτε άλλη της μορφής $t_1 - t_3 < c$ όπου $c > 2$, το σύστημα των ανισοτήτων παραμένει το ίδιο. Είναι δυνατόν να πάρουμε μια μοναδική αναπαράσταση ενός συνόλου ανισοτήτων, αν ο ΠΦΔ που το περιγράφει, θεωρηθεί πίνακας γειτονιάς (adjacency matrix) των ακμών του γράφου και επιλυθεί πάνω του το πρόβλημα συντομότερου μονοπατιού.



Σχήμα 4.1: WF_i: Το αυτόματο ελέγχου ορθής λειτουργίας για τον μετρητή t_i

---

Ο αλγόριθμος Floyd-Warshall χρησιμοποιείται για να λύσουμε το πρόβλημα συντομότερου μονοπατιού. Η συνάρτηση που υλοποιεί την παραπάνω εργασία ονομάζεται "κανονικοποίηση" και σημειώνεται ως *cf()*. Ο πίνακας D' που προκύπτει απο την εφαρμογή της *cf()* στον D μπορεί να είναι μη ικανοποιήσιμος. Αυτό σημαίνει οτι ο D περιγράφει ένα σύνολο ασύμβατων μεταξύ τους περιορισμών. Αν D αντιστοιχεί σε ένα σύνολο συμβατών μεταξύ τους περιορισμών, τότε η εφαρμογή της *cf* στον D θα παράγει έναν D' με τις ελάχιστες δυνατές τιμές σε όλα τα στοιχεία του d_{ij} και ο οποίος είναι μοναδικός. Αλλιώς, ένα στοιχείο d_{ii} που ανήκει στην κεντρική διαγώνιο του πίνακα έχει αρνητική τιμή.

Κατασκευή του αυτομάτου

Δοσμένης περιγραφής ενός συστήματος πραγματικού χρόνου στην μορφή (B, T), μπορεί να κατασκευαστεί ένα Büchi αυτόματο που αναγνωρίζει το ίδιο σύνολο συμπεριφορών με το (B, T).

Πρώτα απ' όλα κατασκευάζεται το αυτόματο

$$B_{UT} = B \cap WF_1 \cap WF_2 \cap \ldots \cap WF_n,$$

Είναι η τομή κάποιων πιο απλών αυτομάτων. Είμαστε ήδη εξοικιωμένοι με το αυτόματο B. Είναι το πρώτο μέλος του ζευγαριού (B, T).

Τα WF_i, ένα για κάθε μετρητή $t_i \in T$, είναι αιτιοκρατικά Büchi αυτόματα που περιγράφονται στην εικόνα 4.1. Κάθε WF_i εξασφαλίζει οτι η ακολουθία των γεγονότων που αντιστοιχούν στον μετρητή t_i είναι "καλά ορισμένη". Ας θυμηθούμε από την ενότητα 2.3.2 τον ορισμό των καλά ορισμένων ακολουθιών.

Ετσι λοιπόν το Büchi αυτόματο B_{UT} δέχεται όλα τα καλά ορισμένα ίχνη του συστήματος (B, T).

Τώρα αυτό που μας έχει απομείνει να κάνουμε είναι να προσθέσουμε στο B_{UT} αυτόματο την πληροφορία που φέρουν οι μετρητές. Ας θυμηθούμε και πάλι από την ενότητα 2.3.2 οτι οι καταστάσεις του συστήματος περιγράφονται από ζεύγη της μορφής (s, ν). Δηλαδή σε κάθε κατάσταση του συστήματος ισχύει ένα σύνολο συνθηκών πάνω στις τιμές των μετρητών. Ας θυμηθούμε οτι ισχύει $(s, \nu) \xrightarrow{a, \mathcal{B}} (s', \nu')$, όπου $\nu' = nexttv(\nu)$. Με κάθε κίνηση του αυτομάτου, συμβαίνει και ένα σύνολο από γεγονότα μετρητών $\mathcal{B}$. Το σύνολο των συνθηκών στις τιμές των μετρητών αλλάζει ανάλογα με το σύνολο $\mathcal{B}$ που συμβαίνει. Οι κανόνες με τους οποίους αλλάζουν οι τιμές των μετρητών δίνονται από τον ορισμό της *nexttv()* που δώσαμε στην ενότητα 2.3.2.

Δοσμένης της μεθόδου αναπαράστασης των χρονικών περιορισμών με ΠΦΔ, μπορούμε να αντιστοιχίσουμε με κάθε κατάσταση του αυτομάτου B_{UT} ένα ΠΦΔ που κωδικοποιεί την χρονική πληροφορία γι' αυτή την κατάσταση. Ας ανακαλέσουμε από την ενότητα 2.3.2 τους ορισμούς των συνόλων A, $\mathcal{B}$, E, S. Ενας τρόπος να υλοποιήσουμε την *nexttv()* χρησιμοποιώντας ΠΦΔ δίνεται παρακάτω:

1ο βήμα $D' \leftarrow rte(D, B)$

Η συνάρτηση *rte(D, B)* ("restrict to event set") εφαρμόζεται στο D και τον μετατρέπει σε έναν D'. Ο D' αντιστοιχεί ακριβώς στο σύνολο των τιμών των μετρητών που επιτρέπει στο σύνολο B των γεγονότων μετρητών να συμβεί. Ολοι οι εκπνέοντες μετρητές του B, πρέπει να έχουν την ίδια τιμή, η οποία πρέπει να είναι μικρότερη από τις τιμές εκείνων που δεν εκπνέουν. Οι μετρητές που τίθενται τώρα δεν επηρεάζουν την συνάρτηση *rte*. Η *rte* υλοποιείται όπως ακολουθεί:

$$d'_{ij} = min(d_{ij}, (0, \leq)), \text{ όταν } t_i, t_j \in E$$
$$d'_{ij} = min(d_{ij}, (0, <), \text{ όταν } t_i \in E \text{ και } t_j \in A - E$$
$$d'_{ij} = d_{ij} \text{ αλλιώς}$$

2ο βήμα Αν D' είναι σε κανονική μορφή, τότε

$D'' \leftarrow cf(D')$

Αν D'' είναι ικανοποιήσιμος

πήγαινε στο (βήμα 3)

αλλιώς

ανάφερε(λάθος)

3ο βήμα $D'' \leftarrow elapse(D', B)$

Η συνάρτηση *elapse* υλοποιείται όπως ακολουθεί:

$$d''_{i0} = d'_{ij} \wedge d''_{0i} = d'_{ji}, \text{ όταν } t_i \in A - E \text{ και } t_j \in E$$
$$d''_{ij} = d'_{ij} \text{ αλλιώς}$$

4ο βήμα Ολοι οι εκνέοντες μετρητές σβύνονται από το A. Αυτή είναι μια πράξη προβολής που υλοποιείται σβύνοντας όλες τις γραμμές και στήλες του πίνακα που αντιστοιχούν σε μετρητές που εκπνέουν.

5ο βήμα Τίθονται όλοι οι μετρητές των οποίων το *set* γεγονός τους ανήκει στο B. Αυτό υλοποιείται όπως ακολουθεί:

$$d_{i0} \leftarrow u_i \text{ for all } t_i \in S$$
$$d_{0i} \leftarrow -l_i \text{ for all } t_i \in S$$
$$d_{ij} = d_{ji} \leftarrow (\infty, <), \text{ για όλους τους } t_j \in A - E$$

Οι πράξεις $+$ και min σε ΠΦΔ ορίζονται όπως παρακάτω:

$$\begin{aligned} (z,r) + (z',r') &= (z+z', min(r,r')) \\ min((z,r),(z',r')) &= \begin{cases} (z,r) & if(z,r) \leq (z',r') \\ (z',r') & \text{αλλιώς} \end{cases} \end{aligned}$$

Οι καταστάσεις του αυτομάτου είναι ζεύγη της μορφής (s, D), όπου s είναι μια κατάσταση του B_{UT} και D ένας ΠΦΔ. Οι μεταβάσεις αυτού του αυτομάτου είναι της μορφής:

$$(s, D) \xrightarrow{a,\mathcal{B}} (s', D')$$

όπου

$$s \xrightarrow{a,\mathcal{B}} s'$$

είναι μια μετάβαση του B_{UT} και D' υπολογίζεται από τον D χρησιμοποιώντας τον παραπάνω αλγόριθμο.

Οι καταστάσεις αποδοχής του παραπάνω αυτομάτου είναι όλες οι καταστάσεις αποδοχής του B_{UT} για τις οποίες ο ΠΦΔ που τους αντιστοιχεί σε ένα σύνολο συμβατών περιορισμών. Αν για μια δοσμένη κατάσταση του αυτομάτου B_{UT}, ο αντίστοιχος ΠΦΔ είναι μη ικανοποιήσιμος, τότε αυτή η κατάσταση θεωρείται κατάσταση λάθους. Το σύστημα φτάνει σε κατάσταση λάθους, αν κατά τη διάρκεια της λειτουργίας του παραβιάσει τους χρονικούς περιορισμούς. Αποδεικνύεται στο [Dil89] οτι αυτό το αυτόματο αναγνωρίζει το ίδιο σύνολο ιχνών με το ζεύγος (B, T).

4.2 Ελεγχος μοντέλου

Ο έλεγχος μοντέλου (model checking) είναι μια ισχυρή τεχνική για αυτόματη επαλήθευση συστημάτων. Οι αλγόριθμοι που βασίζονται στην παραπάνω τεχνική, αποφασίζουν το σύνολο των καταστάσεων που ικανοποιούν μια λογική έκφραση, ενός συστήματος μεταβάσεων με ετικέττες που περιγράφει ένα σύστημα πραγματικού χρόνου. Το βασικό μειονέκτημα αυτών των αλγορίθμων είναι οτι το πλήθος των καταστάσεων που ελέγχονται αυξάνει εκθετικά με τον αριθμό των παράλληλων διεργασιών που αποτελούν το σύστημα. Ενας τρόπος να ξεπεράσουμε αυτό το πρόβλημα (state explosion problem) είναι να αναπαριστούμε το σύνολο των καταστάσεων "συμβολικά" και όχι "απαριθμητικά".

Οι απαριθμητικές μέθοδοι βασίζονται στην κατασκευή ενός γράφου που περιγράφει το συνολικό σύστημα και μετά στον υπολογισμό της λογικής έκφρασης, με εξαντλητική απαρίθμηση των καταστάσεων του γράφου οι οποίες ικανοποιούν την έκφραση.

Οι συμβολικές μέθοδοι βασίζονται στην αναπαράσταση των καταστάσεων του γράφου, ο οποίος περιγράφει το σύστημα, με κατηγορήματα και στον υπολογισμό της έκφρασης άμεσα, χωρίς δηλαδή να πρέπει να κατασκευάσουμε ολόκληρο το γράφο. Θα περιγράψουμε στην επόμενη ενότητα πώς επιτυγχάνεται αυτή η συμβολική αναπαράσταση.

4.2.1 Συμβολικός Ελεγχος Μοντέλου

Υπάρχουν δυο καίρια σημεία, πίσω από την ιδέα του αλγορίθμου για συμβολικό έλεγχο μοντέλου. Το πρώτο είναι ο ορισμός μιας κατάλληλης σχέσης "επόμενης κατάστασης", της οποίας η επαναλειπτική εφαρμογή επιτρέπει τον υπολογισμό της επιθυμητής ιδιότητας. Το δεύτερο πολύ σημαντικό σημείο είναι η αναπαράσταση των καταστάσεων ενός χρονικού συστήματος μεταβάσεων με ετικέττες, με κατηγορήματα καταστάσεων. Ετσι το σύνολο των καταστάσεων που ικανοποιούν μια έκφραση υπολογίζεται σαν ένα λογικό κατηγόρημα.

Σε οτι ακολουθεί θα σκιαγραφήσουμε την κεντρική ιδέα του αλγορίθμου. Εκτεταμένες λεπτομέρειες για την μέθοδο μπορούν να ανευρεθούν στα [HNSY92, Yov93].

Ας θεωρήσουμε το χρονικό σύστημα μεταβάσεων με ετικέττες $B = (\Sigma, S, \rightarrow, s_0)$ και $S_1, S_2 \subset S$. Η σχέση "επόμενης κατάστασης" ορίζεται ως εξης: Για όλες τις $s \in S$, $S_1 \triangleright S_2$ είναι το σύνολο των καταστάσεων τέτοιο ώστε $s \in S_1 \triangleright S_2$, αν υπάρχουν s' και $t \in R^+$ με $s \overset{t}{\triangleright} s'$, και για όλα τα $t' \leq t$, $s + t \in S_1 \vee S_2$.

Μια "συμβολική κατάσταση" ορίζεται σαν (s, ψ), όπου s είναι μια κορυφή ενός χρονικού γράφου και ψ είναι ένας χρονικός περιορισμός. Μια συμβολική κατάσταση (s,ψ) παριστά το σύνολο όλων των καταστάσεων του συστήματος μεταβάσεων, που είναι το μοντέλο του παραπάνω χρονικού γράφου, της μορφής (s,ν), τέτοιες ώστε ν ικανοποιεί το ψ. Με άλλα λόγια $(s,\psi) = \{(s,\nu) \in S \mid \nu \; satisfies \; \psi\}$.

Οι χρονικοί περιορισμοί ψ αναπαρίστανται με τους ΠΦΔ. Εχουμε ήδη παρουσιάσει συνοπτικά τον λογισμό πάνω σε ΠΦΔ στην ενότητα 4.1.1.

Παρουσιάζουμε τώρα τον ορισμό της συνάρτησης $\mathcal{D}$ για την αναπαράσταση των κατηγορημάτων καταστάσεων ϕ με ένα σύνολο από συμβολικές καταστάσεις. Ο ορισμός της $\mathcal{D}$ γίνεται επαγωγικά ως προς την κατασκευή του κατηγορήματος:

$$
\begin{array}{lcl}
\mathcal{D}(p) & = & \bigcup_{s\in S, p\in\mu(s)}(s,f) \;\cup\; \bigcup_{s\in S, p\notin\mu(s)}(s, false) \\
\mathcal{D}(x \prec c) & = & \bigcup_{s\in S}(s, f \wedge x \prec c) \\
\mathcal{D}(x - y \prec c) & = & \bigcup_{s\in S}(s, f \wedge x - y \prec c) \\
\mathcal{D}(\neg\phi) & = & \neg\mathcal{D}(\phi) \wedge \bigcup_{s\in S}(s,f) \\
\mathcal{D}(\phi_1 \wedge \phi_2) & = & \mathcal{D}(\phi_1) \wedge \mathcal{D}(\phi_2)
\end{array}
$$

Τελικά, παρουσιάζουμε τον αλγόριθμο συμβολικού ελέγχου για μια έκφραση της ϕ χρονικής λογικής TCTL. Ο αλγόριθμος υπολογίζει το σύνολο $\mathcal{S}(\phi)$ των συμβολικών καταστάσεων που ικανοποιούν την έκφραση ϕ.

Ας υποθέσουμε οτι $G = (\Sigma, S, s_0, C, E, f)$ είναι ένας χρονικός γράφος και ϕ μια έκφραση TCTL. Το $\mathcal{S}(\phi)$ υπολογίζεται όπως παρακάτω:

$$
\begin{array}{lcl}
\mathcal{S}(p) & = & \mathcal{D}(p) \\
\mathcal{S}(x \prec c) & = & \mathcal{D}(x \prec c) \\
\mathcal{S}(x - y \prec c) & = & \mathcal{D}(x - y \prec c) \\
\mathcal{S}(\neg\phi) & = & \neg\mathcal{S}(\phi) \\
\mathcal{S}(\phi_1 \wedge \phi_2) & = & \mathcal{S}(\phi_1) \wedge \mathcal{S}(\phi_2) \\
\mathcal{S}(\phi_1 \exists\mathcal{U} \phi_2) & = & \bigvee_{k\in N} X_k, \\
 & & \text{όπου } X_0 = \mathcal{S}(\phi_2) \text{ και} \\
 & & X_{k+1} = X_k \vee \mathcal{S}(\phi_1) \rhd X_k \\
\mathcal{S}(\phi_1 \forall\mathcal{U} \phi_2) & = & \bigvee_{k\in N} X_k, \\
 & & \text{όπου } X_0 = \mathcal{S}(\phi_2) \text{ και} \\
 & & X_{k+1} = X_k \vee (\neg X_k[z := 0])\exists\mathcal{U}((\neg\mathcal{S}(\phi_1) \wedge \neg X_k) \vee z > 1)
\end{array}
$$

Κεφάλαιο 5

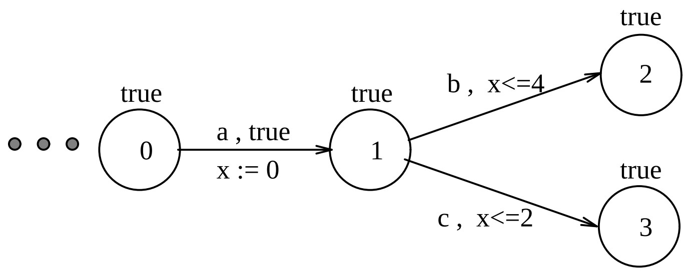
Αναπαράσταση χρονικών γράφων με Set/Expire χρονικούς γράφους

Στο κεφάλαιο 2 περιγράφουμε δυο διαφορετικές μεθόδους για την περιγραφή συστημάτων πραγματικού χρόνου. Η βασική ιδέα πίσω από αυτή την εργασία είναι η ενοποίηση αυτών των δυο μεθόδων. Σ' αυτό το κεφάλαιο, θα περιγράψουμε ένα μεταφραστικό σχήμα τέτοιο ώστε, δοσμένου ενός χρονικού γράφου G κατασκευάζει έναν *Set/Expire* χρονικό γράφο ο οποίος περιγράφει τις ίδιες ακριβώς συμπεριφορές με τον G.

5.1 Μια επέκταση του μοντέλου των Set/Expire χρονικών γράφων

Το μοντέλο των *Set/Expire* χρονικών γράφων που περιγράψαμε στο κεφάλαιο 2 είναι ένα αρκετά περιοριστικό μοντέλο. Είναι πραγματικά δύσκολο να περιγράψουμε κάποια είδη συμπεριφορών, όπως timeout, μη αιτιοκρατική επιλογή κλπ., χρησιμοποιώντας το μοντέλο των *Set/Expire* χρονικών γράφων. Ας δούμε το παρακάτω παράδειγμα:

Παράδειγμα 5.1 Ας θεωρήσουμε ένα σύστημα το οποίο κάποια στιγμή εμφανίζει την ακόλουθη συμπεριφορά: Εκτελεί ένα a και πηγαίνει σε μια κατάσταση όπου μπορεί είτε να μείνει για πάντα, είτε να μπορεί να αποφασίσει να εκτελέσει b αλλά σε χρόνο μικρότερο των 4 χρονικών μονάδων μετά το a ή να εκτελέσει c αλλά μόνο σε χρόνο μικρότερο των 2 χρονικών μονάδων μετά το a. Η παραπάνω συμπεριφορά μπορεί να περιγραφεί πολύ απλά με τον χρονικό γράφο που φαίνεται στην εικόνα 5.1.

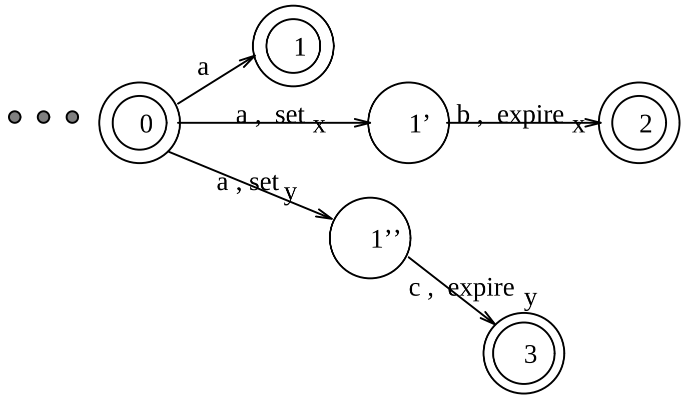


Σχήμα 5.1: Ενας χρονικός γράφος που περιγράφει ένα σύστημα με μη αιτιοκρατική συμπεριφορά

Αλλά αν χρησιμοποιήσουμε το μοντέλο των *Set/Expire* χρονικών γράφων, η περιγραφή αυτής της συμπεριφοράς γίνεται πολύ πιο πολύπλοκη. Η περιγραφή δίνεται στην εικόνα 5.2. Η κατάσταση νούμερο 1 του χρονικού γράφου, πρέπει να διασπαστεί σε τρεις διαφορετικές καταστάσεις. Αυτό συμβαίνει λόγω των περιοριστικών σημασιολογικών κανόνων του μοντέλου *Set/Expire* χρονικών γράφων, ειδικά λόγω των συνθηκών για τα καλά ορισμένα ίχνη.

Παρατηρήστε οτι το μοντέλο της εικόνας 5.2 έχει το ίδιο σύνολο ιχνών με τον χρονικό γράφο της εικόνας 5.1. Αλλά δεν έχουμε παρ' όλα αυτά καταφέρει να δημιουργήσουμε ένα λειτουργικά ισοδύναμο μοντέλο. Ο χρονικός γράφος, ενώ είναι στην κατάσταση νούμερο 1 και για το χρονικό διάστημα $[0, 2]$ μπορεί μη αιτιοκρατικά να αποφασίσει να εκτελέσει είτε b είτε c. Αυτό όμως δεν συμβαίνει για το μοντέλο της εικόνας 5.2. Το μοντέλο των *Set/Expire* χρονικών γράφων, δεν είναι ευέλικτο στο να περιγράφει τέτοιου είδους συμπεριφορές.

Σε αυτή την ενότητα, θα περιγράψουμε μια επέκταση του μοντέλου των *Set/Expire* χρονικών γράφων. Το επιθυμητό σύστημα περιγράφεται και πάλι σαν ένα ζεύγος (B, T). B είναι το Büchi αυτόματο που περιγράφει τις χρονικά ανεξάρτητες συμπεριφορές του συστήματος μαζί με τα γεγονότα θέσης και εκπνοής των μετρητών και T είναι ένα πεπερασμένο σύνολο μετρητών που εκφράζουν την χρονική πληροφορία του συστήματος. Παρ' όλα αυτά, στην επέκταση που εμείς προτείνουμε, η πληροφορία που εκφράζουν οι μετρητές, διαφέρει από αυτήν των παραδοσιακών *Set/Expire* χρονικών γράφων. Εμείς βρήκαμε οτι μπορούμε να τροποποιήσουμε το μοντέλο των *Set/Expire* χρονικών γράφων, με το να χαλαρώσουμε τους κανόνες ορθής λειτουργίας των μετρητών, χωρίς να χάσουμε σε εκφραστικότητα. Επίσης, με την επέκταση του μοντέλου των *Set/Expire* χρονικών γράφων που προτείνουμε, μπορούμε με απλούστερες περιγραφές, να περιγράφουμε τα ίδια πράγματα που περιγράφουμε και με το παραδοσιακό μοντέλο.



