ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΚΥΠΡΟΥ

Παραδοτέο 3: Τυπικές μέθοδοι περιγραφής και ανάλυσης στοχαστικών συστημάτων

ΕΝΔΙΚΤΗΣ, Αρ. Πρ. 24/2001, ΠΕΝΕΚ, ΙΠΕ

1. ΕΙΣΑΓΩΓΉ ... 3
1.1. ΤΥΠΙΚΈΣ ΜΈΘΟΔΟΙ (FORMAL METHODS) ... 3
1.2. ΑΝΑΛΥΣΗ ΤΗΣ ΕΠΙΔΟΣΗΣ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΣΤΟΧΑΣΤΙΚΑ ΣΥΣΤΗΜΑΤΑ ... 3
1.3. ΣΤΟΧΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ ... 6
2. ΆΛΓΕΒΡΕΣ ΔΙΕΡΓΑΣΙΩΝ (PROCESS ALGEBRAS) ... 7
2.1. ΧΡΟΝΙΚΈΣ ΆΛΓΕΒΡΕΣ ΔΙΕΡΓΑΣΙΏΝ ... 9
2.2. ΠΙΘΑΝΟΤΙΚΈΣ ΆΛΓΕΒΡΕΣ ΔΙΕΡΓΑΣΙΏΝ ... 9
2.3. ΣΤΟΧΑΣΤΙΚΈΣ ΆΛΓΕΒΡΕΣ ΔΙΕΡΓΑΣΙΏΝ ... 10
2.3.1. TIPP - TImed Processes and Performance analysis ... 12
2.3.2. PEPA - Process Algebra for Performance Modelling ... 13
2.3.3. EMPA - Extended Markovian Process Algebra ... 14
2.4. ΤΥΧΑΊΕΣ ΜΕΤΑΒΛΗΤΕΣ ΚΑΙ ΣΥΝΑΡΤΉΣΕΙΣ ΚΑΤΑΝΟΜΏΝ ... 15
2.5. ΣΤΟΧΑΣΤΙΚΕΣ ΆΛΓΕΒΡΕΣ ΚΑΙ ΣΥΝΕΧΕΙΣ ΚΑΤΑΝΟΜΕΣ ... 17
ΣΥΜΠΕΡΑΣΜΑ ... 19
3. EPACSR ... 19
3.1. Η ΣΥΝΤΑΞΗ ΤΟΥ ΜΟΝΤΈΛΟΥ ... 20
3.1.1. Διεργασίες ... 22
3.2. ΣΗΜΑΣΙΟΛΟΓΊΑ ΤΟΥ ΜΟΝΤΈΛΟΥ ... 23
4. ΙΣΧΥΡΗ ΔΥΠΡΟΣΟΜΟΙΩΣΗ ... 27
4.1. ΤΟ ΜΟΝΤΈΛΟ ... 28
4.2. ΙΣΧΥΡΉ ΔΥΠΡΟΣΟΜΟΊΩΣΗ ... 29
4.3. Ο ΑΛΓΟΡΙΘΜΟΣ ... 30
4.3.1. Διαδικασία για καταστάσεις με μη-ντετερμινιστική συμπεριφορά ... 32
4.3.2. Διαδικασία για καταστάσεις με στοχαστική και πιθανοτική συμπεριφορά ... 33
4.3.3. Ο ολοκληρωμένος αλγόριθμος ... 36
5. ΑΛΥΣΙΔΕΣ MARKOV ΚΑΙ ΜΟΝΤΕΛΟΕΛΕΓΧΟΣ ... 37
5.1. ΑΛΥΣΙΔΕΣ ΜΑΡΚΟΒ ΔΙΑΚΡΙΤΟΥ ΧΡΟΝΟΥ ... 39
5.2. ΣΥΝΕΧΗΣ ΧΡΟΝΟΥ ΑΛΥΣΙΔΕΣ ΜΑΡΚΟΒ ... 41
5.3. ΠΑΡΟΔΙΚΕΣ ΚΑΙ ΜΑΚΡΟΠΡΟΘΕΣΜΕΣ ΠΙΘΑΝΟΤΗΤΕΣ (TP ΚΑΙ SSP) ΓΙΑ ΤΗΝ EPACSR ... 41
5.4. ΜΟΝΤΕΛΟΕΛΕΓΧΟΣ ΓΙΑ ΤΗΝ EPACSR ... 43
6. ΣΥΜΠΕΡΑΣΜΑΤΑ ... 45
ΒΙΒΛΙΟΓΡΑΦΙΑ ... 46
ΠΑΡΑΡΤΗΜΑ Α ... 49