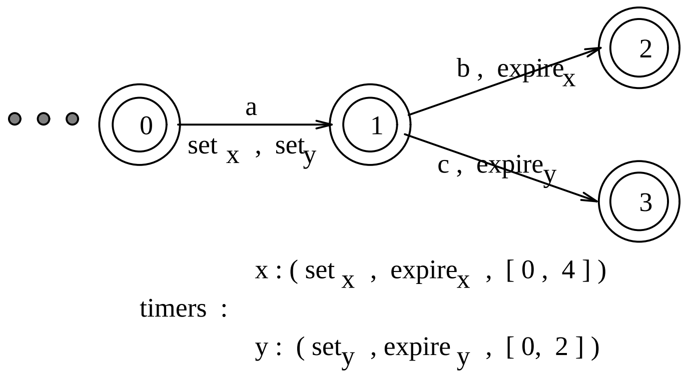
timers $x : (set_x,\ expire_x,\ [0,\ 4])$

$y : (set_y,\ expire_y,\ [0,\ 2])$

Σχήμα 5.2: Το μοντέλο ενός συστήματος με μη αιτιοκρατική συμπεριφορά, που χρησιμοποιεί το μοντέλο των *Set/Expire* χρονικών γράφων

Οι μετρητές του δικού μας μοντέλου, μεταφέρουν λιγότερη πληροφορία από αυτούς του παραδοσιακού μοντέλου. Ας θυμηθούμε από την περιγραφή του μοντέλου των *Set/Expire* χρονικών γράφων οτι κάθε μετρητής θέτει ένα περιορισμό της μορφής: Μιας και τεθεί ένας μετρητής, αναγκάζεται να εκπνεύσει οποσδήποτε και ο χρόνος που έχει περάσει μεταξύ της θέσης και της εκπνοής του μετρητή είναι μια πραγματική τιμή μέσα σε ένα διάστημα χρόνων. Οι μετρητές στο παραδοσιακό μοντέλο των *Set/Expire* χρονικών γράφων όχι μόνο μεταφέρουν χρονική πληροφορία, αλλά και θέτουν συνθήκες αποδοχής στις συμπεριφορές του συστήματος. Αυτές οι συνθήκες, εκφράζονται με τις συνθήκες καλού ορισμού των ακολουθιών των γεγονότων μετρητών. Ολες οι συμπεριφορές του συστήματος που δεν σέβονται τις παραπάνω συνθήκες απορρίπτονται σαν λάθος συμπεριφορές.

Στο μοντέλο μας, η πληροφορία των μετρητών εξηγείται διαφορετικά: Αν ένας μετρητής τεθεί, εαν ποτέ εκπνεύσει, ο χρόνος που έχει περάσει είναι μια πραγματική τιμή σε ένα συγκεκριμένο διάστημα χρόνων. Με άλλα λόγια, στο μοντέλο μας, οι μετρητές εκφράζουν μόνο χρονική πληροφορία για το σύστημα. Στο μοντέλο μας λοιπόν, ο μόνος τρόπος να εκφράζουμε συνθήκες αποδοχής για το σύστημα είναι να αποφασίζουμε κατάλληλα το σύνολο F των καταστάσεων αποδοχής του Büchi αυτομάτου, το οποίο περιγράφει τις χρονικά ανεξάρτητες συμπεριφορές.



Σχήμα 5.3: Το μοντέλο ενός συστήματος με μη αιτιοκρατική συμπεριφορά χρησιμοποιώντας την επέκταση του μοντέλου *Set/Expire* χρονικών γράφων

Στην εικόνα 5.3 δίνουμε την περιγραφή του συστήματος της 5.1, χρησιμοποιώντας την επέκταση του μοντέλου των *Set/Expire* χρονικών γράφων. Οπως μπορούμε να παρατηρήσουμε, η δομή του γράφου μας είναι η ίδια με αυτή του χρονικού γράφου της εικόνας 5.1.

Τώρα είναι ώρα να ορίσουμε τυπικά το μοντέλο που προτείνουμε.

5.1.1 Μετρητές

Ενας "μετρητής" ορίζεται και πάλι σαν μια τριάδα: $t_i = (set_i, expire_i, interval_i)$. set_i και $expire_i$ είναι γεγονότα μετρητών και $interval_i$ είναι ένα διάστημα χρόνων. Τα παραπάνω τρία στοιχεία ορίζονται ακριβώς όπως για το μοντέλο των *Set/Expire* χρονικών γράφων. Εστω T ένα πεπερασμένο σύνολο μετρητών, που παίρνουν τις τιμές τους στο σύνολο των πραγματικών αριθμών, R. Οταν ένας μετρητής τίθεται, παίρνει μια μη αρνητική πραγματική τιμή t_i τέτοια ώστε $l_i \prec t_i \prec u_i$. l_i είναι το κάτω φράγμα και u_i είναι το άνω φράγμα του διαστήματος $interval_i$ και τέλος $\prec$ είναι ένα από τα σύμβολα $\leq$ or $<$ ανάλογα με το αν το διάστημα είναι κλειστό ή ανοιχτό αντίστοιχα. Ενας μετρητής μπορεί να θεωρηθεί "ενεργός" αν έχει τεθεί και δεν έχει ακόμα εκπνεύσει. Στο μοντέλο μας, υπάρχουν περιπτώσεις που ένας μετρητής έχει αρνητική τιμή. Αυτό μπορεί να ερμηνευθεί διαισθητικά όπως παρακάτω: Ενας μετρητής με αρνητική τιμή, θα' πρεπε να έχει ήδη εκπνεύσει, επειδή το διάστημα χρόνων που μετρούσε, έχει ήδη περάσει, σε σχέση με τους άλλους ενεργούς μετρητές. Η τιμή αυτού του συγκεκριμένου μετρητή είναι πλέον αδιάφορη αν από εδώ και πέρα, στο μέλλον, δεν θα συναντήσουμε το γεγονός εκπνοής του. Με αυτή την προυπόθεση, ο μετρητής δεν επηρεάζει τον υπολογισμό της επόμενης κατάστασης του συστήματος. Αλλά αν συμβεί το γεγονός εκπνοής του να εμφανισθεί σε ένα ίχνος, πριν από ένα γεγονός θέσης, που επαναρχικοποιεί τον μετρητή, αυτό είναι μια λανθασμένη λειτουργία του συστήματος. Μιας και τα γεγονότα μετρητών μπορούν να εμφανισθούν στα ίχνη του συστήματος, όπως το σύστημα εξελίσσεται, παράγει ακολουθίες από γεγονότα μετρητών. Σ' αυτό το σημείο, το μοντέλο μας διαφέρει, ριζικά από παραδοσιακό μοντέλο.

Πρώτ' απ' όλα, έχουμε χαλαρώσει τις συνθήκες καλού ορισμού (well-formedness conditions). Την μόνη που έχουμε διατηρήσει, είναι οτι οι μετρητές εκπνέουν μόνο όταν είναι ενεργοί. Πιο συγκρκριμένα, οι μετρητές του μοντέλου μας έχουν τις ακόλουθες ιδιότητες:

1. Οι ενεργοί μετρητές μπορούν να επανατεθούν αυθαίρετο αριθμό φορών πριν εκπνεύσουν. Αυτό διαισθητικά σημαίνει οτι μεταξύ δυο διαδοχικών εμφανίσεων του γεγονότος *set* ενός μετρητή, δεν είναι απαραίτητο να εμφανισθεί το γεγονός εκπνοής του. Οταν συμβαίνει το *expire* γεγονός ενός μετρητή, αντιστοιχεί πάντα στο τελευταίο *set* γεγονός που έχει συμβεί.

2. Οι μετρητές εκπνέουν μόνο αν είναι ενεργοί.

3. Οι ενεργοί μετρητές μπορεί να μην εκπνέουν ποτέ.

Εστω $\nu : T \rightarrow R$ μια συνάρτηση που ονομάζεται "αποτίμηση μετρητών". Μια "αποτίμηση μετρητών" αναθέτει σε κάθε μετρητή $t_i \in T$, μια τιμή στο σύνολο R. Το σύνολο V είναι το σύνολο $[T \rightarrow R]$ όλων των αποτιμήσεων μετρητών. Για $t \in R$ και $\nu \in V$, $\nu - t$, δηλώνει την αποτίμηση $\nu'(t_i) = \nu(t_i) - t, \forall t_i \in T$.

Οπως στο παραδοσιακό μοντέλο των *Set/Expire* χρονικών γράφων, δεν μπορούμε να γνωρίζουμε την τιμή ενός μετρητή, εκτός και αν συμβαίνει κάποιο γεγονός μετρητή. Εστω $A \subseteq T$ το σύνολο των ενεργών μετρητών, ν μια αποτίμηση μετρητών και $\mathcal{B}$ ένα σύνολο από γεγονότα μετρητών. Εστω ακόμα $E = \{t_i \mid expire_i \in \mathcal{B}\}$ και $S = \{t_j \mid set_j \in \mathcal{B}\}$. Θέλουμε να υπολογίσουμε την επόμενη συνάρτηση αποτίμησης μετρητών. Οι μετρητές με αρνητική τιμή δεν λαμβάνουν μέρος στον υπολογισμό. Γι' αυτό ακριβώς το λόγο, υπολογίζουμε πρώτα το σύνολο των "σχετικών μετρητών", $A_{relevant} = A - \{t_i \mid t_i \in A \wedge t_i \notin E \wedge \exists t_j \in E, \nu(t_i) - \nu(t_j) < 0\}$. Η $nexttv(\nu)$ δηλώνει την αποτίμηση ν' τέτοια ώστε:

$$\begin{array}{lll} \exists t \in R^+ : & \forall t_i \in E : & \nu(t_i) = t \\ & \wedge \;\; \forall t_k \in A_{relevant} - E : & \nu(t_k) - t > 0 \wedge \nu'(t_k) = \nu(t_k) - t \\ & \wedge \;\; \forall t_j \in S : & l_j \prec \nu'(t_j) \prec u_j, \prec \in \{\leq, <\} \end{array}$$

Το νέο σύνολο ενεργών μετρητών που προκύπτει είναι το $A' = (A_{relevant} - E) \cup S$.

Μια αποτίμηση μετρητών που υπολογίζεται με την παραπάνω μέθοδο, μπορεί να είναι "κενή". Αυτό σημαίνει οτι δεν υπάρχει μη αρνητικός πραγματικός αριθμός t που να ικανοποιεί τους παραπάνω κανόνες. Μια κενή αποτίμηση μετρητών σημαίνει διαισθητικά οτι το σύνολο $\mathcal{B}$ των γεγονότων μετρητών δεν μπορεί να συμβεί κάτω από τις τρέχουσες τιμές των μετρητών. Αυτό επιβάλλει συνθήκες διάταξης μεταξύ των εκπνοών των διαφόρων μετρητών. Οι παραπάνω κανόνες εγγυόνται την ακόλουθη ιδιότητα:
"Για κάθε μετρητή $t_i \in T$ ο χρόνος που περνά μεταξύ της χρονικής στιγμής που συμβαίνει το set_i και αυτής που συμβαίνει το πρώτο $expire_i$, μετά το set_i, αν ποτέ συμβαίνει, είναι $l_i \prec t_{expire_i} - t_{set_i} \prec u_i, \prec \in \{\leq, <\}$ ".

5.1.2 Συντακτικό και σημασιολογία

Τυπικά, ένα (B, T) ορίζεται ως εξής:

1. $B = (\Sigma, S, S_0, E, F)$ είναι ένα Büchi αυτόματο, όπου:

   - $\Sigma = Act \cup 2^{\{set_i, expire_i | \forall t_i \in T\}}$. Το Σ περιέχει όλα τα γεγονότα του συστήματος που ανήκουν στο σύνολο Act και όλα τα γεγονότα μετρητών,
   - S είναι ένα πεπερασμένο σύνολο καταστάσεων,
   - $S_0 \subseteq S$ είναι ένα σύνολο αρχικών καταστάσεων,
   - $E \subseteq S \times Act \times 2^{\{set_i, expire_i | \forall t_i \in T\}} \times S$ ένα σύνολο ακμών,
   - F είναι ένα σύνολο καταστάσεων αποδοχής

2. $T = \{t_1, t_2 \ldots t_n\}$ είναι ένα πεπερασμένο σύνολο μετρητών.

Τώρα θα ορίσουμε το σύνολο των συμπεριφορών του (B, T). Ας σημειώνουμε με τ^{set_i} τη χρονική στιγμή που συμβαίνει το σύνολο e_j από γεγονότα μετρητών. Και έστω οτι $set_i \in e_j$. Ομοια ορίζουμε την τ^{expire_i}. Το σύνολο των συμπεριφορών του (B, T) είναι το σύνολο όλων των χρονικών ιχνών (e, τ), τα οποία ικανοποιούν την ακόλουθη συνθήκη:

1. e είναι ένα ίχνος του Büchi αυτομάτου B,

2. τ είναι μια χρονική ακολουθία,

3. Αν $\exists \tau^{expire_i}$ τότε $\exists \tau'^{set_i} < \tau^{expire_i}$ τέτοια ώστε $\tau^{expire_i} - \tau'^{set_i} \in interval_i$ και $\nexists \tau''^{set_i}$ τέτοιο ώστε $\tau'^{set_i} \leq \tau''^{set_i} < \tau^{expire_i}$

Τώρα που δώσαμε τη σημασιολογία των ζευγών (B, T), είναι προφανές πώς ορίζεται η τομή δυο τέτοιων ζευγαριών. Η συμπεριφορές της τομής δυο ζευγών (B_1, T_1), (B_2, T_2) είναι ακριβώς αυτές του $(B_1 \cap B_2, T_1 \cup T_2)$.

Το μοντέλο

Τα (B, T) μπορούν να αναπαριστούν συστήματα πραγματικού χρόνου. Οι καταστάσεις του συστήματος μπορούν να παρασταθούν με ένα ζεύγος (s, ν) όπου s είναι μια θέση ελέγχου και ν είναι μια αποτίμηση των μετρητών. Οι θέσεις ελέγχου μπορούν να παρασταθούν με καταστάσεις του αυτομάτου B.

Το σύστημα ξεκινά από την αρχική κατάσταση $(s_0, \nu_{undefined})$ όπου s_0 είναι μια αρχική κατάσταση του παραπάνω αυτομάτου, και όλοι οι μετρητές του συστήματος είναι ανενεργοί, που σημαίνει οτι η τιμή τους είναι αόριστη. Το σύστημα κινείται από μια κατάσταση $(s, \nu) \overset{a,\mathcal{B}}{\to} (s', \nu')$ εκτελώντας ένα σύνολο γεγονότων του συστήματος a και ένα σύνολο γεγονότων μετρητών $\mathcal{B}$, αν υπάρχει ακμή του αυτομάτου τέτοια ώστε $s \xrightarrow{a,\mathcal{B}} s'$. Η αποτίμηση ν' είναι η $nexttv(\nu)$ που προκύπτει εκτελώντας το συνολο γεγονότων μετρητών $\mathcal{B}$.

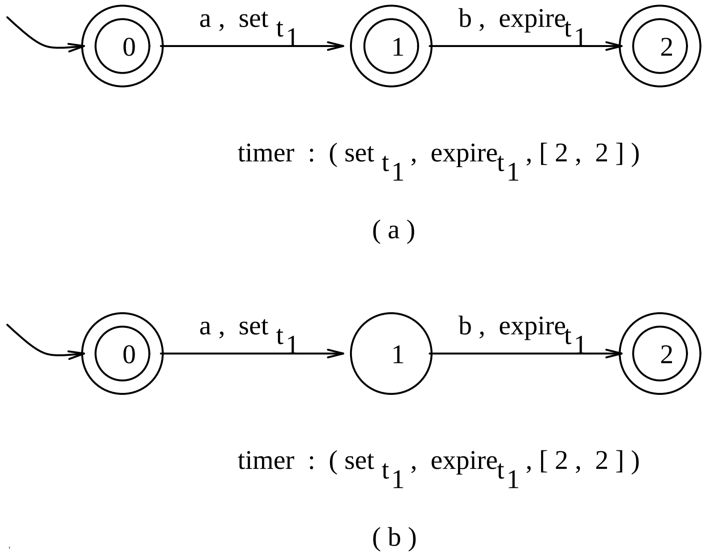
Τελικά, η συμπεριφορά του συστήματος μπορεί να περιγραφεί από τον ακόλουθο κανόνα μετάβασης:

$$\frac{s \xrightarrow{a,\mathcal{B}} s'}{(s, \nu) \overset{a,\mathcal{B}}{\to} (s', nexttv(\nu))}$$

5.1.3 Σύγκριση του μοντέλου των Set/Expire χρονικών γράφων και της επέκτασης του

Η διαφορά μεταξύ του μοντέλου των *Set/Expire* χρονικών γράφων και της επέκτασής του, θα δωθεί με ένα παράδειγμα.

Παράδειγμα 5.2 Ας θεωρήσουμε τα δυο μοντέλα που δίνονται στις εικόνες 5.4(a) και 5.4(b). Σύμφωνα με τη σημασιολογία των *Set/Expire* χρονικών γράφων, αυτά τα δυο μοντέλα περιγράφουν το ίδιο σύστημα. Το σύστημα έχει την ακόλουθη συμπεριφορά: Είτε δεν κάνει τίποτα και μένει στην κατάσταση 0 για πάντα, είτε εκτελεί a και ακριβώς 2 μονάδες χρόνου μετά το a εκτελεί b. Τα δυο μοντέλα λοιπόν της εικόνας 5.4 δεν μπορούν να διαχωριοθούν μιας



Σχήμα 5.4: Δυο μοντέλα που δείχνουν τη διαφορά μεταξύ των *Set/Expire* χρονικών γράφων και της επεκτασής τους

και η συνθήκη αποδοχής για την κατάσταση 1 δεν παίζει κανένα ρόλο. Αυτό συμβαίνει γιατί η συνθήκη ορθής λειτουργίας του μοντέλου των *Set/Expire* χρονικών γράφων, αναγκάζει τον μετρητή t_1 να εκπνεύσει, από τη στιγμή που έχει ήδη τεθεί. Ετσι, όλες οι συμπεριφορές που είναι συμβατές με τους χρονικούς περιορισμούς, δεν μένουν για πάντα στην κατάσταση 1. Αυτός ο περιορισμός τίθεται από τον μετρητή t_1.

Από την άλλη μεριά, με την σημασία που εμείς δίνουμε στους μετρητές, τα δυο μοντέλα της εικόνας 5.4 περιγράφουν διαφορετικά συστήματα. Το μοντέλο της εικόνας 5.4(a) περιγράφει ένα σύστημα το οποίο μπορεί να μείνει για πάντα στην κατάσταση 1 ή να εκτελέσει b ακριβώς 2 χρονικές μονάδες μετά την εκτέλεση του a. Μιας και η κατάσταση 1 είναι σημειωμένη σαν κατάσταση αποδοχής και κάτω από τη σημασιολογία του δικού μας μοντέλου, το σύνολο των "καλών" συμπεριφορών του συστήματός μας περιέχει και εκείνες που μένουν για πάντα στην κατάσταση 1. Αλλά αν το b εμφανιστεί σε κάποιο ίχνος του συστήματος, θα εμφανιστεί ακριβώς 2 χρονικές μονάδες μετά το a.

Αν θέλαμε το σύστημά μας να έχει το ίδιο σύνολο συμπεριφορών με το μοντέλο των *Set/Expire* χρονικών γράφων, θα έπρεπε να το εξαναγκάσουμε να εκτελεί πάντα b μετά τα a. Για να το πετύχουμε αυτό, πρέπει να απαλείψουμε την συνθήκη αποδοχής από την κατάσταση 1.

5.2 Περιορισμένοι χρονικοί γράφοι

Σ' αυτή την ενότητα θα ορίσουμε τους περιοριομένους χρονικούς γράφους. Ας ορίσουμε πρώτα τη μορφή των χρονικών περιορισμών για τους περιορισμένους χρονικούς γράφους. Εστω C ένα πεπερασμένο σύνολο από ρολόγια, που παίρνουν τις τιμές τους στο σύνολο R^+, το σύνολο των μη αρνητικών πραγματικών αριθμών. Το σύνολο Φ των χρονικών περιορισμών γεννιέται από την ακόλουθη γραμματική:

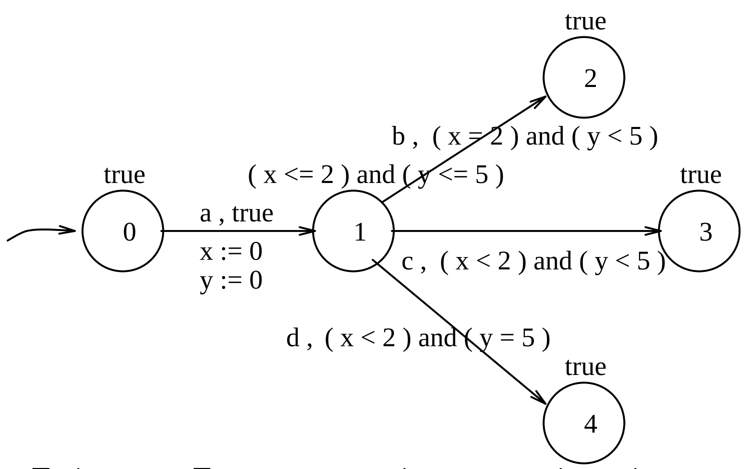
$$\begin{array}{rcl} \phi & ::= & true \\ & | & c \leq n \\ & | & c < n \\ & | & c \geq n \\ & | & c > n \\ & | & c = n \\ & | & \phi_1 \wedge \phi_2 \end{array}$$

όπου $c \in C$ και $n \in N$.

Ορίζουμε ακόμα την συνάρτηση $bound(\phi)$ όπως ακολουθεί:

$$bound(\phi) \;\; ::= \;\; \begin{cases} true & \text{αν} \left\{ \begin{array}{rcl} \phi & = & true \\ & \text{ή} & \\ \phi & = & (c \geq n) \\ & \text{ή} & \\ \phi & = & (c > n) \end{array} \right. \\ \\ c \leq n & \text{αν} \left\{ \begin{array}{rcl} \phi & = & (c \leq n) \\ & \text{ή} & \\ \phi & = & (c = n) \end{array} \right. \\ \\ c < n & \text{αν} \quad \phi = (c < n) \\ \\ bound(\phi_1) \wedge bound(\phi_2) & \text{αν} \quad \phi = (\phi_1 \wedge \phi_2) \end{cases}$$

Ενας "περιορισμένος χρονικός γράφος" είναι ένας χρονικός γράφος $G = (\Sigma, S, s_0, C, E, f)$ που ικανοποιεί την ακόλουθη ιδιότητα:



Σχήμα 5.5: Ενας περιορισμένος χρονικός γράφος

$$\forall \quad s \in S \quad f(s) \;=\; \begin{cases} true & \text{ή} \\ \cup_{i=1}^{n} bound(\phi_i) \end{cases}$$

όπου n είναι ο αριθμός των ακμών που εξέρχονται της κατάστασης s και ϕ_i είναι ο περιορισμός της $i-$ στής εξέρχουσας ακμής. Η $f(s)$ δηλώνει το αναλλοίωτο της κατάστασης s και ϕ_i είναι ο φύλακας της $i-$ οστης εξέρχουσας ακμής από την κατάσταση s.