1. Εισαγωγή

1.1. Τυπικές Μέθοδοι (Formal Methods)

Οι τυπικές μέθοδοι είναι μία συλλογή από σημειογραφίες και τεχνικές οι οποίες χρησιμοποιούνται για την περιγραφή και την ανάλυση συστημάτων. Αυτές οι μέθοδοι είναι τυπικές (formal) με την έννοια ότι στηρίζονται πάνω σε μαθηματικές θεωρίες, όπως η λογική, τα αυτόματα ή η θεωρία γράφων και στόχο έχουν να βελτιστοποιήσουν την ποιότητα του συστήματος [1, 2].

Οι τεχνικές των τυπικών προδιαγραφών ξεκινούν από μια ακριβή περιγραφή του συστήματος, σημείο πολύ σημαντικό στο να συγκεκριμενοποιηθεί η συμπεριφορά του και να καταστεί δυνατή η σωστή ανάλυσή του. Μπορούν να χρησιμοποιηθούν από τις αρχικές φάσεις κατασκευής του συστήματος, όπως για παράδειγμα, τη φάση του σχεδιασμού, και να χρησιμοποιηθούν στην επαλήθευση των προδιαγραφών του συστήματος, στη συστηματική αναζήτηση περιπτώσεων όπου το σύστημα αποτυγχάνει να τις ικανοποιήσει, ή ακόμα στην αξιολόγηση της ποιότητας υπηρεσιών που μπορεί να προσφέρει..

Οι αρχικές έρευνες πάνω στις τυπικές μεθόδους επικεντρώνονταν στην επιβεβαίωση της ορθότητας των συστημάτων. Με τον όρο αυτό εννοούμε ότι το σύστημα θα ικανοποιεί τις προτεινόμενες προδιαγραφές. Η επαλήθευση λογισμικού (software verification) ήταν από τις κυριότερες τυπικές μεθόδους που μελετήθηκαν. Αρχικός στόχος τέτοιων τεχνικών ήταν η τυπική επαλήθευση (formal verification) συστημάτων κριτικής ασφάλειας, όπως π.χ. ο πύργος ελέγχου στα αεροδρόμια. Στην συνέχεια, προτάθηκε όπως οι αυτές οι μέθοδοι επαλήθευσης χρησιμοποιηθούν στο να βοηθήσουν την ανάπτυξη λογισμικών και υλικών συστημάτων. Σύμφωνα λοιπόν με την εισήγηση αυτή, τα προγράμματα έπρεπε να εκλεπτυνθούν με τέτοιο τρόπο ώστε ο έλεγχος να γίνεται βήμα-βήμα, ξεκινώντας από τις προδιαγραφές και τελειώνοντας με τον κυρίως κώδικα.

1.2. Ανάλυση της Επίδοσης Συστημάτων και Στοχαστικά Συστήματα

Τα μεταφορικά και πληροφοριακά συστήματα μπορούν να χωριστούν σε δύο βασικές κατηγορίες:

- Συστήματα πραγματικού χρόνου (π.χ. συστήματα για έλεγχο διεργασιών, συστήματα κατασκευής, ρομπότ, συστήματα αβιονικού ελέγχου), και
- Συστήματα διαμοιρασμού πόρων (resource-sharing systems) (π.χ. υπολογιστές, τηλεφωνικά συστήματα και συστήματα μετάδοσης πληροφοριών, συστήματα παραγωγής, κλπ)

Ενώ ο κύριος στόχος κατά τη δημιουργία συστημάτων πραγματικού χρόνου είναι η παροχή εγγυήσεων σχετικά με την ορθότητα στην εσωτερική επικοινωνία ανάμεσα σε ένα πλήθος διεργασιών, σε συστήματα όπου υπάρχει διαμοιρασμός πόρων, η οικονομική χρήση των πόρων του συστήματος είναι κύριας σημασίας. Κατά συνέπεια, σε συστήματα πραγματικού χρόνου είναι σκόπιμη η ανάλυση ιδιοτήτων ασφάλειας και ελέγχου της συμπεριφοράς, ενώ, σε συστήματα διαμοιρασμού πόρου είναι πιο ωφέλιμο να μετρήσουμε ποσότητες όπως βαθμός χρήσης των πόρων, καθυστερήσεις, πιθανότητες χασίματος μηνυμάτων, και ρυθμοαπόδοση. Βάσει αυτού του διαχωρισμού, μπορούμε ανάλογα να διαχωρίσουμε τα διάφορα πρότυπα ανάλυσης και μοντελοποίησης συστημάτων ως εξής:

- Ντετερμινιστική χρονική ανάλυση για συστήματα πραγματικού χρόνου: μεταβάσεις διαδραματίζονται μέσα σε διακριτές χρονικές στιγμές ή μέσα σε καθορισμένα χρονικά διαστήματα. Μοντέλα που επιτρέπουν την περιγραφή τέτοιων συστημάτων περιλαμβάνουν χρονικά αυτόματα, χρονικές άλγεβρες διεργασιών, χρονικά δίκτυα Petri, και, γενικά, μηχανές πεπερασμένων καταστάσεων επεκταμένες με χρόνο.
- Πιθανοτική χρονική ανάλυση για συστήματα διαμερισμού πόρων: λάθη και μαζικά φαινόμενα μπορούν να οδηγήσουν σε εμφάνιση γεγονότων σε τυχαίες χρονικές στιγμές ή διαστήματα. Διαδεδομένα μοντέλα για ανάλυση τέτοιων συστημάτων είναι τα queuing networks, στοχαστικά δίκτυα Petri και στοχαστικά γραφικά μοντέλα. Οι στοχαστικές άλγεβρες διεργασιών αρχίζουν να αποδεικνύονται ως χρήσιμες επιλογές.

Ο στόχος της ανάλυσης της επίδοσης συστημάτων (performance evaluation) είναι η εξέταση και η βελτίωση ποσοτικών συμπεριφορών ενός συστήματος που σχετίζονται και μεταβάλλονται με το χρόνο. Συστήματα που επωφελούνται τέτοιας ανάλυσης περιλαμβάνουν συστήματα επεξεργασίας και μεταφορών όπως επίσης και μοντέρνα πληροφοριακά συστήματα όπως τα δικτυακά πρωτόκολλα. Κατά την ανάλυση της επίδοσης ενός συστήματος, μετρείται και μοντελοποιείται η χρονική συμπεριφορά συστημάτων, ορίζονται και προσδιορίζονται χαρακτηριστικές ποσότητες επίδοσης και αναπτύσσονται κανόνες οι οποίοι εγγυούνται ένα ικανοποιητικό επίπεδο ποιότητας υπηρεσιών. Στα πλείστα συστήματα υπό μελέτη υπάρχει για οικονομικούς λόγους διαμερισμός πόρων, δηλαδή, υπάρχει ένας μεταβλητός αριθμός απαιτήσεων (π.χ. χρηστών) οι οποίοι ανταγωνίζονται για τη χρήση των ίδιων πόρων. Προφανείς συνέπειες αυτού του γεγονότος περιλαμβάνουν παρεμβάσεις, καθυστερήσεις, και αβέβαιη ποιότητα υπηρεσίας. Επιπλέον, λάθη στη μεταφορά πληροφοριών, βλάβες στο επίπεδο των πόρων επίσης επηρεάζουν τη συμπεριφορά των συστημάτων.