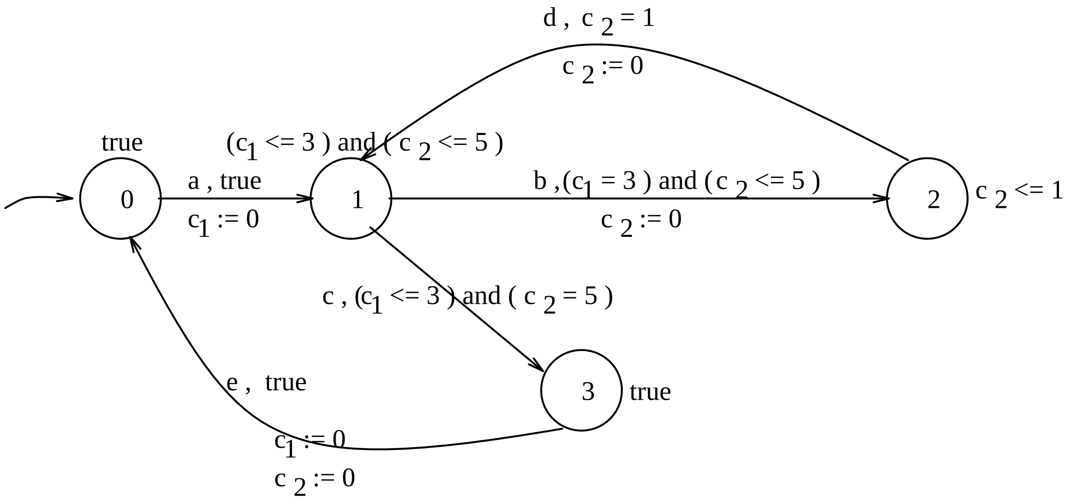
Ο γράφος της εικόνας 5.5 είναι ένας περιοριομένος χρονικός γράφος επειδή όλες του οι καταστάσεις ικανοποιούν την ιδιότητα που διατυπώσαμε παραπάνω.

5.3 Το μεταφραστικό σχήμα

Σ'αυτή την ενότητα θα προτείνουμε ένα μεταφραστικό σχήμα για να μεταφράζουμε χρονικούς γράφους στην επέκταση του μοντέλου των *Set/Expire* χρονικών γράφων. Πιο συγκεκριμένα, το μεταφραστικό σχήμα παίρνει σαν είσοδο έναν "περιορισμένο χρονικό γράφο", ο οποίος περιγράφει ένα σύστημα πραγματικού χρόνου, και παράγει ένα ισοδύναμο μοντέλο του συστήματος δοσμένο με την μορφή ενός ζεύγους (B, T).

Η κεντρική ιδέα της μετάφρασης είναι να αποσπάσουμε τη χρονική πληροφορία που μεταφέρει στις ακμές και στις κορυφές του ο χρονικός γράφος G και να την εκφράσουμε με ένα σύνολο μετρητών T.

Το Büchi αυτόματο B έχει την ίδια δομή με τον χρονικό γράφο. Αυτό που απομένει είναι να ορισθούν οι συνθήκες αποδοχής του αυτομάτου. Οι συνθήκες αποδοχής εξάγονται από την χρονική πληροφορία που δίνεται από τα αναλλοίωτα του χρονικού γράφου. Στην έκδοση των



Σχήμα 5.6: Ενας περιορισμένος χρονικός γράφος

χρονικών γράφων που χρησιμοποιούμε, οι συνθήκες αποδοχής συνεπάγονται έμμεσα από ένα συνδυασμό της πληροφορίας που παρέχουν τα αναλλοίωτα και οι φύλακες του γράφου.

5.3.1 Ενα παράδειγμα

Εισάγουμε την μέθοδο με ένα παράδειγμα.

Ας θεωρήσουμε τον περιορισμένο χρονικό γράφο της εικόνας 5.6.

Η ιδέα της μετάφρασης είναι η ακόλουθη:

Για κάθε περιορισμό της μορφής $(c_i \prec n)$, όπου $\prec$ ανήκει στο $\leq, <, \geq, >, =$, που εμφανίζεται στις ακμές του χρονικού γράφου, δημιουργούμε έναν μετρητή. Το διάστημα που αντιστοιχίζεται σε κάθε μετρητή, καθορίζει το ίδιο διάστημα τιμών με τον αντίστοιχο περιορισμό. Αν για παράδειγμα ο περιορισμός είναι της μορφής $c_i \leq 5$ τότε το διάστημα τιμών του μετρητή είναι $[0, 5]$. Παρόμοια αν ο περιορισμός είναι $c_i > 5$ το διάστημα του δημιουργούμενου μετρητή είναι $(5, \propto)$.

Το Büchi αυτόματο έχει την ίδια δομή με τον χρονικό γράφο. Εχει το ίδιο σύνολο καταστάσεων. Κάθε κατάσταση του χρονικού γράφου, που το αναλλοίωτό της είναι $true$ μαρκάρεται σαν κατάσταση αποδοχής του Büchi. Ολες οι υπόλοιπες καταστάσεις του χρονικού γράφου είναι απλώς και καταστάσεις του Büchi.

Το Büchi αυτόματο, έχει και το ίδιο σύνολο ακμών με αυτό του χρονικού γράφου. Το γεγονός θέσης (set) του μετρητή, που έχει δημιουργηθεί για να αντιστοιχεί στον περιορισμό

$(c_i \prec n)$, εμφανίζεται σε κάθε ακμή του αυτομάτου η οποία αντιστοιχεί στην ακμή του γράφου όπου εμφανίζεται η επανάθεση του ρολογιού c_i. Το γεγονός εκπνοής ($expire$) του μετρητή, εμφανίζεται στην ακμή του αυτομάτου η οποία αντιστοιχεί στην ακμή του γράφου όπου εμφανίζεται ο περιορισμός $(c_i \prec n)$.

Προσθέτουμε επίσης ένα στάδιο αρχικοποίησης στο Büchi αυτόματο. Η αρχική κατάσταση του αυτομάτου δεν είναι η αρχική κατάσταση του χρονικού γράφου. Είναι μια καινούρια κατάσταση, από την οποία υπάρχει μια και μόνη εξέρχουσα ακμή προς την κατάσταση που αντιστοιχεί στην αρχική κατάσταση του χρονικού γράφου. Υποθέτουμε οτι αυτή η μετάβαση εκτελείται σε μηδενικό χρόνο. Προστείθεται και ένας επιπλέον μετρητής ο οποίος εξασφαλίζει ακριβώς το γεγονός οτι αυτή η μετάβαση παίρνει μηδέν χρόνο για να εκτελεστεί. Αυτός ο "μηδενικός μετρητής" είναι μια τεχνική λεπτομέρεια που στην πραγματικότητα δεν επηρεάζει καθόλου το γενικό μοντέλο. Με αυτήν την μετάβαση όλοι οι μετρητές του μοντέλου τίθενται στην τιμή που μετράνε. Το ζεύγος (B, T) που προκύπτει άπό την παραπάνω μετατροπή δίνεται στην εικόνα 5.7.

5.3.2 Ορισμός του μεταφραστικού σχήματος

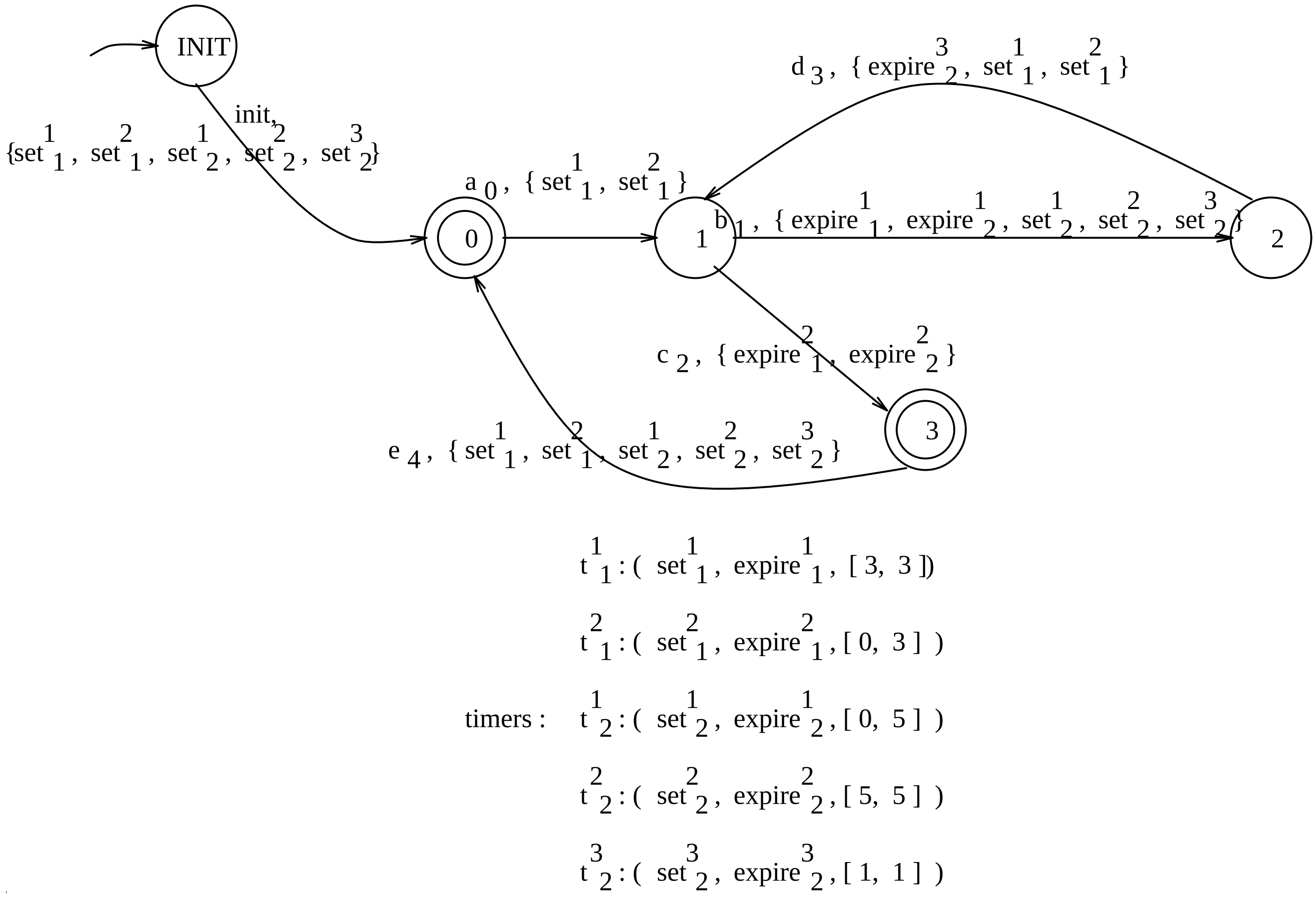
Σ' αυτή την ενότητα περιγράφουμε και τυπικά τον ορισμό του μεταφραστικού σχήματος. Ας θεωρήσουμε τον "περιορισμένο χρονικό γράφο" $G = (\Sigma, S, s_0, C, E, f)$.

Θέλουμε να αριθμήσουμε μοναδικά καθεμιά από τις ακμές του γράφου. Ετσι, ορίζουμε την συνάρτηση: $e : N \rightarrow E$, από το σύνολο N των ακεραίων αριθμών, στο σύνολο E των ακμών του γράφου. Ας σημειώνουμε τις ακμές σαν e_i αντί για $e(i)$. Ορίζουμε ακόμα το σύνολο $\hat{E} = \{e_i \mid e(i) \in E\}$.

Παρόμοια, αριθμούμε μοναδικά τις καταστάσεις του γράφου: Ορίζουμε τη συνάρτηση $s : N \rightarrow S$ από το σύνολο των ακεραίων στο σύνολο των καταστάσεων του γράφου. Γράφουμε s_i αντί για $s(i)$, και ορίζουμε το $\hat{S} = \{s_i \mid s(i) \in S\}$.

Θέλουμε ακόμα να μπορούμε να αναγνωρίζουμε μοναδικά τις ενέργειες του συστήματος. Εστω σ η ενέργεια του συστήματος πάνω στην ακμή e_i. Μετονομάζουμε αυτή την ενέργεια σε σ_i. Ετσι, έστω Σ το σύνολο των ενεργειών του γράφου G. Ορίζουμε το σύνολο: $\hat{\Sigma} = \{\sigma_i \mid \sigma \in \Sigma$ $e_i = (\_ , \sigma , \_ , \_ , \_) \in \hat{E}\}$

Ας σημειώνουμε με $\phi(c_k)$ τον περιορισμό της μορφής $(c_k \prec m)$ όπου $\prec$ είναι ένα από τα $\leq, <, \geq, >, =$. Κάθε περιορισμός $\phi(c_k)$ που εμφανίζεται στην ακμή e_i του γράφου μετονομάζεται σε $\phi_{i_j}(c_k)$ το οποίο σημαίνει οτι είναι ο $j-$οστός περιορισμός της μορφής $\phi(c_k)$ που εμφανίζεται στην $i-$οστή ακμή.



Σχήμα 5.7: Το ζεύγος (B, T) που προκύπτει

Εστω $C = \{c_1, c_2, \ldots, c_n\}$, το σύνολο των ρολογιών του γράφου G. Ορίζουμε το σύνολο T των μετρητών όπως ακολουθεί:
$T = \{t_k^{i_j}, \forall \phi_{i_j}(c_k), k = 1, \ldots, n\}$.

Εστω οτι το σύνολο $\mathcal{R}_i$ δηλώνει το σύνολο των ρολογιών που επανατίθενται στην ακμή e_i του γράφου G. Ορίζουμε το σύνολο $\hat{\mathcal{R}}_i = \{t_k^{i_j} \mid c_k \in \mathcal{R}_i \text{ και } t_k^{i_j} \in T\}$.

Τώρα που ορίσαμε όλα τα παραπάνω στοιχεία, είμαστε έτοιμοι να δώσουμε την κατασκευή του (B, T).

Το σύνολο T είναι ακριβώς το σύνολο που ορίσαμε παραπάνω. Σ' αυτό το σύνολο προσθέτουμε τον "μηδενικό μετρητή" t_{zero} του βήματος αρχικοποίησης, για τον οποίο μιλήσαμε στην προηγούμενη ενότητα. Ας σημειώνουμε με set_{zero} το γεγονός θέσης αυτού του μετρητή και με

$expire_{zero}$ το γεγονός εκπνοής του. Προφανώς το διάστημα τιμών του είναι $[0,0]$. Δεν αναφερόμαστε σε αυτόν τον μετρητή στους ορισμούς που ακολουθούν για λόγους απλότητας. Ολοι οι υπόλοιποι μετρητές ορίζονται σαν τριάδες της μορφής: $t_k^{i_j} = (set_k^{i_j}, expire_k^{i_j}, interval_k^{i_j})$, όπου $set_k^{i_j}$ και $expire_k^{i_j}$ τα γεγονότα τους και $interval_k^{i_j}$ τα διαστήματα τιμών τους που ορίζονται όπως παρακάτω:

$$interval_k^{i_j} = \begin{cases} [0,n] & \text{αν} \quad \phi_{i_j}(c_k) = (c_k \leq n) \\ [0,n) & \text{αν} \quad \phi_{i_j}(c_k) = (c_k < n) \\ [n,\propto) & \text{αν} \quad \phi_{i_j}(c_k) = (c_k \geq n) \\ (n,\propto] & \text{αν} \quad \phi_{i_j}(c_k) = (c_k > n) \\ [n,n] & \text{αν} \quad \phi_{i_j}(c_k) = (c_k = n) \end{cases}$$

Το Büchi αυτόματο ορίζεται σαν μια πεντάδα: $B = (\Sigma, S, S_0, E, F)$.

Το αλφάβητό του ορίζεται να είναι $\Sigma = \hat{\Sigma} \cup \{init\} \cup 2^{\{set_k^{i_j}, expire_k^{i_j} | \forall t_k^{i_j} \in T\}}$. Το σύνολο $\hat{\Sigma}$ το ορίσαμε λίγο παραπάνω. $init$ είναι μια επιπλέον ενέργεια για το βήμα αρχικοποίησης.

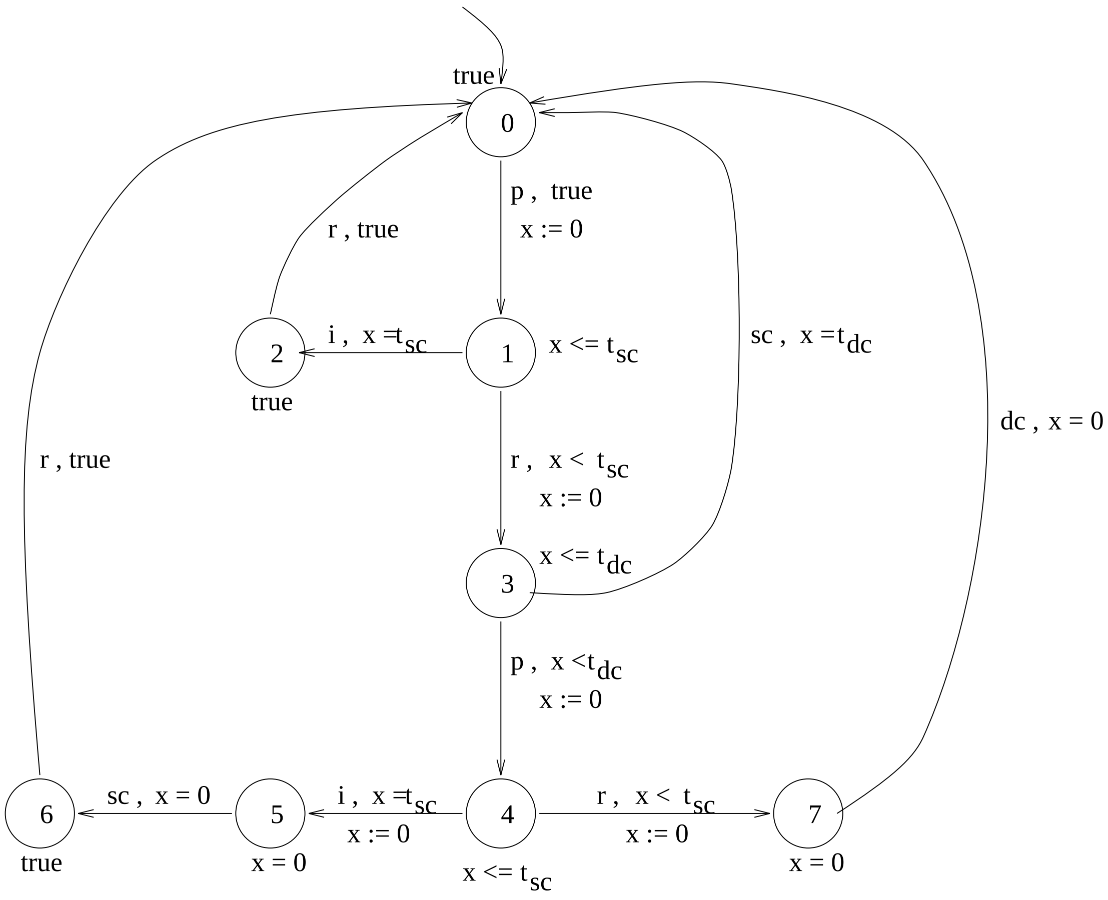
Το σύνολο των καταστάσεων του αυτομάτου είναι το σύνολο των καταστάσεων του χρονικού γράφου συν μια ακόμα που ονομάζεται $INIT$, η οποία είναι και η αρχική κατάσταση του αυτομάτου. Εχουμε λοιπόν $S = \hat{S} \cup \{INIT\}$. Το σύνολο των αρχικών καταστάσεών του είναι το μονοσύνολο $S_0 = \{INIT\}$.

Το σύνολο των ακμών του είναι το παρακάτω:
$E = \{(INIT, init \cup \{set_k^{i_j} \mid t_k^{i_j} \in T\}, s_0)\} \cup \{(s_k, \sigma_i \cup \{set_k^{i_j} \mid \forall t_k^{i_j} \in \hat{R}_i\} \cup \{expire_k^{i_j} \mid \forall t_k^{i_j} :$ $\phi_{i_j}(c_k)$ εμφανίζεται στην $e_i\}, s_l) \mid e_i = (s_k, \sigma, \phi, R_i, s_l) \in \hat{E}\}\}$.

Τελικά, το σύνολο των καταστάσεων αποδοχής του Büchi αυτομάτου είναι το: $F = \{s_i \mid s_i \in \hat{\Sigma} \wedge f(s_i) = true\}$.

5.4 Παράδειγμα

Ας θεωρήσουμε το σύστημα που αναγνωρίζει απλά και διπλά πατήματα ενός από τα κουμπιά της συσκευής του ποντικιού ενός υπολογιστή [Yov93]. Υπάρχουν δυο ειδών γεγονότα. Το πρώτο είναι το πάτημα του κουμπιού και το δεύτερο η απελευθέρωσή του. Το πάτημα του κουμπιού σημειώνεται με το γεγονός p και η απελευθέρωσή του με το r. Ενα "απλό πάτημα" συμβαίνει, όταν ο χρόνος που περνά μεταξύ του πατήματος του κουμπιού και της απελευθέρωσής του είναι μικρότερος από t_{sc}. Ενα "διπλό πάτημα" συμβαίνει, όταν δυο διαδοχικά απλά πατήματα απέχουν το ένα από το άλλο το πολύ t_{dc} χρονικές μονάδες.



Σχήμα 5.8: Ο χρονικός γράφος που περιγράφει την λειτουργία ενός κουμπιού του ποντικού ενός υπολογιστή

Ο χρονικός γράφος που περιγράφει το σύστημα δίνεται στην εικόνα 5.8. Το σύστημα ξεκινά από την κατάσταση 0 με το ρολόι x αρχικοποιημένο στο μηδέν. Στην κατάσταση 0 περιμένει για ένα γεγονός p. Οταν το γεγονός p συμβεί, το σύστημα κινείται στην κατάσταση 1 και το ρολόι x αρχίζει να μετρά. Στην κατάσταση 1 η τιμή του ρολογιού x δίνει το χρόνο που έχει περάσει από τη στιγμή που πρωτοπατήθηκε το κουμπί. Αν συμβεί ένα r σε χρόνο μικρότερο του t_{sc}, το σύστημα κινείται από την κατάσταση 1 στην 3. Το ρολόι x επανατίθεται μ' αυτήν τη μετάβαση. Αλλιώς, το σύστημα κινείται από την κατάσταση 1 στην 2, εκτελώντας μια εσωτερική ενέργεια i σε χρόνο t_{sc}. Στην κατάσταση 2 το σύστημα μπορεί να περιμένει μια ενέργεια r και τότε επιστρέφει πίσω στην κατάσταση 0.