Η έννοια των στοχαστικών διεργασιών επιτρέπει την ακριβή μοντελοποίηση και εξέταση των φαινομένων που αναφέραμε πιο πάνω και η χρήση τους έχει αρχίσει να διαδίδεται δίνοντας ήδη θετικά αποτελέσματα στην περιοχή των τηλεφωνικών συστημάτων. Η ανάγκη όμως για δημιουργία ισχυρών θεωρητικών βάσεων και κατάλληλων προτύπων και εργαλείων μέσω των οποίων θα μπορέσουν να εφαρμοστούν στα πολύπλοκα και προχωρημένα συστήματα που διαδίδονται όλο και περισσότερο στην καθημερινή μας ζωή ολοένα και αυξάνεται. Τα κύρια κενά που αντιμετωπίζονται είναι τα εξής:

- Για το σχεδιασμό πολύπλοκων συστημάτων απαιτούνται ιεραρχικές γλώσσες σχεδιασμού. Οι κύριες μέθοδοι που χρησιμοποιούνται για εκτίμηση επίδοσης συστημάτων δεν περιλαμβάνουν τέτοιες γλώσσες. Επιπλέον, δεν επιτρέπουν το αυτοματοποιημένο κτίσιμο του συνόλου καταστάσεων του μοντέλου του συστήματος, ή τη μείωση του συνόλου αυτού, και οι ειδικοί αναγκάζονται να υλοποιούν μετατροπές και μελέτη του συστήματος στο χέρι.

- Κατά τη μοντελοποίηση παράλληλων και κατανεμημένων συστημάτων οι υπάρχουσες τεχνικές αγνοούν σε μεγάλο βαθμό τις σχέσεις και αλληλεπιδράσεις ανάμεσα στα υποσυστήματα.

- Το συναρτησιακό κομμάτι του συστήματος μελετάται ξεχωριστά από την εκτίμηση της επίδοσης αυτού.

Οι άλγεβρες διεργασιών και τα ιδιαίτερα τους χαρακτηριστικά προσφέρουν ένα πρότυπο εμπλουτισμένο με διάφορες έννοιες και τεχνικές που μπορούν να βοηθήσουν στο να ξεπεραστούν τα πιο πάνω προβλήματα [2-4]. Συγκεκριμένα περιλαμβάνουν μία ιεραρχική μαθηματική γλώσσα για περιγραφή πολύπλοκων συστημάτων. Η γλώσσα αυτή είναι δυνατόν να συλλαμβάνει τόσο τη χρήση πόρων, όσο και την επικοινωνία ανάμεσα στα υποσυστήματα του συστήματος, Μέσω της αυστηρής σημασιολογίας τους συστήματα γραμμένα σε αυτές μπορούν να μεταφραστούν σε μαθηματικά αντικείμενα (διάφοροι τύποι αλυσίδων Markov) τα οποία μπορούν να μετατραπούν σε μικρότερα ισοδύναμα συστήματα, να αναλυθούν και να δώσουν στο χρήστη εγγυήσεις τόσο για τη συναρτησιακή συμπεριφορά όσο και για την επίδοση του συστήματος που μοντελοποιούν.

1.3. Στόχος της εργασίας

Στην εργασία αυτή προτείνουμε μια στοχαστική άλγεβρα διεργασιών, τη EPACSR, για τη μοντελοποίηση και ανάλυση δικτυακών συστημάτων. Η άλγεβρα αυτή περιλαμβάνει τις έννοιες του πόρου, του πιθανοτικού σφάλματος πόρων καθώς και της στοχαστικής συμπεριφοράς μέσω εκθετικών κατανομών. Οι εκθετικές κατανομές χρησιμοποιούνται για να ορίσουν τη διάρκεια ενεργειών που μπορούν να συμβούν σε ένα σύστημα. Η σημασιολογία της EPACSR δίνεται μέσω ενός συνόλου από κανόνες οι οποίοι μεταφράζουν διεργασίες της άλγεβρας σε συστήματα μεταβάσεων που περιέχουν τόσο πιθανοτικές όσο και στοχαστικές μεταβάσεις. Δηλαδή, το μοντέλο αποτελεί ένα υβρίδιο των διακριτών αλυσίδων Markov και των συνεχών αλυσίδων Markov. Ορίζουμε για το μοντέλο αυτό την έννοια της ισχυρής δυπροσομοίωσης και τέλος προτείνουμε τεχνικές μοντελο-ελέγχου και εκτίμησης της επίδοσης.

Στο επόμενο κεφάλαιο κάνουμε μια γενική ανασκόπηση του προτύπου των αλγεβρών διεργασιών και αναφερόμαστε στις κύριες ΑΔ που σχετίζονται με τη EPACSR. Στο Κεφάλαιο 3 παρουσιάζουμε τη σύνταξη και σημασιολογία της EPACSR ενώ στο Κεφάλαιο 4, ορίζουμε την έννοια της ισχυρής δυπροσομοίωσης για το μοντέλο και προτείνουμε ένα αλγόριθμο ο οποίος, με δεδομένο εισόδου δύο διεργασίες αποφασίζει κατά πόσο είναι ισοδύναμες. Στο Κεφάλαιο 5, παραθέτουμε τις μεθόδους ανάλυσης που προτείνουμε για το μοντέλο, ενώ τελειώνουμε στο Κεφάλαιο 6 με κάποια συμπεράσματα και κατευθύνσεις για μελλοντική έρευνα.

2. Άλγεβρες Διεργασιών (Process Algebras)

Οι Άλγεβρες Διεργασιών [5, 6] είναι φορμαλισμοί (φορμαλιστικές μέθοδοι) που χρησιμοποιούνται για τη μοντελοποίηση της συμπεριφοράς συστημάτων. Οι τεχνικές και οι αλγόριθμοι που χρησιμοποιούνται από τις άλγεβρες αυτές μπορούν να χρησιμοποιηθούν για να δείξουν ότι δύο μοντέλα συστημάτων σχετίζονται με κάποιο ακριβή καθορισμένο τρόπο, ότι δηλαδή το ένα προσομοιώνει το άλλο. Αυτό μπορεί να φανεί χρήσιμο αρκετές φορές, στην προσπάθεια μας να δείξουμε ότι ένα σύστημα αποτελεί την εφαρμογή (εκτέλεση) ή την βελτιστοποίηση κάποιου άλλου ή ακόμα για σμίκρυνση του μοντέλου που αντιστοιχεί σε κάποιο σύστημα.

Τα κύρια πλεονεκτήματα των αλγεβρών διεργασιών συνοψίζονται ως εξής:

- Οι γλώσσες αποτελούν το κυριότερο μέσο για το σχεδιασμό του hardware και του software. Οι άλγεβρες διεργασιών είναι αφαιρετικές γλώσσες (abstract languages) προσαρμοσμένες στην περιγραφή παράλληλων και κατανεμημένων συστημάτων.

- Η οργάνωση/δόμηση αποτελεί τον πιο σωστό τρόπο για την αντιμετώπιση των πολύπλοκων συστημάτων. Οι άλγεβρες διεργασιών προσφέρουν μια σχεδιαστική μεθοδολογία (constructivity), που επιτρέπει το συστηματικό σχεδιασμό πολύπλοκων συστημάτων από μικρότερα (υπό)συστήματα. Επίσης, οι άλγεβρες διεργασιών προσφέρουν αλγεβρικούς χαρακτηρισμούς υπεύθυνους για την ισοδυναμία της συμπεριφοράς των συστημάτων (equivalence of system behavior)

- Συνδυάζοντας τη στοχαστική μοντελοποίηση και ανάλυση με τις καλά διερευνημένες φορμαλιστικές τεχνικές περιγραφής των άλγεβρες διεργασιών, προϊδεάζουμε ένα πλούσιο έδαφος για περαιτέρω ανάπτυξη και έρευνα στον τομέα της μοντελοποίησης των παράλληλων/κατανεμημένων συστημάτων συμπεριλαμβανομένων των δικτύων υψηλής ταχύτητας, των δικτύων κινητής τηλεφωνίας κτλ.