Στην κατάσταση 3 η τιμή του ρολογιού δίνει τον χρόνο που έχει περάσει από την στιγμή που απελευθερώθηκε το κουμπί. Αν άλλο ένα γεγονός p συμβεί μέσα σε χρόνο t_{dc} από το τελευταίο r, το σύστημα κινείται από την 3 στην 4 και το x επανατίθεται στο μηδέν. Αν δεν συμβεί το p σε χρόνο μικρότερο του t_{dc} το σύστημα αναφέρει την εμφάνιση ενός απλού πατήματος του κουμπιού και κινείται από την 3 πίσω στην 0.

Στην 4 το σύστημα περιμένει ξανά για ένα γεγονός r σε χρόνο μικρότερο του t_{sc} μετά το τελευταίο p. Αν r συμβαίνει μέσα σ' αυτόν το χρόνο, το σύστημα κινείται στην κατάσταση 7 απ' όπου αμέσως αναφέρει την εμφάνιση ενός διπλού πατήματος και κινείται πίσω στην κατάσταση 0. Αν το r δεν συμβεί μέσα στον προκαθορισμένο χρόνο, το σύστημα κινείται από την κατάσταση 4 στην κατάσταση 5 εκτελώντας μια εσωτερική ενέργεια i. Από την 5 κινείται στην 6 αναφέροντας ένα απλό πάτημα του κουμπιού. Στην 6, περιμένει για ένα r γεγονός και κινείται πίσω στην 0.

Οπως μπορούμε να παρατηρήσουμε, ο χρονικός γράφος που περιγράφει το κουμπί ενός ποντικιού, είναι ένας περιορισμένος χρονικός γράφος. Ετσι μπορούμε να εφαρμόσουμε το μεταφραστικό μας σχήμα για να κατασκευάσουμε το ισοδύναμο (B, T) που περιγράφει το παραπάνω σύστημα.

Αρχικά, δίνουμε ένα μοναδικό αναγνωριστικό στις ακμές του χρονικού γράφου. Το σύνολο $\hat{E}$ ορίζεται αμέσως παρακάτω:

$$\begin{aligned}
\hat{E} = \{ \quad & e_0 = (0, p, true, \{x\}, 1), \\
& e_1 = (1, i, x = t_{sc}, \emptyset, 2), \\
& e_2 = (1, r, x < t_{sc}, \{x\}, 3), \\
& e_3 = (2, r, true, \emptyset, 0), \\
& e_4 = (3, p, x < t_{dc}, \{x\}, 4), \\
& e_5 = (3, sc, x = t_{dc}, \emptyset, 0), \\
& e_6 = (4, i, x = t_{sc}, \{x\}, 5), \\
& e_7 = (4, r, x < t_{sc}, \{x\}, 7), \\
& e_8 = (5, sc, x = 0, \emptyset, 6), \\
& e_9 = (6, r, true, \emptyset, 0), \\
& e_{10} = (7, dc, x = 0, \emptyset, 0)\}
\end{aligned}$$

Κάθε σύνολο $\mathcal{R}_i$ περιέχει τα ρολόγια που επανατίθενται στην ακμή e_i.

$$
\begin{aligned}
\mathcal{R}_0 &= \{x\} \\
\mathcal{R}_1 &= \emptyset \\
\mathcal{R}_2 &= \{x\} \\
\mathcal{R}_3 &= \emptyset \\
\mathcal{R}_4 &= \{x\} \\
\mathcal{R}_5 &= \emptyset \\
\mathcal{R}_6 &= \{x\} \\
\mathcal{R}_7 &= \{x\} \\
\mathcal{R}_8 &= \emptyset \\
\mathcal{R}_9 &= \emptyset \\
\mathcal{R}_{10} &= \emptyset
\end{aligned}
$$

Δίνουμε ένα μοναδικό αναγνωριστικό και στις καταστάσεις του γράφου. Το σύνολο $\hat{S}$ ορίζεται όπως ακολουθεί: $\hat{S} = \{0, 1, 2, 3, 4, 5, 6, 7\}$

Ορίζουμε και το σύνολο $\hat{\Sigma}$ να είναι το σύνολο των ενεργειών σ_i τέτοιο ώστε σ είναι το γεγονός που συμβαίνει στην ακμή e_i του χρονικού γράφου. $\hat{\Sigma} = \{p_0, i_1, r_2, r_3, p_4, sc_5, i_6, r_7, sc_8, r_9, dc_{10}\}$

Οι φύλακες των ακμών μετονομάζονται. Αν ο φύλακας μιας ακμής είναι ίσος με $true$ τότε δεν μετονομάζεται. Κάθε άλλος περιορισμός της μορφής $(x \prec n)$ που εμφανίζεται στις ακμές του γράφου μετονομάζεται όπως φαίνεται παρακάτω:

$$
\begin{aligned}
\phi_{1_0}(x) &= (x = t_{sc}) \\
\phi_{2_0}(x) &= (x < t_{sc}) \\
\phi_{4_0}(x) &= (x < t_{dc}) \\
\phi_{5_0}(x) &= (x = t_{dc}) \\
\phi_{6_0}(x) &= (x = t_{sc}) \\
\phi_{7_0}(x) &= (x < t_{sc}) \\
\phi_{8_0}(x) &= (x = 0) \\
\phi_{10_0}(x) &= (x = 0)
\end{aligned}
$$

Ορίζουμε και το σύνολο των μετρητών T να είναι:

$$T = \{t^{1_0}, t^{2_0}, t^{4_0}, t^{5_0}, t^{6_0}, t^{7_0}, t^{8_0}, t^{10_0}\}$$

Τέλος ορίζουμε τα σύνολα $\hat{\mathcal{R}}_i$:

$$
\begin{aligned}
\hat{\mathcal{R}}_0 &= \{t^{1_0}, t^{2_0}, t^{4_0}, t^{5_0}, t^{6_0}, t^{7_0}, t^{8_0}, t^{10_0}\} \\
\hat{\mathcal{R}}_1 &= \emptyset \\
\hat{\mathcal{R}}_2 &= \{t^{1_0}, t^{2_0}, t^{4_0}, t^{5_0}, t^{6_0}, t^{7_0}, t^{8_0}, t^{10_0}\} \\
\hat{\mathcal{R}}_3 &= \emptyset \\
\hat{\mathcal{R}}_4 &= \{t^{1_0}, t^{2_0}, t^{4_0}, t^{5_0}, t^{6_0}, t^{7_0}, t^{8_0}, t^{10_0}\} \\
\hat{\mathcal{R}}_5 &= \emptyset \\
\hat{\mathcal{R}}_6 &= \{t^{1_0}, t^{2_0}, t^{4_0}, t^{5_0}, t^{6_0}, t^{7_0}, t^{8_0}, t^{10_0}\} \\
\hat{\mathcal{R}}_7 &= \{t^{1_0}, t^{2_0}, t^{4_0}, t^{5_0}, t^{6_0}, t^{7_0}, t^{8_0}, t^{10_0}\} \\
\hat{\mathcal{R}}_8 &= \emptyset \\
\hat{\mathcal{R}}_9 &= \emptyset \\
\hat{\mathcal{R}}_{10} &= \emptyset
\end{aligned}
$$

Τώρα είμαστε έτοιμοι να δώσουμε το (B, T) για το κουμπί του ποντικιού. Το Büchi αυτόματο, περιγράφεται στην εικόνα 5.9. Τα σύνολα $\mathcal{B}_i$ που εμφανίζονται στις ακμές του Büchi, αντιστοιχούν στα γεγονότα μετρητών που συμβαίνουν σε κάθε ακμή. Ετσι λοιπόν έχουμε:

$\mathcal{B}_{init} = \{set^{1_0}, set^{2_0}, set^{4_0}, set^{5_0}, set^{6_0}, set^{7_0}, set^{8_0}, set^{10_0}\}$

$\mathcal{B}_0 = \{set^{1_0}, set^{2_0}, set^{4_0}, set^{5_0}, set^{6_0}, set^{7_0}, set^{8_0}, set^{10_0}\}$

$\mathcal{B}_1 = \{expire^{1_0}\}$

$\mathcal{B}_2 = \{expire^{2_0}, set^{1_0}, set^{2_0}, set^{4_0}, set^{5_0}, set^{6_0}, set^{7_0}, set^{8_0}, set^{10_0}\}$

$\mathcal{B}_3 = \emptyset$

$\mathcal{B}_4 = \{expire^{4_0}, set^{1_0}, set^{2_0}, set^{4_0}, set^{5_0}, set^{6_0}, set^{7_0}, set^{8_0}, set^{10_0}\}$

$\mathcal{B}_5 = \{expire^{5_0}\}$

$\mathcal{B}_6 = \{expire^{6_0}, set^{1_0}, set^{2_0}, set^{4_0}, set^{5_0}, set^{6_0}, set^{7_0}, set^{8_0}, set^{10_0}\}$

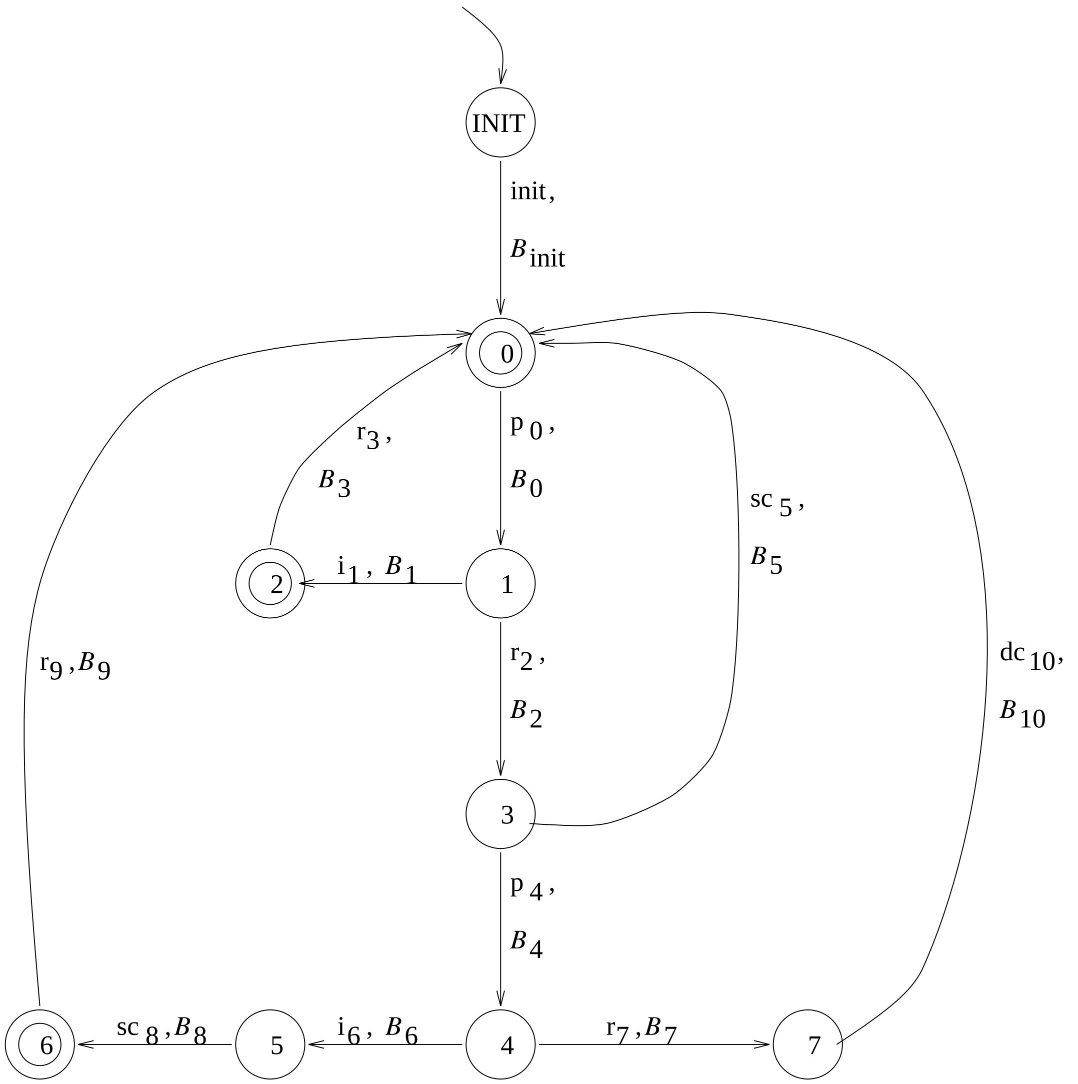
$\mathcal{B}_7 = \{expire^{7_0}, set^{1_0}, set^{2_0}, set^{4_0}, set^{5_0}, set^{6_0}, set^{7_0}, set^{8_0}, set^{10_0}\}$

$\mathcal{B}_8 = \{expire^{8_0}\}$

$\mathcal{B}_9 = \emptyset$

$\mathcal{B}_{10} = \{expire^{10_0}\}$

Τελικά, το σύνολο των μετρητών του συστήματος δίνεται στην εικόνα 5.10.



Σχήμα 5.9: Το Büchi αυτόματο B του ζεύγους (B, T) που περιγράφει το κουμπί του ποντικιού

5.5 Υλοποίηση

Αναπτύξαμε έναν μεταφραστή που υλοποιεί το μεταφραστικό σχήμα που περιγράψαμε παραπάνω. Η υλοποίηση έγινε σε δυο βήματα.

Το πρώτο βήμα αποτελείται από την συντακτική και σημασιολογική ανάλυση του δοσμένου χρονικού γράφου. Για την συντακτική και σημασιολογική ανάλυση, χρησιμοποιήθηκαν τα εργαλεία Lex και Yacc [AHU86]. Το αποτέλεσμα αυτής της φάσης της ανάλυσης είναι μια ενδιάμεση μορφή αναπαράστασης του χρονικού γράφου.

Το δεύτερο βήμα αποτελείται από την παραγωγή του συνόλου των μετρητών T, και του Büchi αυτομάτου B του ζεύγους (B, T) από την ενδιάμεση αναπαράσταση του χρονικού γράφου.

timers :

$t^{1}_{0} : \{ set^{1}_{0},\ expire^{1}_{0},\ [t_{sc},\ t_{sc}] \}$

$t^{2}_{0} : \{ set^{2}_{0},\ expire^{2}_{0},\ ,[0,\ t_{sc}) \}$

$t^{4}_{0} : \{ set^{4}_{0},\ expire^{4}_{0},\ [0,\ t_{sc}) \}$

$t^{5}_{0} : \{ set^{5}_{0},\ expire^{5}_{0},\ [t_{dc},\ t_{dc}] \}$

$t^{6}_{0} : \{ set^{6}_{0},\ expire^{6}_{0},\ [t_{sc},\ t_{sc}]\}$

$t^{7}_{0} : \{ set^{7}_{0},\ expire^{7}_{0},\ [0,\ t_{sc}) \}$

$t^{8}_{0} : \{ set^{8}_{0},\ expire^{8}_{0},\ [0,\ 0] \}$

$t^{10}_{0} : \{ set^{10}_{0},\ expire^{10}_{0},\ [0.\ 0] \}$

Σχήμα 5.10: Το σύνολο των μετρητών T του ζεύγους (B, T) που περιγράφει το κουμπί του ποντικιού

Στο μοντέλο που προτείναμε στην ενότητα 5.1 οι μετρητές ορίζονται σαν τριάδες της μορφής $(set, expire, interval)$, όπου set και $expire$ είναι τα γεγονότα μετρητών. Οπως έχουμε ήδη αναφέρει, αυτά τα γεγονότα εμφανίζονται πάνω στις ακμές του γράφου μαζί με τα υπόλοιπα γεγονότα του συστήματος.

Στην υλοποίηση έχουμε αλλάξει λίγο την επικοινωνία μεταξύ των μετρητών και του αυτομάτου. Αυτή η αλλαγή χρησιμοποιείται και στο [Tzo92]. Οι μετρητές υλοποιούνται σαν λογικά στοιχεία που έχουν μια “συνθήκη θέσης”, μια “συνθήκη εκπνοής” και ένα “ορισμό διαστήματος”. Οι συνθήκες θέσης και εκπνοής είναι λογικές εκφράσεις πάνω στις ενέργειες του αυτομάτου. Αντί λοιπόν να δημιουργούμε για κάθε μετρητή τα γεγονότα set και $expire$ και μετά να τα αντιστοιχούμε στις ακμές του αυτομάτου, δημιουργούμε το αυτομάτο με τις ακμές του σημειωμένες μόνο με τα γεγονότα του συστήματος. Οι μετρητές τίθενται ή εκπνέουν, ανάλογα με το αν οι συνθήκες θέσης ή εκπνοής γίνονται αληθείς αντιστοίχως.

Παρακάτω παρουσιάζουμε τις βασικές δομές δεδομένων που χρησιμοποιούνται για την υλοποίηση του μεταφραστικού σχήματος.

5.5.1 Δομές Δεδομένων

```
/* Uniquely identified edges */
struct edgeinfo {
    int fromstate;
    int tostate;
    char *action;
    struct constrlist *constrptr;
};

/* The clocks of the timed graph */
struct clckinfo {
    char *clckid;
    struct resetlist *resetptr;
};

/* Form of constraints */
struct constrlist {
    char *clckid;
    char *symbol;
    int bound;
    struct constrlist *next;
};

/* The set of reset timers on the edge edgeid */
struct resetlist {
    int edgeid;
    struct resetlist *next;
};

/* The set of states of the timed graph that corresponds to the
   accepting states of the Buchi */
struct cyset {
```

```
    int state;
    struct cyset *next;
};
```

Κατασκευάζουμε δυο πίνακες. Ο πρώτος αντιστοιχεί στο σύνολο των ακμών και ο δεύτερος αντιστοιχεί στο σύνολο των ρολογιών του χρονικού γράφου. Κάθε στοιχείο του πρώτου πίνακα είναι μια δομή τύπου *edgeinfo*. Σε μια τέτοιου είδους δομή κρατάμε όλη τη σχετική πληροφορία μιας ακμής του χρονικού γράφου. Κάθε ακμή αναγνωρίζεται μοναδικά από τη θέση της μέσα σ' αυτόν τον πίνακα. Το πεδίο *fromstate* κρατά το αναγνωριστικό της κατάστασης από την οποία εξέρχεται αυτή η ακμή, και το πεδίο *tostate* κρατά τον αριθμό της κατάστασης στην οποία εισέρχεται αυτή η ακμή. Το πεδίο *action* κρατά το όνομα της ενέργειας που εκτελεί το σύστημα όταν παίρνει αυτή την ακμή, και το *constrptr* είναι ένας δείκτης σε μια λίστα που κρατά όλους τους περιορισμούς που εμφανίζονται σε αυτήν την ακμή. Δίνουμε ένα μοναδικό όνομα στην ενέργεια αυτής της ακμής με το να επεκτείνουμε το όνομά της με τον αναγνωριστικό αριθμό αυτής της ακμής.

Κάθε περιορισμός της μορφής $(c \prec n)$ ο οποίος εμφανίζεται στην ακμή, προστείθεται στη λίστα που τα στοιχεία της είναι του τύπου *constrlist*.

Η δομή *clckinfo* κρατά πληροφορία για κάθε ρολόι. Η πληροφορία αυτή, απαρτίζεται από τα εξής στοιχεία: το όνομα του ρολογιού (*clckid*) και την λίστα των ακμών στις οποίες το συγκεκριμένο ρολόι επανατείθεται. Ο δείκτης *resetptr* δείχνει σ' αυτήν την λίστα.

5.5.2 Η κατασκευή του (B, T)

Σ' αυτήν την ενότητα παρουσιάζουμε τον αλγόριθμο για την κατασκευή του (B, T). Ας υποθέσουμε οτι ο δοσμένος χρονικός γράφος είναι ο $G = (\Sigma, S, s_0, C, E, f)$. Ας υποθέσουμε ακόμα οτι έχουμε ήδη αριθμήσει μοναδικά τις καταστάσεις και τις ακμές του γράφου μέσω των συναρτήσεων s και e αντίστοιχα. Ετσι το σύνολο των καταστάσεών του είναι $\hat{S}$ και το σύνολο των ακμών του είναι $\hat{E}$. Δοσμένης ακόμα της αλλαγής στον ορισμό των μετρητών, που περιγράψαμε παραπάνω, είμαστε έτοιμοι να δώσουμε την κατασκεύη του ζεύγους (B, T).

Ας σημειώνουμε με S_B, E_B και F το σύνολο των καταστάσεων, το σύνολο των ακμών και το σύνολο των καταστάσεων αποδοχής του Büchi αντίστοιχα. Η κατασκευή του (B, T) δίνεται από τον ακόλουθο αλγόριθμο:

```
begin
    S_B := {INIT}
    E_B := {(INIT, init, s_0)}
    F = ∅
    T = ∅
    forall s_i ∈ Ŝ
        if (f(s_i) = true) then
            F = F ∪ s_i
        S_B := S_B ∪ s_i
        forall e_i ∈ E_i : e_i = (s_m, σ, φ, R_i, s_n)
            constrlist_{e_i} := create_constrlist(φ)
            forall φ_k^{i_j} ∈ constrlist_{e_i}
                t_k^{i_j} := create_timer(φ_k^{i_j})
                T := T ∪ t_k^{i_j}
            end
            forall c_k ∈ R_i
                resetlist_{c_k} = resetlist_{c_k} ∪ e_i
            e := create_edge(s_m, σ_i, s_n)
            end
            E_B := E_B ∪ e
        end
    end
    forall t_k^{i_j} ∈ T
        forall e_l ∈ resetlist_{c_k}
            set_cond(t_k^{i_j}) := set_cond(t_k^{i_j}) + σ_l
        end
    end
end
```

όπου η συνάρτηση $create\_timer()$ είναι πολύ απλή και δίνεται παρακάτω:

```
create_timer (φ_k^{i_j})
        begin
            set_cond(t_k^{i_j}) := init
```

$$expire\_cond(t_k^{i_j}) := \sigma_i$$
$$interval(t_k^{i_j}) := create\_interval(\phi_k^{i_j})$$

end

5.6 Σύνοψη του κεφαλαίου

Το βασικό πρόβλημα στην περιοχή της αυτόματης επαλήθευσης είναι να μπορούμε να απαντήσουμε στην ερώτηση αν ένα σύστημα πραγματικού χρόνου ικανοποιεί μια ιδιότητα. Στο κεφάλαιο 4 περιγράψαμε δυο διαφορετικούς αλγορίθμους που απαντάνε σ' αυτήν την ερώτηση. Ο πρώτος απαντά στην παραπάνω ερώτηση λύνοντας το πρόβλημα έγκλεισης γλωσσών αυτομάτων. Η είσοδος του αλγορίθμου είναι δυο (B, T)s. Το πρώτο παριστάνει το σύστημα και το δεύτερο την ιδιότητα.