Παραδοσιακά, οι άλγεβρες διεργασιών ασχολούνταν με το λειτουργικό κομμάτι των συστημάτων, όπως την εξωτερική τους συμπεριφορά, τον έλεγχο ροής, τον συγχρονισμό και την επικοινωνία μεταξύ των συστημάτων. Σε όλα τα πιο πάνω στοιχεία, ο χρόνος δεν λαμβανόταν υπόψη. Γνωστές Άλγεβρες Διεργασιών, όπως η CCS [2], η CSP [3] και η ACP[4], αναπτύχθηκαν κυρίως για τους πιο πάνω λόγους:

για να περιγράψουν και να αναλύσουν συστήματα που επικοινωνούν μεταξύ τους και λειτουργούν παράλληλα. Στηρίζονταν δηλαδή στην ιδέα ότι τα δύο πιο σημαντικά δομικά στοιχεία για την κατανόηση των πολύπλοκων δυναμικών συστημάτων είναι ο παραλληλισμός και η μεταξύ τους επικοινωνία. Και για να το πετύχουν αυτό, βασίστηκαν πάνω στην ιδέα ότι, μπορεί κάποιος να επαληθεύσει ολόκληρο το σύστημα με το να αναλύσει και να περιγράψει τα δομικά του στοιχεία.

Τις τελευταίες δεκαετίες όμως, καθώς το ενδιαφέρον μεγάλωνε, οι άλγεβρες που προαναφέραμε, επεκτάθηκαν περισσότερο με την προσθήκη ποσοτικών πληροφοριών, όπως ο χρόνος και η πιθανότητες. Έτσι έχουμε την εμφάνιση των **Χρονικών και Πιθανοτικών Αλγεβρών Διεργασιών** [7-11], οι οποίες βοήθησαν πάρα πολύ στην περιγραφή συστημάτων πραγματικού χρόνου (real-time systems) και κυρίως στην ανάλυση της ποιότητας που προσφέρουν τα συστήματα.

Οι μελέτες όμως και οι ανάγκες για πιο ακριβή περιγραφή και ανάλυση των συστημάτων, οδήγησαν στη ανάγκη για περαιτέρω επέκταση των Αλγεβρών. Έτσι οι νέες άλγεβρες επεκτάθηκαν ακόμη περισσότερο με την εισαγωγή της έννοιας του στοχαστικού χρόνου. Συνεπώς, οι νέες άλγεβρες ονομάστηκαν **Στοχαστικές Άλγεβρες Διεργασιών** [14-20], στις οποίες ο χρόνος και η πιθανότητα αφομοιώνονται με τέτοιο τρόπο έτσι ώστε οι οποιεσδήποτε καθυστερήσεις να ακολουθούν μία συνεχόμενη πιθανοτική δομή. Ορισμένα παραδείγματα που απαιτούν τέτοιες βελτιώσεις είναι:

- Τα επικοινωνιακά πρωτόκολλα τα οποία απαιτούσαν μηχανισμούς time-out. Σε σχέση με την επιλεγμένη χρονική τιμή, ένα πρωτόκολλο μπορούσε να οδηγήσει το σύστημα σε αδιέξοδο ή όχι.

- Αναλύοντας την επίδοση των παράλληλων επεξεργαστών και των κατανεμημένων συστημάτων, έπρεπε οπωσδήποτε να λάβουμε υπόψη τις λειτουργικές εξαρτήσεις και το συγχρονισμό των διαφόρων επιμέρους εργασιών. Άρα, οι χρόνοι εκτέλεσης των διαφόρων προγραμμάτων, η χρησιμοποίηση των πόρων (resources), και οι speedup values εξαρτιόνταν σημαντικά από τη χρησιμοποίηση αυτών των νέων στοιχείων [21].

- Τα κέντρα εξυπηρέτησης πελατών, όπου εξυπηρετούνται πελάτες με διαφορετική προτεραιότητα ο καθένας .

- Η πιθανότητα ενός επικοινωνιακού καναλιού να τεθεί εκτός λειτουργίας.
- Ο ρυθμός επίδοσης και η χρησιμοποίηση των κέντρων εξυπηρέτησης

Στην συνέχεια θα αναφερθούμε ξεχωριστά στις διάφορες επεκτάσεις που πραγματοποιήθηκαν: στις χρονικές, πιθανοτικές και στοχαστικές άλγεβρες διεργασιών.

2.1. Χρονικές Άλγεβρες Διεργασιών

Οι χρονικές επεκτάσεις στις Άλγεβρες Διεργασιών απέκτησαν μεγάλο ενδιαφέρον τα τελευταία χρόνια, κυρίως γιατί ο χρόνος αποτελεί βασικό στοιχείο στην προσπάθεια μας να περιγράψουμε, να αναλύσουμε και να βγάλουμε συμπεράσματα πάνω στην συμπεριφορά συστημάτων που λειτουργούν σε συνθήκες πραγματικού χρόνου (real-time systems).

Η βασική ιδέα αυτών των αλγεβρών, είναι η προσθήκη ενός χρονικού τελεστή (t).P, ο οποίος δηλώνει ότι η διεργασία P εκτελείται μετά από καθυστέρηση διάρκειας t μονάδων. Στην προσθήκη αυτή λαμβάνονται υπόψη τόσο οι διακριτού χρόνου όσο και οι πραγματικού χρόνου παραλλαγές, ανάλογα με το πεδίο τιμών του χρόνου t.

Η κύρια διαφορά μεταξύ των διαφόρων Χρονικών Αλγεβρών Διεργασιών αφορά την ερμηνεία που δίνεται όσον αφορά την εκτέλεση των ενεργειών (actions), δηλαδή το πότε μία ενέργεια λαμβάνει χώρα. Υπάρχουν περιπτώσεις όπου μία ενέργεια εκτελείται μετά από μία συγκεκριμένη χρονική καθυστέρηση, ενώ μπορεί να καθυστερήσει ακόμη περισσότερο λόγω συγχρονισμού. Σε κάποιες άλλες περιπτώσεις, οι ενέργειες εκτελούνται μόλις τους δοθεί η δυνατότητα να το πράξουν (αυτό συμβαίνει κυρίως σε εσωτερικές ενέργειες αφού αυτές δεν επηρεάζονται από το εξωτερικό περιβάλλον και συνεπώς δεν υπάρχει λόγος καθυστέρησης). Παράδειγμα χρονικών αλγεβρών είναι η TPCCS, η CCSR και η ACSR.

2.2. Πιθανοτικές Άλγεβρες Διεργασιών

Οι πιθανοτικές επεκτάσεις στις αρχικές Άλγεβρες Διεργασιών παρατηρήθηκαν, όπως και στις χρονικές, την τελευταία δεκαετία, κυρίως πάνω σε γνωστές άλγεβρες όπως η ACP, η CCS, CSP κ.α. Η βασική ιδέα μίας πιθανοτικής Άλγεβρας Διεργασιών κρύβεται γύρω στην προσθήκη ενός πιθανοτικού τελεστή, ο οποίος μπορεί να δηλωθεί, ως $P +_p Q$ με $p \in (0,1)$, όπου η P μπορεί να επιλεγεί με πιθανότητα p και η Q με πιθανότητα 1-p.

Διάφορα μοντέλα έχουν προταθεί ως Πιθανοτικές Άλγεβρες Διεργασιών, ανάλογα με το αν επιτρέπουν μη-ντετερμινιστικές μεταβάσεις ή όχι. Οι πιθανοτικές επιλογές αποτελούν μέρος των εσωτερικών λειτουργιών των διεργασιών, αφού πραγματοποιούνται χωρίς την επίδραση του (εξωτερικού) περιβάλλοντος.