Ο δεύτερος αλγόριθμος είναι ένας αλγόριθμος ελέγχου μοντέλου. Παίρνει σαν είσοδο έναν χρονικό γράφο που περιγράφει το σύστημα και μια έκφραση TCTL που περιγράφει την ιδιότητα.

Σ' αυτό το κεφάλαιο παρουσιάζουμε μια μέθοδο για την κατασκεύη ενός (B, T) από ένα περιορισμένο χρονικό γράφο. Σαν αποτέλεσμα, μπορούμε να έχουμε μια μοναδική αναπαράσταση του συστήματος, σαν ένα χρονικό γράφο, και τη δυνατότητα να εφαρμόζουμε δυο αλγόριθμους επαλήθευσης για την απόδειξη ιδιοτήτων του συστήματος.

Κεφάλαιο 6

Εργαλεία επαλήθευσης

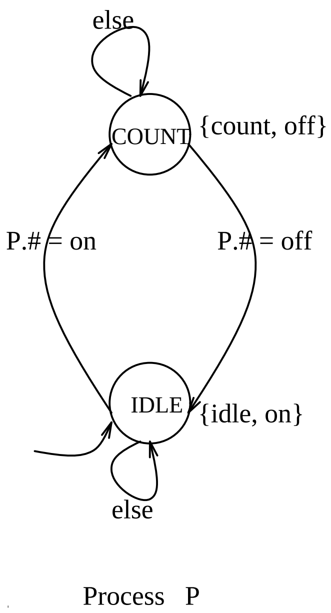
Σ' αυτό το κεφάλαιο θα παρουσιάσουμε δυο εργαλεία για αυτόματη επαλήθευση συστημάτων πραγματικού χρόνου που υιοθετούν τις τεχνικές που περιγράψαμε στα προηγούμενα κεφάλαια και έχουν αναπτυχθεί τα τελευταία χρόνια.

6.1 Το εργαλείο επαλήθευσης RT-COSPAN

Το RT-COSPAN είναι ένα εργαλείο για αυτόματη επαλήθευση συστημάτων πραγματικού χρόνου [CDT92, CDCT92, Tzo92]. Είναι μια επέκταση του COSPAN [HK90, KH87] το οποίο είναι ένα πολύ γνωστό εργαλείο επαλήθευσης σύγχρονων συστημάτων που όμως δεν περιέχουν πληροφορία πραγματικού χρόνου. Ο χρόνος σε αυτό το εργαλείο θεωρείται διακριτός. Το RT-COSPAN είναι δομημένο σε κάμποσα επίπεδα. Για να παρουσιάσουμε τον βασικό κορμό του εργαλείου, πρέπει να συζητήσουμε κάθε επίπεδο χωριστά. Πρώτα πρέπει να μιλήσουμε για το υποκείμενο μαθηματικό μοντέλο, που ονομάζεται μοντέλο "Επιλογής/Απόφασης" (*Selection/Resolution* model) [KK86, GK80]. Κατόπιν θα παρουσιάσουμε την αρχιτεκτονική του εργαλείου COSPAN που είναι βασισμένη στο μοντέλο που αναφέραμε παραπάνω. Τελικά θα περιγράψουμε πως το COSPAN έχει επεκταθεί στο RT-COSPAN.

6.1.1 Μοντέλο Επιλογής/Απόφασης

Το μοντέλο "Επιλογής/Απόφασης" (*Selection/Resolution* ή S/R μοντέλο), είναι ένα μαθηματικό μοντέλο για σύγχρονες διεργασίες, όπου ένα πολύπλοκο σύστημα μπορεί να περιγραφεί σαν ένα σύνολο από συζευγμένες πεπερασμένες μηχανές. Ενα γνώρισμα του S/R μοντέλου που



Σχήμα 6.1: Παράδειγμα μιας S/R διεργασίας που περιγράφει έναν διακόπτη δυο καταστάσεων

το κάνει ιδιαίτερα ελκυστικό είναι οτι η σύζευξη μεταξύ πεπερασμένων μηχανών εκφράζεται με λογικά κατηγορήματα πάνω σε σήματα επικοινωνίας των μηχανών. Το S/R μοντέλο υιοθετεί την σημασιολογία ιχνών διακριτού χρόνου. Ενας λογισμός ορίζεται πάνω στις διεργασίες που μπορούν να συντεθούν χρησιμοποιώντας είτε σύγχρονο είτε ασύγχρονο γινόμενο. Το μοντέλο περιέχει συνθήκες αποδοχής για να εκφράζει οποιαδήποτε επιθυμητή ω-κανονική ιδιότητα του συστήματος. Η επαλήθευση επιτυγχάνεται με το να ελέγχουμε για έγκλειση γλωσσών.

Ενα σύστημα που περιγράφεται στο S/R μοντέλο αποτελείται από ένα πεπερασμένο σύνολο από "διεργασίες". Κάθε διεργασία μπορεί να περιγραφεί σχηματικά με ένα σύστημα μεταβάσεων με ετικέττες, όπως αυτό της εικόνας 6.1. Οι διεργασίες μπορούν να θεωρηθούν σαν μηχανές πεπερασμένων καταστάσεων. Καθεμιά, έχει μια "μεταβλητή κατάστασης", οι τιμές της οποίας είναι οι διαφορετικές καταστάσεις της διεργασίας. Οι καταστάσεις μιας διεργασίας αναπαριστώνται με τις κορυφές του συστήματος μεταβάσεων. Το διάνυσμα που αποτελείται από την τρέχουσα, κάθε φορά, κατάσταση κάθε διεργασίας ονομάζεται "ολική κατάσταση" του συστήματος.

Κάθε διεργασία έχει επίσης μια "μεταβλητή επιλογής" η οποία παίρνει την τιμή της από ένα πεπερασμένο σύνολο πιθανών τιμών, που ονομάζονται "επιλογές". Οι ετικέττες των ακμών είναι λογικά κατηγορήματα πάνω στις επιλογές του γράφου που περιγράφει την διεργασία. Οπως αναφέραμε και προηγουμένως, οι διεργασίες μπορούν να θεωρηθούν μηχανές πεπερασμένων καταστάσεων. Καθεμιά τέτοια μηχανή παράγει μια γλώσσα η οποία αποτελείται από άπειρες σε μήκος ακολουθίες των τιμών των επιλογών της. Οι "επιλογές" αντιστοιχούν στα σήματα επικοινωνίας που αναφέραμε παραπάνω, μέσω των οποίων μπορούν και επικοινωνούν οι διεργασίες. Κάθε διεργασία μπορεί να διαβάζει τις επιλογές όλων των άλλων διεργασιών αλλά μπορεί να τροποποιήσει μόνο την δική της μεταβλητή. Το διάνυσμα από τις τρέχουσες επιλογές όλων των διεργασιών ονομάζεται "ολική επιλογή".

Κάθε ακμή του γράφου αντιστοιχεί σε μια μετάβαση της διεργασίας από μια κατάσταση σε μια άλλη, που πραγματοποιείται σε ένα υπολογιστικό βήμα. Ενα υπολογιστικό βήμα, του μοντέλου "Επιλογής/Απόφασης" αποτελείται από δυο φάσεις. Η πρώτη, είναι η φάση της "επιλογής", όπου κάθε διεργασία αναθέτει μια τιμή στην μεταβλητή επιλογής της. Στο τέλος αυτής της φάσης η ολική επιλογή του συστήματος έχει αποφασισθεί. Η φάση της "επιλογής" ακολουθείται από τη φάση της "απόφασης", στην οποία, κάθε διεργασία αποφασίζει την επόμενη κατάστασή της ανάλογα με το ποιές μεταβάσεις ενεργοποιούνται. Αν ενεργοποιούνται περισσότερες από μια μεταβάσεις ταυτόχρονα, η διεργασία αποφασίζει την επόμενη κατάστασή της μη αιτιοκρατικά.

Οπως είναι φανερό, οι S/R διεργασίες είναι στην ουσία μια επέκταση των Büchi αυτομάτων. Για να αποδείξουμε την ορθότητα ενός συστήματος που περιγράφεται με το S/R μοντέλο, χρειάζεται να να χρησιμοποιήσουμε μια κατάλληλη γλώσσα για την περιγραφή των ιδιοτήτων. Χρησιμοποιούμε την μέθοδο περιγραφής με αυτόματα, που αναπτύξαμε στο κεφάλαιο 3. Το συμπλήρωμα της επιθυμητής ιδιότητας περιγράφεται και αυτό σαν μια S/R διεργασία, που ονομάζεται "ελεγκτής" (*monitor*). Το πρόβλημα της επαλήθευσης λύνεται με το απαντήσουμε, κατά πόσο η τομή των γλωσσών του συστήματος και του ελεγκτή είναι η κενή γλώσσα. Για περισσότερες λεπτομέρειες, [KK86, GK80].

6.1.2 Το εργαλείο επαλήθευσης COSPAN

Το COSPAN είναι ένα πολύ γνωστό εργαλείο για την περιγραφή και επαλήθευση συστημάτων και βασίζεται στο μοντέλο"Επιλογής/Απόφασης". Η λογική ανάλυση στο COSPAN γίνεται με συμβολικό έλεγχο του συστήματος και της επιθυμητής συμπεριφοράς που έχει περιγράψει ο χρήστης. Το σύστημα ούτε προσομοιώνεται, ούτε εκτελείται, αλλά κατασκευάζεται μια μαθηματική απόδειξη πάνω στην περιγεγραμμένη συμπεριφορά. Τυπικές μέθοδοι ελαχιστοποίησης χρησιμοποιούνται από τους αλγορίθμους του COSPAN για να αντιμετωπίζουν τον μεγάλο αριθμό καταστάσεων που προκύπτει από την ανάλυση σύγχρονων συστημάτων.

```
proc    P
        selvar    # : (idle, on, count, off)
        stvar     $ : (IDLE, COUNT)
        init      IDLE
        trans
        IDLE      {idle, on}
               -> COUNT   : # = on
               -> IDLE    : else;

        COUNT   {count, off}
               -> IDLE    : # = off
               -> COUNT   : else;
end
```

Σχήμα 6.2: Ενα πρόγραμμα S/R που περιγράφει τον διακόπτη δυο καταστάσεων

Η γλώσσα περιγραφής του COSPAN

Η γλώσσα του COSPAN είναι μια γλώσσα που μας βοηθά να περιγράφουμε *S/R* διεργασίες και ελεγκτές. Στην εικόνα 6.2 παρουσιάζουμε πώς περιγράφεται ο διακόπτης της εικόνας 6.1 στην γλώσσα του COSPAN.

Το βασικό στοιχείο της γλώσσας είναι η "διεργασία". Η περιγραφή μιας διεργασίας ξεκινά με την προκαθορισμένη λέξη **proc** ή **monitor** για τις διεργασίες ελεγκτές, και ακολουθείται από το όνομα της διεργασίας. Πάντα τελειώνει με την προκαθορισμένη λέξη **end**. Σχόλια μπορούν να δωθούν ανάμεσα στους χαρακτήρες /* και */. Οι προκαθορισμένες λέξεις **selvar** και **statevar** χρησιμοποιούνται για τον ορισμό του ονόματος και του πεδίου ορισμού των μεταβλητών επιλογής και κατάστασης αντίστοιχα. Το όνομα που συνήθως χρησιμοποιείται για την μεταβλητή επιλογής είναι # και για την μεταβλητή κατάστασης $. Το **import** χρησιμοποιείται για να δηλώνει μια λίστα από διεργασίες, οι επιλογές και/ή καταστάσεις των οποίων χρησιμοποιούνται σαν είσοδοι της συγκεκριμένης διεργασίας. Η προκαθορισμένη λέξη **init** δηλώνει το σύνολο των αρχικών καταστάσεων της διεργασίας. Η **trans** δηλώνει την αρχή του τμήματος ορισμού των κανόνων μεταβάσεων, δηλαδή των κανόνων κάτω από τους οποίους η διεργασία κινείται από μια κατάσταση σε μια άλλη. Το τμήμα ορισμού των κανόνων μεταβάσεων έχει την ακόλουθη μορφή:

$$\begin{array}{rlll} current\_state & & \{selection\_list\} & \\ & -> & next\_state & : condition \\ & & \ldots & \\ & -> & next\_state & : condition; \end{array}$$

Η "συνθήκη" είναι ένα πιθανό λογικό κατηγόρημα πάνω στην τρέχουσα τιμή της μεταβλητής επιλογής της διεργασίας και των τιμών των μεταβλητών επιλογής των διεργασιών που ανήκουν στην λίστα **import**. Ακόμα η λέξη **else** χρησιμοποιείται για να δηλώνει την "άρνηση" της λογικής διάζευξης των συνθηκών που προηγούνται, στο τμήμα ορισμού των μεταβάσεων.

Οι συνθήκες αποδοχής προστείθενται σε μια διεργασία με κατάλληλο ορισμό των συνόλων **cyset** *state list*, όπου *state list* είναι ένα σύνολο καταστάσεων της διεργασίας.

Τελικά ο τύπος **proctype** υποστηρίζεται, για τον ορισμό παραμετρικών διεργασιών. Εχει δε, την ακόλουθη μορφή:

$$\begin{array}{ll} proctype & type\_name(formal\_parameters) \\ & \ldots \\ end & \end{array}$$

Μια παραμετρική διεργασία ορίζεται σαν:

proc proc_name(actual parameters)

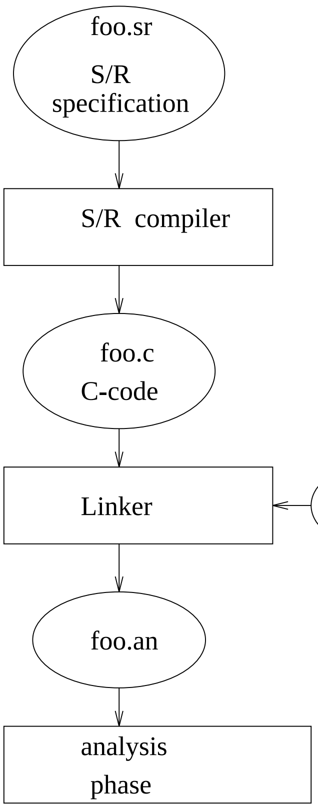
και ένας πίνακας από παραμετρικές διεργασίες σαν:

proc proc_name[index_range] : type_name(actual_parame ters)

Η αρχιτεκτονική του COSPAN

Η αρχιτεκτονική του COSPAN φαίνεται στην εικόνα 6.3.

Η είσοδος του COSPAN είναι ένα αρχείο S/R, που για παράδειγμα ονομάζεται **foo.sr**, και το οποίο περιέχει την περιγραφή ενός συστήματος στην γλώσσα περιγραφής του SR. Στο πρώτο βήμα η S/R περιγραφή του συστήματος μεταφράζεται σε ένα πρόγραμμα C και φυλάσσεται σε ένα αρχείο που ονομάζεται **foo.c**. Ο μεταφραστής S/R είναι υπεύθυνος για την εκτέλεση αυτής της εργασίας.



Σχήμα 6.3: Η αρχιτεκτονική του εργαλείου COSPAN

Επειτα, ο C κώδικας που παράγεται ενώνεται με μια βιβλιοθήκη που ονομάζεται **libsr.a** για να παράγει εκτελέσιμο κώδικα στο αρχείο **foo.an**. Αυτή η βιβλιοθήκη περιέχει τις ρουτίνες που στην ουσία υλοποιούν τους αλγορίθμους ανάλυσης του συστήματος.

Τελικά στην φάση της ανάλυσης, ο κώδικας του **foo.an** εκτελείται. Για λόγους απόδοσης, η ανάλυση του συστήματος γίνεται δυναμικά ("on the fly"), το οποίο σημαίνει οτι ο ολικός γράφος που αναπαριστά το σύστημα κατασκευάζεται μια-μια κατάσταση και ταυτόχρονα αναλύεται. Εκτεταμένες λεπτομέρειες για την λειτουργία του COSPAN μπορούν να βρεθούν στα [KH87, HK90].

6.1.3 Η επέκταση του COSPAN στο RT-COSPAN

Το RT-COSPAN είναι μια επέκταση του COSPAN. Το RT-COSPAN έχει υλοποιηθεί χωρίς καμιά αλλαγή στα βασικά χαρακτηριστικά και της λειτουργίας του COSPAN. Η επέκταση στηρίζεται στην υλοποίηση του μοντέλου των *Set/Expire* χρονικών γράφων που περιγράψαμε

```
proc    t1  :  R_TIMER( \[, 2, 2, \],  P.# = on,   P.# = off)

    proc     P
            selvar       # : (idle, on, count, off)
            stvar        $ : (IDLE, COUNT)
            init         IDLE
            trans
            IDLE         {idle, on}
                    -> COUNT   : # = on
                    -> IDLE      : else;

            COUNT    {count, off}
                    -> IDLE      : # = off
                    -> COUNT    : else;
    end
```

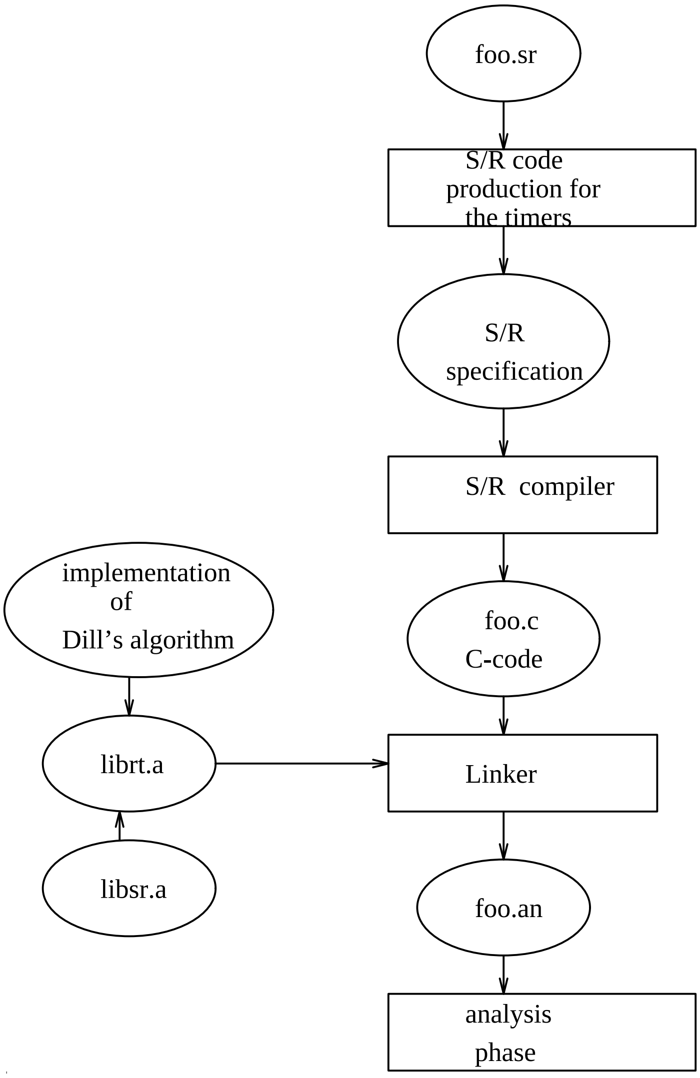
Σχήμα 6.4: Ενα παράδειγμα μιας S/R διεργασίας μαζί με έναν μετρητή που περιγράφει ένα διακόπτη δυο καταστάσεων

στο κεφάλαιο 4.

Η γλώσσα περιγραφής του RT-COSPAN

Η γλώσσα περιγραφής του RT-COSPAN παρουσιάζεται με ένα παράδειγμα στην εικόνα 6.4. Η παραπάνω εικόνα δείχνει έναν διακόπτη που μπορεί να είναι ενεργός μόνο για 2 χρονικές μονάδες. Η μόνη διαφορά μεταξύ της γλώσσας του RT-COSPAN και του COSPAN είναι οι δηλώσεις των μετρητών. Η δήλωση ενός μετρητή έχει το ακόλουθο γενικό σχήμα:
proc *name* : **R_TIMER**(lr, L, U, ur, set_cond, expire_cond),
όπου

- το lr δηλώνει αν το κάτω όριο του διαστήματος είναι ανοιχτό ή κλειστό
- το L είναι ένας ακέραιος για το κάτω όριο του διαστήματος
- το U είναι ένας ακέραιος για το άνω όριο του διαστήματος
- το ur δηλώνει αν το άνω όριο του διαστήματος είναι ανοιχτό ή κλειστό



Σχήμα 6.5: Η αρχιτεκτονική του RT-COSPAN

- η set_cond είναι μια λογική έκφραση πάνω στις μεταβλητές επιλογής/απόφασης των S/R διεργασιών που περιγράφουν το σύστημα. Δηλώνει την συνθήκη η οποία πρέπει να γίνει αληθής για να τεθεί ο μετρητής

- η expire_cond είναι μια λογική έκφραση πάνω στις μεταβλητές επιλογής/απόφασης των S/R διεργασιών που περιγράφουν το σύστημα. Δηλώνει την συνθήκη η οποία πρέπει να γίνει αληθής για να εκπνεύσει ο μετρητής.

Η αρχιτεκτονική του RT-COSPAN

Η αρχιτεκτονική του εργαλείου RT-COSPAN δίνεται στην εικόνα 6.5. Οπως φαίνεται ο κορμός της αρχιτεκτονικής είναι ο ίδιος με του COSPAN.

Σε οτι ακολουθεί θα περιγράψουμε συνοπτικά την λειτουργία του RT-COSPAN. Και πάλι, όπως και στο COSPAN, η είσοδος είναι ένα S/R αρχείο **foo.sr**. Αυτό το αρχείο περιέχει την περιγραφή ενός συστήματος πραγματικού χρόνου, η οποία αποτελείται από δηλώσεις μετρητών και διεργασιών. Οι δηλώσεις των μετρητών μεταφράζονται κατάλληλα σε ένα σύνολο S/R διεργασιών. Ετσι, η περιγραφή ενός συστήματος πραγματικού χρόνου, αποτελείται από ένα σύνολο S/R διεργασιών που περιγράφουν τους μετρητές και ένα σύνολο διεργασιών που περιγράφουν το υπόλοιπο σύστημα. Ο μεταφραστής S/R μεταφράζει την παραπάνω περιγραφή σ' ένα C πρόγραμμα.

Ο C κώδικας συνενώνεται με μια βιβλιοθήκη από ρουτίνες που ονομάζεται **librt.a**. Αυτή η βιβλιοθήκη περιέχει τις ρουτίνες που υλοποιούν το μοντέλο των *Set/Expire* χρονικών γράφων καθώς και τις ρουτίνες που περιέχονται στην βιβλιοθήκη **libsr.a** του COSPAN που υλοποιούν τον αλγόριθμο ανάλυσης του COSPAN. Το αρχείο που προκύπτει, περιέχει εκτελέσιμο κώδικα. Εκτεταμένες λεπτομέρειες για την υλοποίηση του RT-COSPAN μπορούν να βρεθούν στο [Tzo92].

Το τροποποιημένο RT-COSPAN

Οπως έχει αναφερθεί και προηγουμένως, η έκδοση του RT-COSPAN που προτείνεται στο [Tzo92] βασίζεται στην επέκταση του COSPAN με το μοντέλο των *Set/Expire* χρονικών γράφων.

Εμείς τροποποιήσαμε κατάλληλα αυτήν την υλοποίηση, έτσι ώστε η καινούρια έκδοση του εργαλείου να υλοποιεί την επέκταση του μοντέλου των *Set/Expire* χρονικών γράφων που προτείναμε στο κεφάλαιο 5.

Κρατήσαμε αμετάβλητη την αρχιτεκτονική του RT-COSPAN. Τροποποιήσαμε μόνο κάποια βασικά σημεία του κώδικα που υλοποιεί το μοντέλο των *Set/Expire* χρονικών γράφων. Κάναμε ακόμα κάποιες αλλαγές για να αυξήσουμε την απόδοση της νέας έκδοσης του εργαλείου.

Στο κεφάλαιο 6 είδαμε πώς ο χρήστης μπορεί να περιγράψει ένα σύστημα στο εργαλείο RT-COSPAN. Από μια τέτοια περιγραφή το εργαλείο κατασκευάζει (δείτε κεφάλαιο 4) ένα αυτόματο, το

$$B_{UT} = B \cap WF_1 \cap WF_2 \cap \ldots \cap WF_n,$$

όπου το B περιγράφει τη λογική συμπεριφορά του συστήματος και τα WF_i, ένα για κάθε μετρητή ελέγχουν τις συνθήκες καλού ορισμού των μετρητών. Εκτός από τα παραπάνω, το RT-COSPAN διαχειρίζεται την χρονική πληροφορία του συστήματος με τρόπο δυναμικό. Σε κάθε κατάσταση του παραπάνω αυτομάτου αντιστοιχεί έναν ΠΦΔ ο οποίος κωδικοποιεί την χρονική πληροφορία που αντιστοιχεί στην συγκεκριμένη κατάσταση του αυτομάτου. Ο αλγόριθμος που δίνεται στο κεφάλαιο 4 χειρίζεται τους ΠΦΔ με κατάλληλο τρόπο, έτσι ώστε, δοσμένου ενός ΠΦΔ και ενός συνόλου γεγονότων μετρητών, ο αλγόριθμος υπολογίζει έναν νέο ΠΦΔ. Αυτός ο υπολογισμός είναι στην ουσία ο υπολογισμός της *nexttv* με την βοήθεια ΠΦΔ.

Η νέα έκδοση του εργαλείου έχει τις εξής βασικές διαφορές από την προηγούμενη: Πρώτα απ' όλα, δεν χρειάζεται πλέον να υλοποιούμε τα αυτόματα WF_i για κάθε μετρητή, μιας και έχουμε αφαιρέσει από το μοντέλο τις συνθήκες καλού ορισμού των μετρητών. Στη νέα έκδοση, μας αρκεί να κρατάμε σε μια λίστα, τους μετρητές που είναι ενεργοί κάθε φορά.

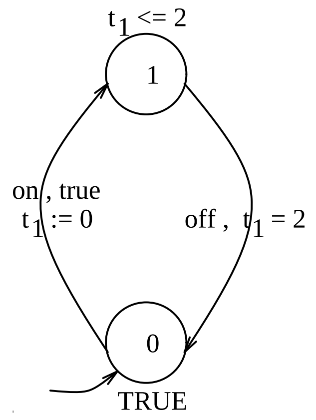
Η δεύτερη βασική διαφορά είναι στον υπολογισμό της *nexttv*. Ο αλγόριθμος που προτείνεται στο κεφάλαιο 4 μένει αμετάβλητος εκτός του ότι του προσθέτουμε ένα βήμα στην αρχή, όπου κάθε φορά πριν υπολογίσει την *nexttv*, υπολογίζει το σύνολο των "σχετικών μετρητών" $A_{relevant}$ σύμφωνα με τον τύπο του κεφαλαίου 5. Ετσι, στον αλγόριθμο του κεφαλαίου 4 αρκεί να αντικαταστήσουμε όπου εμφανίζεται το A σύνολο με το $A_{relevant}$.

6.2 Το εργαλείο επαλήθευσης KRONOS

Ο KRONOS είναι ένα εργαλείο για αυτόματη επαλήθευση συστημάτων πραγματικού χρόνου [OY92]. Ο KRONOS διαβάζει ένα χρονικό γράφο που περιγράφει το σύστημα πραγματικού χρόνου και μια έκφραση TCTL που περιγράφει μια ιδιότητα και ελέγχει την εγκυρότητα της έκφρασης στον χρονικό γράφο. Η εγκυρότητα ελέγχεται με την μέθοδο του "συμβολικού ελέγχου μοντέλου". Οι χρονικοί γράφοι είναι της μορφής που περιγράψαμε στο κεφάλαιο 2. Οι εκφράσεις TCTL είναι της μορφής που περιγράψαμε στι κεφάλαιο 3. Ο αλγόριθμος για συμβολικό έλεγχο μοντέλου έχει περιγραφεί στο κεφάλαιο 4.

Δεν υπάρχουν ακόμα πλήρη τεχνικά κείμενα για τον KRONOS. Αλλά μιας και χρησιμοποιούμε το εργαλείο αυτό για να τρέξουμε κάποια παραδείγματα στα πλαίσια αυτής της εργασίας, θα προσπαθήσουμε να δώσουμε μια γενική περιγραφή του. Σε οτι ακολουθεί θα δώσουμε την γλώσσα περιγραφής που χρησιμοποιεί ο KRONOS, για τους χρονικούς γράφους και τις λογικές εκφράσεις.

Ο KRONOS διαβάζει ένα χρονικό γράφο και μια έκφραση από δυο διαφορετικά αρχεία



Σχήμα 6.6: Ενας χρονικός γράφος που περιγράφει τον διακόπτη δυο καταστάσεων που μπορεί να είναι ενεργός μόνο για δυο χρονικές μονάδες

εισόδου και γράφει το αποτέλεσμα της αξιολόγησης σε ένα αρχείο εξόδου. Τα ονόματα των αρχείων εισόδου για τον χρονικό γράφο και για την έκφραση, έχουν καταληξη **.tg** και **.tctl** αντίστοιχα. Το αρχείο εξόδου περιέχει το αποτέλεσμα του υπολογισμού και το όνομά του λήγει με **.eval**. Το αποτέλεσμα του υπολογισμού, είναι το σύνολο των καταστάσεων του χρονικού γράφου που ικανοποιούν τη λογική έκφραση. Το σύνολο των καταστάσεων γράφεται στο **.eval** αρχείο εξόδου.

6.2.1 Η γλώσσα περιγραφής του KRONOS

Η γλώσσα περιγραφής του KRONOS δίνεται παρακάτω.

Χρονικοί γράφοι

Η εικόνα 6.7 δείχνει πώς ο χρονικός γράφος, που περιγράφει τον διακόπτη δυο καταστάσεων της εικόνας 6.6, εκφράζεται στην γλώσσα περιγραφής του KRONOS.

Η περιγραφή του χρονικού γράφου αποτελείται από μια επικεφαλίδα, που περιέχει πληροφορία για τις καταστάσεις, τις μεταβάσεις και τα ρολόγια του χρονικού γράφου και μια λίστα από ορισμούς μεταβάσεων.

Η εντολή # **states** *num* της επικεφαλίδας ορίζει τον αριθμό των καταστάσεων του χρονικού γράφου. Η λέξη **states** είναι προκαθορισμένη και ο *num* είναι μια ακέραια τιμή. Ομοια οι εντολές # **trans** *num* και # **clocks** *num* ορίζουν τον αριθμό των μεταβάσεων και τον αριθμό των ρολογιών του χρονικού γράφου. Και πάλι οι λέξεις **trans** και **clocks** είναι προκαθορισμένες και *num* είναι μια ακέραια τιμή. Η πληροφορία της επικεφαλίδας τελειώνει με μια λίστα από τα αναγνωριστικά ονόματα των ρολογιών.

Για κάθε κατάσταση ορίζεται το τμήμα ορισμού των μεταβάσεών της. Η εντολή **state:** *num* ξεκινά τον ορισμό του της αριθμημένης με *num* κατάστασης, όπου **state** είναι προκαθορισμένη λέξη και *num* είναι ακέραια τιμή που αντιστοιχει στο αναγνωριστικό της κατάστασης.

Η εντολή **invar:***invariant* ουσχετίζει ένα αναλλοίωτο με την δοσμένη κατάσταση. Η λέξη **invar** είναι προκαθορισμένη και το *invariant* είναι μια λογική έκφραση πάνω στις τιμές των ρολογιών. Ο ορισμός του τμήματος ορισμού των μεταβάσεων κάποιας κατάστασης, ξεκινά με την προκαθορισμένη λέξη **trans** και έχει την ακόλουθη μορφή:

$$
\begin{array}{cccccc}
\textit{condition} & => & \textit{label}; & \textbf{reset} & \{\textit{clock\_id\_list}\}; & \textbf{goto} \quad \textit{state\_num} \\
 & & & \dots & & \\
\textit{condition} & => & \textit{label}; & \textbf{reset} & \{\textit{clock\_id\_list}\}; & \textbf{goto} \quad \textit{state\_num}
\end{array}
$$

Η *condition* είναι μια λογική έκφραση πάνω στις τιμές των ρολογιών του χρονικού γράφου. Η *label* είναι το αναγνωριστικό της ενέργειας που αντιστοιχεί στη δοσμένη μετάβαση. Τα **reset** και **goto** είναι προκαθορισμένες λέξεις. Το **reset** ορίζει το σύνολο των ρολογιών που επανατίθενται στο μήδεν με αυτήν την μετάβαση, και το **goto** δείχνει την κατάσταση στην οποία εισέρχεται αυτή η μετάβαση. Το αναγνωριστικό αυτής της κατάστασης είναι το *state_num.*

TCTL εκφράσεις

Για τους TCTL χρονικούς τελεστές χρησιμοποιείται η παρακάτω σημειολογία:

init παριστάνει την αρχική κατάσταση του χρονικού γράφου, όπου όλα τα ρολόγια είναι αρχικοποιημένα στο μηδέν.

eb$\{\prec$ **n**$\}$ αντιστοιχεί στον τελεστή $\exists\Box_{\prec n}$ όπου $\prec$ είναι ένα από τα $<, \leq, >, \geq, =$.

ed$\{\prec$ **n**$\}$ αντιστοιχεί στον τελεστή $\exists\Diamond_{\prec n}$ όπου $\prec$ είναι ένα από τα $<, \leq, >, \geq, =$.

ab$\{\prec$ **n**$\}$ αντιστοιχεί στον τελεστή $\forall\Box_{\prec n}$ όπου $\prec$ είναι ένα από τα $<, \leq, >, \geq, =$.

ad$\{\prec$ **n**$\}$ αντιστοιχεί στον τελεστή $\forall\Diamond_{\prec n}$ όπου $\prec$ είναι ένα από τα $<, \leq, >, \geq, =$.

eu$\{\prec$ **n**$\}$ αντιστοιχεί στον τελεστή $\exists\mathcal{U}_{\prec n}$ όπου $\prec$ είναι ένα από τα $<, \leq, >, \geq, =$.

```
#states    2
#trans     2
#clocks    1
t1

state:  0
invar:  TRUE
trans:
true    => on;  reset { t1 };  goto  1

state:   1
invar:   ( t1 <= 2)
trans:
( t1 = 2 )   => off;  reset {  };  goto  0
```

Σχήμα 6.7: Ενα πρόγραμμα γραμμένο στη γλώσσα περιγραφής του KRONOS, για τον διακόπτη δυο καταστάσεων

---

au$\{\prec$ **n**$\}$ αντιστοιχεί στον τελεστή $\forall\mathcal{U}_{\prec n}$ όπου $\prec$ είναι ένα από τα $<, \leq, >, \geq, =$.

Τελικά η γραμματική που γεννά τις TCTL εκφράσεις δίνεται παρακάτω:

$$
\begin{array}{rcl}
formula & ::= & \mathbf{true} \\
 & | & \mathbf{init} \\
 & | & \mathbf{state}(num) \\
 & | & clock\_constraint \\
 & | & \mathbf{not}\ formula \\
 & | & formula\ \mathbf{and}\ formula \\
 & | & formula\ \mathbf{or}\ formula \\
 & | & formula\ \mathbf{imp}\ formula \\
 & | & formula\ \mathbf{eu}\{\prec\ n\}\ formula \\
 & | & formula\ \mathbf{au}\{\prec\ n\}\ formula \\
 & | & \mathbf{ed}\{\prec\ n\}\ formula \\
 & | & \mathbf{eb}\{\prec\ n\}\ formula \\
 & | & \mathbf{ad}\{\prec\ n\}\ formula \\
 & | & \mathbf{ad}\{\prec\ n\}\ formula \\
 & | & (\ formula\ )
\end{array}
$$

όπου $clock\_constraint$ είναι της γενικής μορφής:

$$clock\_id_i - clock\_id_j \prec num$$

όπου num είναι μια ακέραια τιμή και $\prec$ είναι ένα από τα $<, \leq, >, \geq, =$.

Κεφάλαιο 7

Ενα μεγάλο παράδειγμα

Σ' αυτό το κεφάλαιο θα παρουσιάσουμε ένα παράδειγμα, που αποδεικνύει την χρησιμότητα του μεταφραστικού σχήματος, που προτείναμε στα πλαίσια αυτής της εργασίας. Θα περιγράψουμε λοιπόν ένα σύστημα και κατόπιν θα δώσουμε τα αποτελέσματα της επαλήθευσης που προκύπτουν από τα εργαλεία RT-COSPAN και KRONOS αντίστοιχα.

7.1 Το πρωτόκολλο επικοινωνίας FDDI

Το FDDI, *(Fiber Distributed Data Interface)*, είναι ένα πρωτόκολλο για τοπικά δίκτυα μεγάλων ταχυτήτων. Το FDDI είναι το πρωτόκολλο που καθορίζει την λειτουργία του επιπέδου ελέγχου πρόσβασης μέσου (medium access control) και του φυσικού επιπέδου για ένα τοπικό δίκτυο δακτυλίου με κουπόνι πρόσβασης, κατασκευασμένο από οπτική ίνα και με χωρητικότητα 100Mbps.

7.1.1 Ο αλγόριθμος του FDDI

Σ' αυτήν την ενότητα περιγράφουμε το πρωτόκολλο FDDI. Στην περιγραφή μας, αποφεύγουμε να δώσουμε κάποιες τεχνικές λεπτομέρειες, όπως για παράδειγμα το μέγεθος του πλαισίου που μεταφέρει τα μηνύματα ή την κωδικοποίηση των μηνυμάτων στο πλαίσιο κλπ. Τις τεχνικές λεπτομέρειες του πρωτοκόλλου μπορείτε να τις βρείτε στα [Joh87, SJ87].

Η τεχνική του FDDI βασίζεται στην χρήση ενός μικρού πακέτου-κουπονιού (token), το οποίο κυκλοφορεί γύρω-γύρω στον δακτύλιο, όταν όλοι οι σταθμοί είναι ανενεργοί. Ενας σταθμός που επιθυμεί να στείλει πακέτα, πρέπει να περιμένει μέχρι να περάσει από αυτόν ένα ελεύθερο πακέτο-κουπόνι. Αφού πρώτα καταλάβει το πακέτο-κουπόνι, ο σταθμός μπορεί να μεταφέρει ένα ή περισσότερα πακέτα, μέχρι να μην έχει τίποτα άλλο να στείλει ή μέχρι ένας μετρητής να εκπνεύσει. Αυτός ο μετρητής, ονομάζεται "μετρητής κατακράτησης του πακέτου-κουπονιού". Κατά τη διάρκεια που ένας σταθμός κατέχει το πακέτο-κουπόνι, δεν υπάρχει πακέτο-κουπόνι ελεύθερο πάνω στο δακτύλιο. Κάθε άλλος σταθμός που επιθυμεί να στείλει πληροφορία, πρέπει να περιμένει και απλώς "ακούει" τον δακτύλιο και αφήνει να περνάνε από αυτόν τα πακέτα που στέλνει ο σταθμός που διαθέτει το πακέτο-κουπόνι. Κάθε σταθμός προκαλεί, προσεγγιστικά, 1bit καθυστέριση στον δακτύλιο. Αν ο σταθμός ανιχνεύσει την δική του διεύθυνση πάνω σε κάποιο απο τα πακέτα που περνάνε, τότε αντιγράφει το πακέτο. Ενα πακέτο πάνω στο δακτύλιο κάνει πάντα μια πλήρη περιφορά και απορροφάται από τον σταθμό που το έστειλε. Ο σταθμός που στέλνει, εισάγει ένα νέο πακέτο-κουπόνι στον δακτύλιο μόλις ολοκληρώσει την μεταφορά των πακέτων που θέλει να στείλει.

Αν συμβεί κάποιο λάθος, το FDDI πρωτόκολλο δεν προσπαθεί να το διορθώσει. Δεν προσπαθεί δηλαδή να ξαναστείλει το πακέτο που μεταφέρει το λάθος. Αυτό είναι δουλειά κάποιου πρωτοκόλλου υψηλότερου επιπέδου.

Το FDDI έχει σχεδιαστεί για να υποστηρίζει ένα μείγμα από ομοιόμορφη κίνηση δεδομένων και από κίνηση με απότομες διακυμάνσεις. Δυο ειδών μηνύματα ορίζονται: σύγχρονα μηνύματα, που είναι υψηλής προτεραιότητας, και ασύγχρονα μηνύματα, που είναι χαμηλής προτεραιότητας.

Ορίζεται ακόμα μια σταθερά χρόνου που ονομάζεται "Χρόνος Ολικής Περιφοράς του Πακέτου-κουπονιού" (Target Token Rotation Time, $TTRT$). Η τιμή του $TTRT$ είναι η ίδια για όλους τους σταθμούς πάνω στον δακτύλιο. Για την μεταφορά σύγχρονων μηνυμάτων κάθε σταθμός μπορεί να διαθέσει ένα χρόνο ίσο με $SA_i \geq 0$. Η τιμή του SA_i μπορεί να διαφέρει από σταθμό σε σταθμό.

Το πρωτόκολλο πρέπει να εξασφαλίζει οτι ισχύει η ακόλουθη ανισότητα: $\Sigma SA_i + D_{max} + F_{max} + Token\_Time \leq TTRT$ όπου SA_i είναι ο χρόνος που αφιερώνεται για την μεταφορά σύγχρονων μηνυμάτων από τον σταθμό i, D_{max} είναι η καθυστέριση πλήρους περιφοράς, F_{max} είναι ο χρόνος που απαιτείται για να μεταφέρουμε ένα πακέτο μεγίστου μεγέθους και τελικά $Token\_Time$ είναι ο χρόνος που απαιτείται για την αποστολή του πακέτου-κουπονιού.

Το πρωτόκολλο υποστηρίζει προαιρετικά την διακίνηση σύγχρονων μηνυμάτων. Ενας σταθμός που δεν υποστηρίζει την διακίνηση σύγχρονων μηνυμάτων, μπορεί να στέλνει μόνο ασύγχρονα μηνύματα.

Κάθε σταθμός έχει δυο μετρητές, τον Μετρητή Περιφοράς του Πακέτου-κουπονιού TRT (Token Rotation Timer) και τον Μετρητή Κατακράτησης του Πακέτου-κουπονιού THT (Token

Holding Timer). Κάθε φορά που ένας σταθμός λαμβάνει το πακέτο-κουπόνι, μετρά το χρόνο που περνά από την προηγούμενη λήψη του πακέτου-κουπονιού, ο οποίος κρατείται στον μετρητή TRT. Αυτή η τιμή σώζεται στον THT και έπειτα ο TRT επανατίθεται στο μηδέν και ξαναρχίζει να μετρά. Ο σταθμός μπορεί να μεταφέρει πληροφορία, σύμφωνα με τους ακόλουθους κανόνες:

1. Μπορεί να μεταφέρει σύγχρονα μηνύματα για χρόνο SA_i το πολύ,

2. Αφού έχει τελειώσει με την αποστολή των σύγχρονων μηνυμάτων, ή αν δεν υπάρχουν σύγχρονα μηνύματα για να σταλούν, ο THT ενεργοποιείται και ξεκινά να μετρά από την τιμή στην οποία έχει τεθεί. Ο σταθμός μπορεί να μεταφέρει ασύγχρονα μηνύματα όσο $THT < TTRT$.

Είναι πιθανόν, όταν ένας σταθμός λαμβάνει το πακέτο-κουπόνι, η τιμή του TRT, να είναι μεγαλύτερη από $TTRT$, έτσι η αρχική τιμή του THT υπερβαίνει το $TTRT$. Σ' αυτή την περίπτωση ο σταθμός είναι δυνάτον να στείλει μόνο σύγχρονα μηνύματα, αν έχει σύγχρονα μηνύματα να στείλει. Αλλά μετά τη φάση αποστολής σύγχρονων μηνυμάτων, ο σταθμός δεν μπορεί να στείλει ασύγχρονα μηνύματα και πρέπει οποσδήποτε να ελευθερώσει αμέσως το πακέτο-κουπόνι. Η συμπεριφορά κάθε σταθμού περιγράφεται σχηματικά στην εικόνα 7.1.

Ο αντικειμενικός σκοπός του FDDI πρωτοκόλλου είναι να εξασφαλίζει οτι ο χρόνος μεταξύ δυο διαδοχικών υποδοχών του πακέτου-κουπονιού από ένα σταθμό είναι μικρότερος ή ίσος του $TTRT$. Η παραπάνω ιδιότητα δεν είναι αληθής, μόνο στη φάση αρχικοποίησης του συστήματος και πολύ γρήγορα ο χρόνος αυτός σταθεροποιείται σε ένα διάστημα τιμών με άνω όριο το $TTRT$.

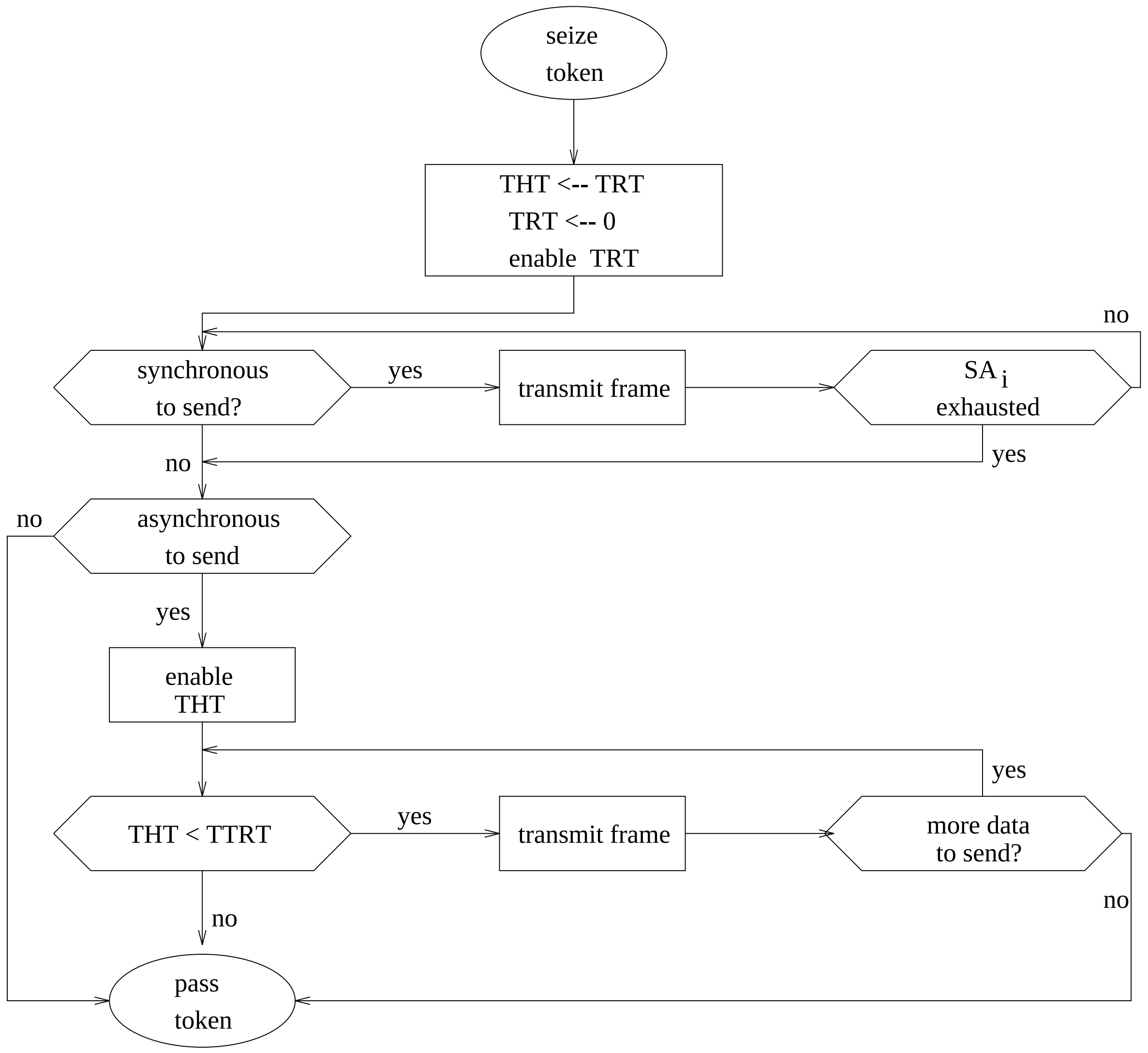
Δυο σημαντικές ιδιότητες του πρωτοκόλλου είναι οι ακόλουθες:

- **δικαιοσύνη (fairness)**, δηλαδή, κάθε σταθμός πάνω στον δακτύλιο πάντα τελικά έχει την ευκαιρία να στείλει ασύγχρονα μηνύματα, για μια θετική ποσότητα χρόνου.

- **φραγμένος χρόνος πρόσβασης μέσου, (bounded medium access time**, δηλαδή, ο χρόνος που χρειάζεται για έναν σταθμό να έχει πρόσβαση στο μέσο, είναι φραγμένος από πάνω με μια σταθερά, ίδια για όλους τους σταθμούς.

7.1.2 Περιγραφή του FDDI πρωτοκόλλου χρησιμοποιώντας χρονικούς γράφους

Το γενικό σχήμα του συστήματος περιγράφεται στην εικόνα 7.2. Υπάρχουν δυο είδη διεργασιών:



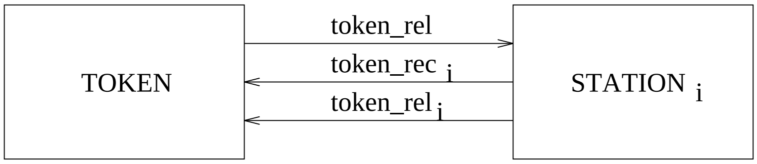
Σχήμα 7.1: Σχηματική αναπαράσταση της λειτουργίας κάθε σταθμού του FDDI δακτυλίου με κουπόνι πρόσβασης

- Η διεργασία *token* που περιγράφει δυο πράγματα. Πρώτ' απ' όλα, περιγράφει την τοπολογία του δακτυλίου. Δεύτερον, αναφέρει ποιος σταθμός έχει υπό την κατοχή του το πακέτο-κουπόνι.

- Η διεργασία *station*, που περιγράφει την λειτουργία κάθε σταθμού στο δακτύλιο.

Τα σήματα μέσω των οποίων η διεργασία *token* επικοινωνεί με την *station* είναι τα ακόλουθα:

- *token_rel* που στέλνεται από την *token* σε όλες τις διεργασίες τύπου *station*, που υπάρχουν στο σύστημα. Υποδηλώνει ένα βήμα αρχικοποίησης για το σύστημα.



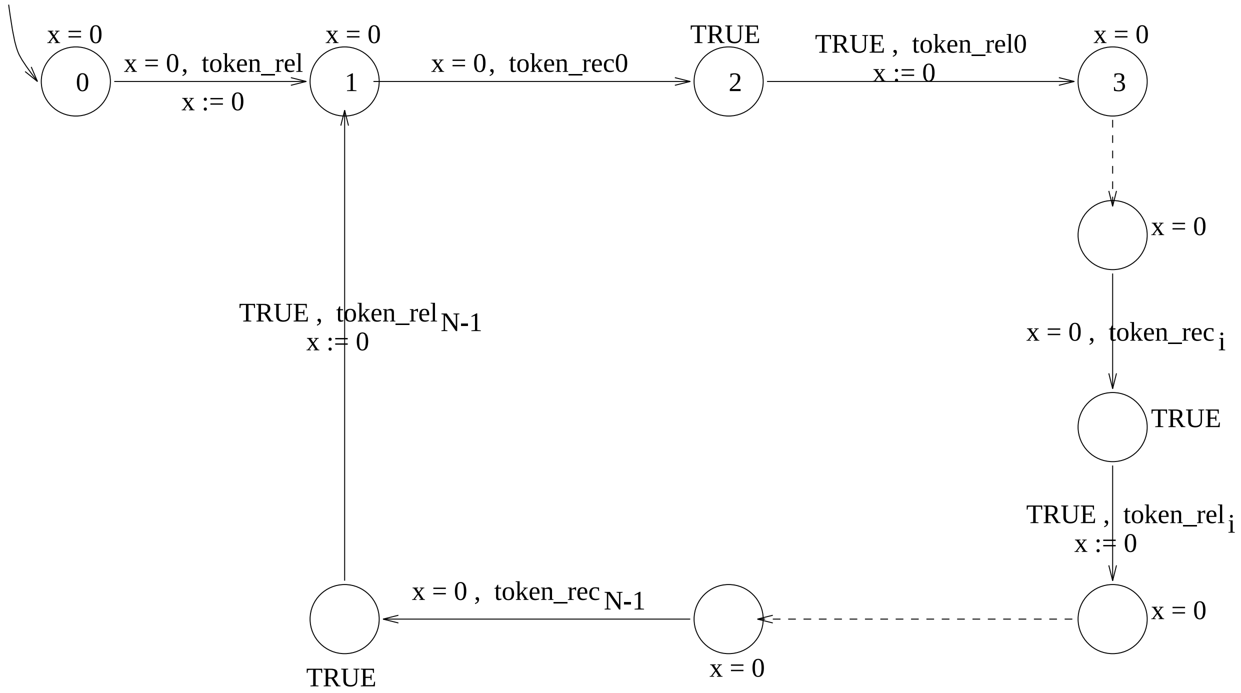
Σχήμα 7.2: Γενικό σχήμα του συστήματος

- $token\_rec_i$ που στέλνεται από κάθε $station_i$ και δηλώνει την υποδοχή του πακέτου-κουπονιού από τον σταθμό $station_i$.
- $token\_rel_i$ που στέλνεται από κάθε $station_i$ και δηλώνει την απελευθέρωση του πακέτου-κουπονιού από τον σταθμό $station_i$.

Για απλότητα έχουμε κάνει τις παρακάτω παραδοχές:

- Θεωρούμε οτι τα πακέτα που στέλνονται είναι όλα του ίδιου μήκους,
- Θεωρούμε αμελητέες τις καθυστερήσεις διάδοσης των μηνυμάτων. Στην πραγματικότητα τις θεωρούμε πολύ μικρές σε σχέση με το $TTRT$. Αυτή είναι μια έγκυρη παραδοχή για τις περισσότερες των περιπτώσεων που συμβαίνουν στην πραγματικότητα,
- Δεν μοντελοποιούμε την μεταφορά των σύγχρονων μηνυμάτων.

Οι διεργασίες του συστήματος περιγράφονται με χρονικούς γράφους. Ολόκληρο το σύστημα περιγράφεται με την παράλληλη σύνθεση αυτών των χρονικών γράφων. Η διεργασία $token$ περιγράφεται από τον χρονικό γράφο $TOKEN$ που δίνεται στην εικόνα 7.3. Το ρολόι x χρησιμοποιείται για να μετρά χρόνους ίσους με μηδέν. Η αρχική κατάσταση του χρονικού γράφου είναι η κατάσταση 0. Η διεργασία πρέπει να φύγει από την κατάσταση 0 αμέσως, μιας και το αναλλοίωτο της κατάστασης 0 είναι $x = 0$. Η μετάβαση από την κατάσταση 0 στην 1 έχει σαν ετικέττα την ενέργεια $token\_rel$. Αυτή η ενέργεια είναι μια ενέργεια αρχικοποίησης του συστήματος που εκφράζει το γεγονός οτι υπάρχει τώρα πακέτο-κουπόνι διαθέσιμο στον δακτύλιο. Ο σταθμός πρώτα που πρώτος λαμβάνει το πακέτο-κουπόνι είναι ο $station_0$. Σε κάθε κατάσταση με αριθμό $2*i+1$ όπου $i = 0, 1, 2, \ldots$ του χρονικού γράφου TOKEN το πακέτο-κουπόνι είναι διαθέσιμο στον $station_i$. Η μετάβαση που ενεργοποιείται είναι η $token\_rec_i$ και η διεργασία $token$ κινείται από την κατάσταση νούμερο $2*i+1$ στην $2*i$ με $i = 1, 2, \ldots$. Στις καταστάσεις $2*i$, $i = 1, 2, \ldots$ το πακέτο-κουπόνι είναι υπό την κατοχή του σταθμού $station_i$ και η εξέρχουσα

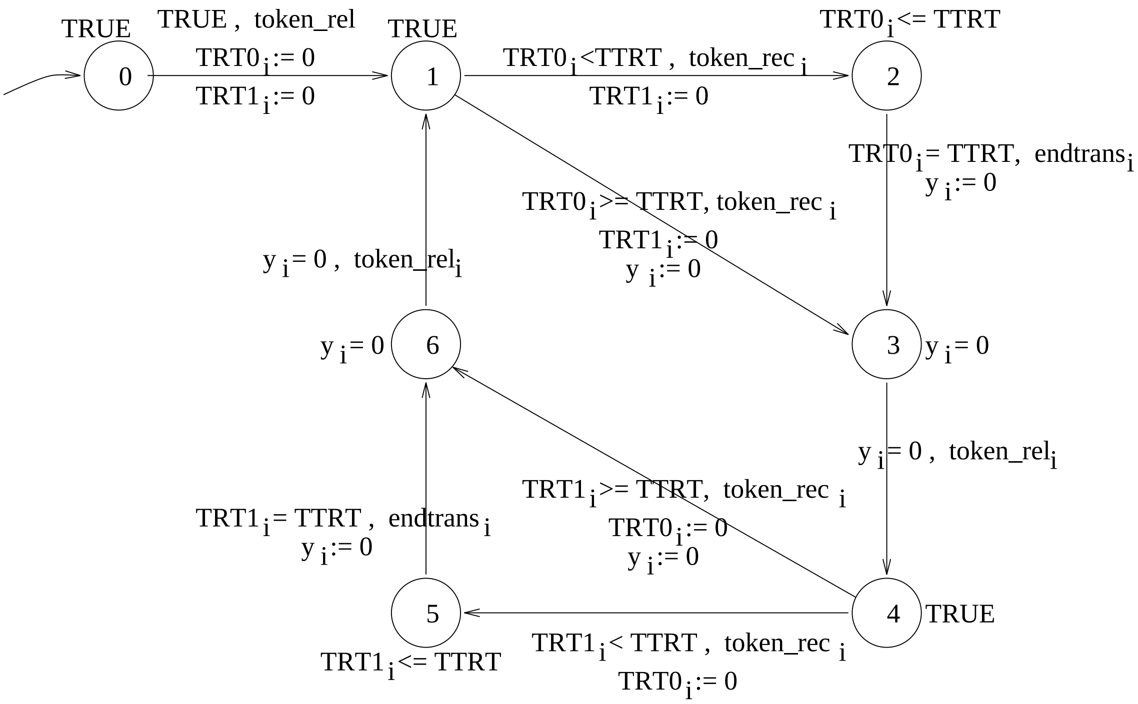


Σχήμα 7.3: Ο χρονικός γράφος TOKEN που περιγράφει την διεργασία *token*

ακμή έχει σαν ετικέττα την ενέργεια $token\_rel_i$, που δηλώνει οτι ο σταθμός $station_i$ ελευθερώνει το πακέτο-κουπόνι. Ετσι στην επόμενη κατάσταση το πακέτο-κουπόνι είναι διαθέσιμο για τον επόμενο σταθμό.

Κάθε $station_i$ περιγράφεται με έναν χρονικό γράφο ST_i που φαίνεται στην εικόνα 7.4. Υπάρχουν δυο ρολόγια, ονομαστικά το $TRT0_i$ και το $TRT1_i$ τα οποία προσομοιώνουν τους πραγματικούς μετρητές TRT και THT αντίστοιχα του κάθε $station_i$. Στις καταστάσεις 1 και 2 του χρονικού γράφου, το ρολόι $TRT0_i$ ενεργεί σαν να ήταν ο μετρητής TRT και το $TRT1_i$ ενεργεί σαν το THT. Στις καταστάσεις 4 και 5, τα ρολόγια $TRT0_i$ και $TRT1_i$ ανταλλάσσουν ρόλους, έτσι ώστε το $TRT1_i$ ενεργεί πλέον σαν να ήταν ο TRT και το $TRT0_i$ σαν τον THT.

Η αρχική κατάσταση του χρονικού γράφου ST_i είναι η κατάσταση 0. Η διεργασία κινείται από την κατάσταση 0 στην 1 εκτελώντας $token\_rel$. Και τα δυο ρολόγια του χρονικού γράφου επανατίθενται στο μηδέν μ' αυτήν την μετάβαση. Στην κατάσταση 1, η $station_i$ είτε λαμβάνει το πακέτο-κουπόνι αν είναι η σειρά της ή περιμένει σ' αυτήν την κατάσταση, μέχρι να υπάρχει διαθέσιμο πακέτο-κουπόνι στο δακτύλιο και είναι η σειρά της να το πάρει. Αν η $station_i$ λαμβάνει το πακέτο-κουπόνι και η τιμή του $TRT0_i$ είναι μικρότερη του $TTRT$ τότε πηγαίνει στην κατάσταση 2. Η κατάσταση 2 αντιστοιχεί στην κατάσταση από την οποία ο σταθμός μπορεί να στάλνει πακέτα. Η μεταφορά των πακέτων δεν μοντελοποιείται, μιας και υποθέτουμε οτι



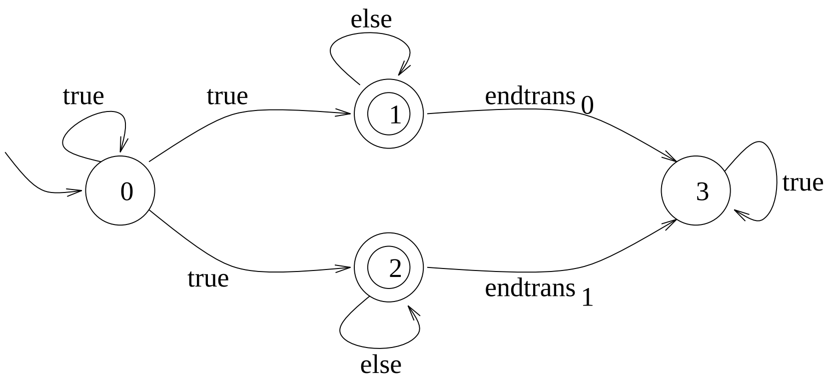
Σχήμα 7.4: Ο χρονικός γράφος ST_i που περιγράφει την διεργασία $station_i$

η καθυστέρηση διάδοσης των πακέτων είναι μηδέν. Αλλά ο σταθμός μπορεί να μείνει στην κατάσταση 2 όσο η τιμή του $TRT0_i$ είναι μικρότερη ή ίση του $TTRT$. Ο σταθμός σταματά να στέλνει είτε γιατί δεν έχει άλλα πακέτα να στείλει, είτε γιατί η τιμή του $TRT0_i$ έχει φτάσει την τιμή $TTRT$, πηγαίνει από την κατάσταση 2 στην κατάσταση 3, απ' όπου αμέσως πηγαίνει στην 4 ελευθερώνοντας το πακέτο-κουπόνι.

Αν ο $station_i$ λάβει το πακέτο-κουπόνι και η τιμή του $TRT0_i$ είναι μεγαλύτερη από $TTRT$, τότε κινείται από την κατάσταση 1 στην 3 απ' όπου πηγαίνει αμέσως στην 4 ελευθερώνοντας το πακέτο-κουπόνι. Η μόνη διαφορά είναι οτι τα ρολόγια $TRT0_i$, $TRT1_i$ έχουν αλλάξει ρόλους.

Εφαρμογή της μεθόδου μας

Ο χρονικός γράφος που περιγράφει ολόκληρο το σύστημα δίνεται από την παράλληλη σύνθεση όλων των ST_i και της διεργασίας $TOKEN$:
$FDDI = ST_0 \| ST_1 \| \ldots \| ST_n \| TOKEN$
Εφαρμόζουμε το μεταφραστικό μας σχήμα στον προκυπτόμενο χρονικό γράφο και παίρνουμε σαν αποτέλεσμα το ζευγάρι (B, T) που περιγράφει το FDDI πρωτόκολλο. Στο Παράρτημα Α,



Σχήμα 7.5: Το συμπλήρωμα του καθήκοντος για την ιδιότητα της δικαιοσύνης

δίνουμε σαν παράδειγμα τον πλήρη χρονικό γράφο στην γλώσσα περιγραφής του KRONOS και το (B, T) στην γλώσσα περιγραφής του RT-COSPAN για την περίπτωση που υπάρχουν δυο σταθμοί στον δακτύλιο.

7.1.3 Επαλήθευση με το εργαλείο RT-COSPAN

Τα Büchi αυτόματα που περιγράφουν το συμπλήρωμα των ιδιοτήτων για τις ιδιότητες που μας ενδιαφέρουν δίνονται στις εικόνες 7.5 για την ιδιότητα της δικαιοσύνης και 7.6 για την ιδιότητα της φραγμένης πρόσβασης μέσου. Το Büchi της εικόνας 7.5 δέχεται όλες τις συμπεριφορές στις οποίες κάποιος σταθμός θα στείλει μηνύματα για πεπερασμένο αριθμό φορών, ενώ το Büchi της εικόνας 7.6 ελέγχει μη αιτιοκρατικά δυο οποιεσδήποτε διαδοχικές υποδοχές του πακέτου-κουπονιού για κάθε σταθμό. Ενας μετρητής (ανά σταθμό) αντιστοιχίζεται με αυτό το Büchi με διάστημα χρόνου $(TTRT, \propto]$ για να μετρά τον χρόνο μεταξύ δυο διαδοχικών υποδοχών του πακέτου-κουπονιού. Ετσι αυτό το (B, T) δέχεται όλες τις συμπεριφορές για τις οποίες υπάρχει ένας σταθμός που υποδέχεται το πακέτο-κουπόνι δυο διαδοχικές φορές, σε χρόνο μεγαλύτερο του $TTRT$.

Τα αποτελέσματα της επαλήθευσης για διαφορετικές τιμές δίνονται στον παρακάτω πίνακα.

property	#stations	#timers	TTRT	states searched	time
fairness	2	25	1	700	26.9
	2	25	10	700	26.4
	2	25	100	700	28.4
	3	37	1	1621	164.3
	3	37	100	1621	163.6
	4	48	1	11862	2414.3
	4	48	100	11862	2418.6
bmat	2	27	1	696	22.9
	2	27	100	696	22.0
	3	40	1	1305	108.4
	3	40	100	1305	95.1
	4	53	1	7944	1164.4
	5	66	1	11553	3082.7

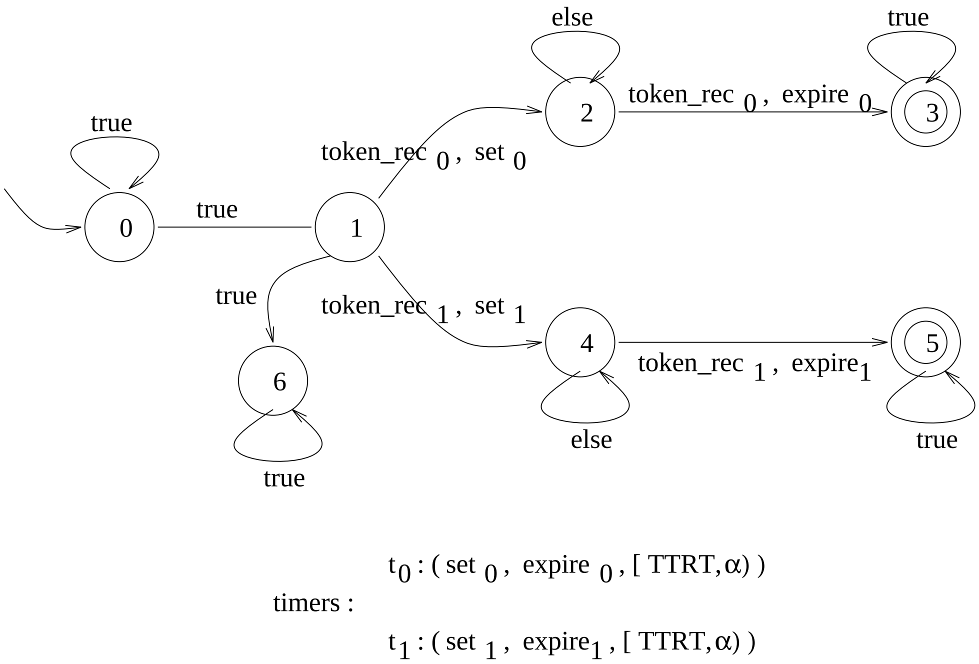
7.1.4 Επαλήθευση με το εργαλείο KRONOS

Οι εκφράσεις TCTL, που περιγράφουν τις παραπάνω ιδιότητες είναι οι ακόλουθες:

1. *safety* $\forall\Box(\forall\Diamond(transmit_i))$

2. *bounded medium access* $after(token\_rec_i) \Rightarrow \forall\Diamond_{\leq TTRT} after(token\_rec_i)$

Τα αποτελέσματα τις επαλήθευσης από το εργαλείο KRONOS για διάφορες περιπτώσεις δίνονται στον ακόλουθο πίνακα.



Σχήμα 7.6: Το συμπλήρωμα του καθήκοντος για την ιδιότητα της φραγμένης πρόσβασης μέσου. Δέχεται όλες τις συμπεριφορές στις οποίες ο χρόνος μεταξύ δυο διαδοχικών υποδοχών του πακέτου-κουπονιού είναι μεγαλύτερος από TTRT

---

property	#stations	#timers	TTRT	time	iterations
fairness	2	7	1	2.45	22
	2	7	100	424.2	1804
	3	10	1	152.1	96
	3	10	100	19391.8	7.206
	4	13	1	9016.1	248
	4	13	100		
bmat	2	7	1	2.05	8
	2	7	100	2.09	8
	3	10	1	8.8	19
	3	10	100	8.6	19
	4	13	1	38.9	37
	4	13	100	39.01	37
	5	16	1	75.09	64

Κεφάλαιο 8

Συμπεράσματα – Μελλοντική εργασία

Στα πλαίσια αυτής της εργασίας εργαστήκαμε σε δυο κατευθύνσεις. Πρώτ' απ' όλα προτείναμε μια επέκταση του μοντέλου των *Set/Expire* χρονικών γράφων. Το κύριο μειονέκτημα του παραπάνω μοντέλου είναι η δυσκολία του να περιγράψει αποδοτικά συστήματα που περιέχουν timeouts ή μη αιτιοκρατική χρονική συμπεριφορά. Στη συνέχεια υλοποιήσαμε την δική μας έκδοση του μοντέλου των *Set/Expire* χρονικών γράφων, τροποποιώντας κατάλληλα το ήδη υλοποιημένο εργαλείο RT-COSPAN. Αυτό τελικά που πετύχαμε, επεκτείνοντας το παραπάνω μοντέλο, είναι μικρότερες περιγραφές συστημάτων, που θέλουμε να τα επαληθεύσουμε με τη βοήθεια του RT-COSPAN.

Δεύτερον, προτείναμε ένα μεταφραστικό σχήμα, που δοσμένης μιας περιγραφής ενός συστήματος σαν ένας χρονικός γράφος, μας δίνει σαν αποτέλεσμα μια ισοδύναμη περιγραφή στο εκτεταμένο μοντέλο των *Set/Expire* χρονικών γράφων. Τα μειονεκτήματα του μεταφραστικού μας σχήματος είναι κυρίως δυο. Πρώτον, παίρνει σαν είσοδο μόνο μια συγκεκριμένη μορφή χρονικών γράφων, που τους έχουμε ονομάσει "περιορισμένους χρονικούς γράφους". Δεύτερον, η περιγραφή που προκύπτει από την μετάφραση περιέχει μεγάλο αριθμό μετρητών μιας και για την ώρα δεν εφαρμόζουμε κάποια μέθοδο ελαχιστοποίησης του αριθμού των ρολογιών. Και τα δυο αυτά μειονεκτήματα επιδέχονται βελτίωση με κατάλληλη γενίκευση του μεταφραστικού μας σχήματος. Η γενίκευσή του είναι στα άμεσα μελλοντικά σχέδια αυτής της εργασίας.

Φέροντας σε πέρας τα παραπάνω πετύχαμε τα εξής: Πρώτα απ' όλα, ενοποιήσαμε ως προς τη γλώσσα περιγραφής δυο μεθόδους επαλήθευσης και κατ' επέκταση δυο εργαλεία επαλήθευσης, ονομαστικά τα RT-COSPAN και KRONOS. Ετσι, μπορούμε πλέον να περιγράφουμε ένα σύστημα σαν ένα χρονικό γράφο και να εφαρμόζουμε δυο διαφορετικούς αλγορίθμους επαλήθευσης ανάλογα με την προς απόδειξη ιδιότητα.

Αυτή η δυνατότητα μας διευκολύνει στην περαιτέρω σύγκριση των μεθόδων, κυρίως των αλγορίθμων, για να μπορέσουμε να βγάλουμε ενδιαφέροντα συμπεράσματα για την απόδοση των μεθόδων επαλήθευσης, ανάλογα με το τί θέλουμε να αποδείξουμε.

Πρωταρχικοί στόχοι επέκτασης αυτής της εργασίας είναι, όπως είπαμε και παραπάνω, η γενίκευση του μεταφραστικού σχήματος για τύχαια μορφή χρονικού γράφου καθώς και η ελαχιστοποίηση του αριθμού των ρολογίων στην προκύπτουσα περιγραφή. Αυτό θα είχε ως αποτέλεσμα την βελτίωση των χρόνων επαλήθευσης με το RT-COSPAN, μιας και ήδη γνωρίζουμε, οτι ο αλγόριθμός του εξαρτάται από τον αριθμό των ρολογιών που περιέχονται στην περιγραφή του συστήματος.

Μια άλλη ιδέα για μελλοντική εργασία είναι να συνεχίσουμε την προσπάθεια σύγκρισης και ανάλυσης των μεθόδων με ανάπτυξη νέων ευριστικών αλγορίθμων που θα προκύπτουν από των συνδυασμό των ήδη γνωστών παραπάνω δυο αλγορίθμων.

Παράρτημα Α

Περιγραφή του πρωτοκόλλου FDDI στα εργαλεία επαλήθευσης KRONOS και RT-COSPAN

A.1 Το πρωτόκολλο FDDI για 2 σταθμούς δοσμένο στη γλώσσα περιγραφής του KRONOS

```
graph: fddi
#states 13
#trans  17
#clocks 7
TRT00 TRT10 T0 X TRT01 TRT11 T1

state: 0
invar: X=0
trans:
X=0 => TOKEN_REL;reset{ TRT00 TRT10 X TRT01 TRT11 };goto 1

state: 1
invar: X=0
trans:
```

```
TRT00<1 and X=0 => TOKEN_REC0;reset{ TRT10 };goto 2
TRT00>=1 and X=0 => TOKEN_REC0;reset{ T0 TRT10 };goto 3

state: 2
invar: TRT00<=1
trans:
TRT00=1 => ENDTRANS0;reset{ T0 };goto 3

state: 3
invar: T0=0
trans:
T0=0 => TOKEN_REL0;reset{ X };goto 4

state: 4
invar: X=0
trans:
X=0 and TRT01<1 => TOKEN_REC1;reset{ TRT11 };goto 5
X=0 and TRT01>=1 => TOKEN_REC1;reset{ T1 TRT11 };goto 6

state: 5
invar: TRT01<=1
trans:
TRT01=1 => ENDTRANS1;reset{ T1 };goto 6

state: 6
invar: T1=0
trans:
T1=0 => TOKEN_REL1;reset{ X };goto 7

state: 7
invar: X=0
trans:
TRT10<1 and X=0 => TOKEN_REC0;reset{ TRT00 };goto 8
```

```
TRT10>=1 and X=0 => TOKEN_REC0;reset{ T0 TRT00 };goto 9

state: 8
invar: TRT10<=1
trans:
TRT10=1 => ENDTRANS0;reset{ T0 };goto 9

state: 9
invar: T0=0
trans:
T0=0 => TOKEN_REL0;reset{ X };goto 10

state: 10
invar: X=0
trans:
X=0 and TRT11<1 => TOKEN_REC1;reset{ TRT01 };goto 11
X=0 and TRT11>=1 => TOKEN_REC1;reset{ T1 TRT01 };goto 12

state: 11
invar: TRT11<=1
trans:
TRT11=1 => ENDTRANS1;reset{ T1 };goto 12

state: 12
invar: T1=0
trans:
T1=0 => TOKEN_REL1;reset{ X };goto 1
```

A.2 Το πρωτόκολλο FDDI για 2 σταθμούς δοσμένο στη γλώσσα περιγραφής του RT-COSPAN

```
#include <RT.h>
```

```
monitor     X0 : R_CLOCK( \[, 0, 0, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REL04) + (fddi.# = TOKEN_REL18)
        + (fddi.# = TOKEN_REL012) + (fddi.# = TOKEN_REL116)),
        (fddi.# = TOKEN_REL0),
        0)

monitor     TRT001 : R_CLOCK( \[, 0, 1, \),
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REC09) + (fddi.# = TOKEN_REC010)),
        (fddi.# = TOKEN_REC01),
        0)

monitor     X2 : R_CLOCK( \[, 0, 0, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REL04) + (fddi.# = TOKEN_REL18)
        + (fddi.# = TOKEN_REL012) + (fddi.# = TOKEN_REL116)),
        (fddi.# = TOKEN_REC01),
        0)

monitor     TRT003 : R_CLOCK( \[, 1, 1073741823, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REC09) + (fddi.# = TOKEN_REC010)),
        (fddi.# = TOKEN_REC02),
        0)

monitor     X4 : R_CLOCK( \[, 0, 0, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REL04) + (fddi.# = TOKEN_REL18)
        + (fddi.# = TOKEN_REL012) + (fddi.# = TOKEN_REL116)),
        (fddi.# = TOKEN_REC02),
        0)
```

```
monitor     TRT005 : R_CLOCK( \[, 1, 1, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REC09) + (fddi.# = TOKEN_REC010)),
        (fddi.# = ENDTRANS03),
        0)

monitor     T06 : R_CLOCK( \[, 0, 0, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REC02)
        + (fddi.# = ENDTRANS03) + (fddi.# = TOKEN_REC010)
        + (fddi.# = ENDTRANS011)),
        (fddi.# = TOKEN_REL04),
        0)

monitor     X7 : R_CLOCK( \[, 0, 0, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REL04) + (fddi.# = TOKEN_REL18)
        + (fddi.# = TOKEN_REL012) + (fddi.# = TOKEN_REL116)),
        (fddi.# = TOKEN_REC15),
        0)

monitor     TRT018 : R_CLOCK( \[, 0, 1, \),
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REC113) + (fddi.# = TOKEN_REC114)),
        (fddi.# = TOKEN_REC15),
        0)

monitor     X9 : R_CLOCK( \[, 0, 0, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REL04) + (fddi.# = TOKEN_REL18)
        + (fddi.# = TOKEN_REL012) + (fddi.# = TOKEN_REL116)),
        (fddi.# = TOKEN_REC16),
        0)
```

```
monitor     TRT0110 : R_CLOCK( \[, 1, 1073741823, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REC113) + (fddi.# = TOKEN_REC114)),
        (fddi.# = TOKEN_REC16),
        0)

monitor     TRT0111 : R_CLOCK( \[, 1, 1, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REC113) + (fddi.# = TOKEN_REC114)),
        (fddi.# = ENDTRANS17),
        0)

monitor     T112 : R_CLOCK( \[, 0, 0, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REC16)
        + (fddi.# = ENDTRANS17) + (fddi.# = TOKEN_REC114)
        + (fddi.# = ENDTRANS115)),
        (fddi.# = TOKEN_REL18),
        0)

monitor     TRT1013 : R_CLOCK( \[, 0, 1, \),
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REC01) + (fddi.# = TOKEN_REC02)),
        (fddi.# = TOKEN_REC09),
        0)

monitor     X14 : R_CLOCK( \[, 0, 0, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REL04) + (fddi.# = TOKEN_REL18)
        + (fddi.# = TOKEN_REL012) + (fddi.# = TOKEN_REL116)),
        (fddi.# = TOKEN_REC09),
        0)
```

```
monitor     TRT1015 : R_CLOCK( \[, 1, 1073741823, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REC01) + (fddi.# = TOKEN_REC02)),
        (fddi.# = TOKEN_REC010),
        0)

monitor     X16 : R_CLOCK( \[, 0, 0, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REL04) + (fddi.# = TOKEN_REL18)
        + (fddi.# = TOKEN_REL012) + (fddi.# = TOKEN_REL116)),
        (fddi.# = TOKEN_REC010),
        0)

monitor     TRT1017 : R_CLOCK( \[, 1, 1, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REC01) + (fddi.# = TOKEN_REC02)),
        (fddi.# = ENDTRANS011),
        0)

monitor     T018 : R_CLOCK( \[, 0, 0, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REC02)
        + (fddi.# = ENDTRANS03) + (fddi.# = TOKEN_REC010)
        + (fddi.# = ENDTRANS011)),
        (fddi.# = TOKEN_REL012),
        0)

monitor     X19 : R_CLOCK( \[, 0, 0, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REL04) + (fddi.# = TOKEN_REL18)
        + (fddi.# = TOKEN_REL012) + (fddi.# = TOKEN_REL116)),
        (fddi.# = TOKEN_REC113),
        0)
```

```
monitor     TRT1120 : R_CLOCK( \[, 0, 1, \),
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REC15) + (fddi.# = TOKEN_REC16)),
        (fddi.# = TOKEN_REC113),
        0)

monitor     X21 : R_CLOCK( \[, 0, 0, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REL04) + (fddi.# = TOKEN_REL18)
        + (fddi.# = TOKEN_REL012) + (fddi.# = TOKEN_REL116)),
        (fddi.# = TOKEN_REC114),
        0)

monitor     TRT1122 : R_CLOCK( \[, 1, 1073741823, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REC15) + (fddi.# = TOKEN_REC16)),
        (fddi.# = TOKEN_REC114),
        0)

monitor     TRT1123 : R_CLOCK( \[, 1, 1, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REL0)
        + (fddi.# = TOKEN_REC15) + (fddi.# = TOKEN_REC16)),
        (fddi.# = ENDTRANS115),
        0)

monitor     T124 : R_CLOCK( \[, 0, 0, \],
        ((fddi.# = initialize) + (fddi.# = TOKEN_REC16)
        + (fddi.# = ENDTRANS17) + (fddi.# = TOKEN_REC114)
        + (fddi.# = ENDTRANS115)),
        (fddi.# = TOKEN_REL116),
        0)

monitor     fddi
```

```
stvar   $ : ( 0..13 )
selvar  # : ( initialize, pause, TOKEN_REL0, TOKEN_REC01,
            TOKEN_REC02, ENDTRANS03, TOKEN_REL04,
            TOKEN_REC15, TOKEN_REC16, ENDTRANS17,
            TOKEN_REL18, TOKEN_REC09, TOKEN_REC010,
            ENDTRANS011, TOKEN_REL012, TOKEN_REC113,
            TOKEN_REC114, ENDTRANS115, TOKEN_REL116 )
cyset   { 13 }, { 0 }, { 1 }, { 2 }, { 3 }, { 4 }, { 5 },
        { 6 }, { 7 }, { 8 }, { 9 }, { 10 }, { 11 }, { 12 }
init    13
trans
13  { initialize }
    ->0 : # = initialize;

0   { pause, TOKEN_REL0 }
    ->1 : # = TOKEN_REL0
    ->0 : # = pause;

1   { pause, TOKEN_REC01, TOKEN_REC02 }
    ->2 : # = TOKEN_REC01
    ->3 : # = TOKEN_REC02
    ->1 : # = pause;

2   { pause, ENDTRANS03 }
    ->3 : # = ENDTRANS03
    ->2 : # = pause;

3   { pause, TOKEN_REL04 }
    ->4 : # = TOKEN_REL04
    ->3 : # = pause;

4   { pause, TOKEN_REC15, TOKEN_REC16 }
    ->5 : # = TOKEN_REC15
```

```
    ->6 : # = TOKEN_REC16
    ->4 : # = pause;

5   { pause, ENDTRANS17 }
    ->6 : # = ENDTRANS17
    ->5 : # = pause;

6   { pause, TOKEN_REL18 }
    ->7 : # = TOKEN_REL18
    ->6 : # = pause;

7   { pause, TOKEN_REC09, TOKEN_REC010 }
    ->8 : # = TOKEN_REC09
    ->9 : # = TOKEN_REC010
    ->7 : # = pause;

8   { pause, ENDTRANS011 }
    ->9 : # = ENDTRANS011
    ->8 : # = pause;

9   { pause, TOKEN_REL012 }
    ->10    : # = TOKEN_REL012
    ->9 : # = pause;

10  { pause, TOKEN_REC113, TOKEN_REC114 }
    ->11    : # = TOKEN_REC113
    ->12    : # = TOKEN_REC114
    ->10    : # = pause;

11  { pause, ENDTRANS115 }
    ->12    : # = ENDTRANS115
    ->11    : # = pause;
```

```
12  { pause, TOKEN_REL116 }
    ->1 : # = TOKEN_REL116
    ->12    : # = pause;

end fddi
```

Βιβλιογραφία

[ACD90] R. Alur, C. Courcoubetis, and D. Dill. Model-checking for real-time systems. In *Proceedings of the 5th Symposium on Logic in Computer Science*, pages 414–425, Philadelphia, June 1990. IEEE Computer Society Press.

[ACW90] S. Aggarwal, C. Courcoubetis, and P. Wolper. Adding Liveness Properties to Coupled Finite-State Machines. *ACM Transactions on Programming Languages and Systems*, 12(2):303–339, 1990.

[AD90] R. Alur and D. Dill. Automata for Modeling Real-Time Systems. In *Automata, Languages and Programming : 17th Annual Colloquium*, volume 443 of *Lecture Notes in Computer Science*, pages 322–335, 1990. Warwick University, July 16-20.

[AHU86] A. Aho, J. Hopcroft, and J. Ulman. *Compilers: Principles Techniques and Tools.* Addison-Wesley, 1986.

[Alu91] R. Alur. *Techniques for Automatic Verification of Real-Time Systems.* PhD thesis, Department of Computer Science Stanford University, August 1991. Ph.D. Thesis.

[Buc62] J.R. Buchi. On a Decision Method in Restricted Second Order Arithmetic. In *Proc. Internat. Congr. Logic Method and Philos. Sci. 1960*, pages 1–12, Stanford, 1962. Stanford University Press.

[CDCT92] C. Courcoubetis, D. Dill, M. Chatzaki, and P. Tzounakis. Verification with real-time COSPAN. In *Proceedings of the 4th CAV.* Springer-Verlag, 1992.

[CDT92] C. Courcoubetis, D. Dill, and P. Tzounakis. Adding dense time properties to finite-state machines. Technical report, ICS Univercity of Crete, Greece, 1992.

[CES86] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, Jan 1986.

[Dil89] D. Dill. Timing Assumptions and Verification of Finite-State Concurrent Systems. In *Proc. Workshop on Computer Aided Verification*, Grenoble, June 1989. Lecture Notes in Computer Science, Springer-Verlag.

[EC82] E.A. Emerson and E.M. Clarke. Using Branching Time Logic to Synthesize Synchronization Skeletons. *Science of Computer Programming*, 2:241–266, 1982.

[GK80] B. Gopinath and B. Kurshan. The Selection/Resolution Model for Coordinating Concurrent Processes. In *AT&T Bell Laboratories Technical Report*, 1980.

[HK90] Z. Har'El and R. P Kurshan. Software for Analytical Development of Communication Protocols. *AT&T Technical Journal*, 1990.

[HNSY92] T. Henziger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model-checking for real-time systems. In *Proceedings of the 7th Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1992.

[Hoa85] C. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.

[Joh87] M. Johnson. Proof that Timing Reguirements of the FDDI Token Ring Protocol are satisfied. *IEEE Transactions on Communications*, 35(6):620–625, June 1987.

[KH87] R. Kurshan and Z. Har'El. COSPAN: User's guide. Technical report, AT&T Bell Labroratories, 1987.

[KK86] J. Katzenelson and B. Kurshan. S/R: A Language for Specifying Protocols and other Coordinating Processes. In *Proc. 5th Ann. Int'l Phoenix Conf. Comput. Commun., IEEE*, 1986.

[Lew90] H. Lewis. A logic of concrete time intervals. In *Proceedings of the 5th Symposium on Logic in Computer Science*, Philadelphia, jun 1990.

[Mil80] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, 1980.

[NRSV90] X. Nicollin, J. Richier, J. Sifakis, and J. Voiron. ATP: an Algebra for Timed Processes. In *Proceedings of the IFIP TC 2 Working Conference on Programming Concepts and Methods*, Israel, April 1990.

[NS90] X. Nicollin and J. Sifakis. ATP the Algebra for Timed Processes: Theory and application. In *Rapport de researche RT-C26, LGI-IMAG*, France, December 1990.

[OY92] A. Olivero and S. Yovine. Kronos: A tool for verifying real-time systems. user's quide and reference manual. Technical report, VERIMAG Grenoble France, 1992.

[Pnu85] A. Pnueli. Applications of Temporal Logic to the Specification and Verification of Reactive Systems: A Survey of Current Trends. In *Proc. Advanced School on Current Trends in Concurrency*, pages 510–584, Berlin, 1985. Volume 224, LNCS, Springer-Verlag.

[QS81] J.P. Quielle and J. Sifakis. Specification and Verification of Concurrent Systems in CAESAR. In *Proc. 5th Int'l Symp. on Programming*, volume 137, pages 337–351. Springer-Verlag, Lecture Notes in Computer Science, 1981.

[Sif82] J. Sifakis. A unified approach for studying the properties of transition systems. In *Theoretical Computer Science, 18*, 1982.

[SJ87] K. Sevcik and M. Johnson. Cycle time properties of the fddi token ring protocol. *IEEE Transactions on Software Engineering*, 13(3):376–385, March 1987.

[Tzo92] P. Tzounakis. Verification of real-time systems: The extension of cospan in dense time. Master's thesis, University of Crete Dept. of Computer Science, September 1992.

[Yov92] S. Yovine. Compiling timed algebras into timed automata. In *At XVIII Conferencia Latinoamericana de Informatica, PANEL' 92*, August 1992.

[Yov93] S. Yovine. *Méthodes et Outils pour la Vérification Symbolique de Systèmes Temporisés.* PhD thesis, Institut National Polytechnique de Grenoble, May 1993.