Σε αρκετές περιπτώσεις μία πιθανοτική επιλογή (probabilistic choice) ορίζεται ως μία πιθανοτική κατανομή σ' ένα σύνολο από δυναμικά 'επιτυχής' (successor) καταστάσεις ή διεργασίες. Οι επιλογές αυτές αρκετές φορές διευκολύνουν τους σχεδιαστές συστημάτων από το να αποφύγουν να ασχοληθούν με τις λεπτομέρειες του πώς πραγματοποιούνται οι επιλογές ανάμεσα στις καταστάσεις, χωρίς ωστόσο να παραλείπουν τις απαραίτητες πληροφορίες για το αποτέλεσμα των επιλογών.

Τέλος, σε κάποιες άλλες άλγεβρες διεργασιών, όπως θα δούμε στη συνέχεια, η έννοια της πιθανότητας συνδέεται με άλλα χαρακτηριστικά, εκτός από την επιλογή καταστάσεων ή διεργασιών. Συγκεκριμένα, στην PACSR δίνεται η δυνατότητα να δικαιολογήσουμε το πότε ένας πόρος είναι διαθέσιμος ή όχι. Με λίγα λόγια σε κάθε πόρο αντιστοιχεί κάποια πιθανότητα p ούτως ώστε ο πόρος να μπορεί να χρησιμοποιηθεί και κάποια πιθανότητα (1 – p) ο πόρος να είναι εκτός λειτουργίας.

2.3. Στοχαστικές Άλγεβρες Διεργασιών

Το βασικό κίνητρο πάνω στο οποίο στηρίχθηκε η ανάπτυξη των στοχαστικών αλγεβρών διεργασιών, είναι η ακριβής περιγραφή και μελέτη της συμπεριφοράς συστημάτων διαμερισμού πόρων – σε αντίθεση με τις χρονικές άλγεβρες που αναφέρονται κυρίως σε συστήματα πραγματικού χρόνου. Κύριο χαρακτηριστικό στις άλγεβρες αυτές, είναι η καθυστέρηση που παρατηρείται κατά την διάρκεια της εκτέλεσης των ενεργειών, η οποία ακολουθεί τώρα μία στοχαστική κατανομή, σε αντίθεση με τις χρονικές άλγεβρες όπου ο χρόνος ήταν διακριτός. Έτσι έχουμε τον ορισμό των χρονικά συνεχόμενων τυχαίων μεταβλητών, οι οποίες μπορούν να χαρακτηρίσουν χρονικά γεγονότα (time instants) καθώς και την διάρκεια των ενεργειών.

Τα χαρακτηριστικά των στοχαστικών αλγεβρών, εξαρτώνται κυρίως από τις κλάσεις των τυχαίων κατανομών που εφαρμόζονται στην περιγραφή των συστημάτων. Αρκετές προσπάθειες έγιναν για την ενσωμάτωση τυχαίων μεταβλητών που ακολουθούσαν την γενική κατανομή, όμως αυτό απέτυχε, διότι η κατανομή αυτή οδηγούσε σε αποτελέσματα που δεν μπορούσαν να αναλυθούν ικανοποιητικά. Έτσι αποφασίστηκε να χρησιμοποιηθεί η (αρνητική) εκθετική κατανομή, η οποία και έλυσε αρκετά προβλήματα, μιας και επέτρεψε την ακριβή περιγραφή αρκετών καταστάσεων των resource-sharing συστημάτων. Παράδειγμα στοχαστικών αλγεβρών είναι η TIPP [22], η PEPA [18], η EMPA [12], η MAC [19] κ.ά.

Στις αρχές της δεκαετίας του 90, ο Herzog πρότεινε τον προπομπό των στοχαστικών αλγεβρών διεργασιών, την TIPP (**Ti**med **P**rocesses and **P**erformance analysis) [22], η οποία αποτελεί προέκταση της CPS. Ανεξάρτητα από την δουλεία του Herzog, η Hillston δημιούργησε μια προέκταση της CSP με την προσθήκη στοχαστικών καθυστερήσεων, καταλήγοντας έτσι στην PEPA (**P**erformance **E**valuation **P**rocess **A**lgebra) [18]. Άλλες προτεινόμενες άλγεβρες που αναπτύχθηκαν μετά από αυτά, ήταν οι άλγεβρες διεργασιών του Buchholz και του Bernardo. Η όλη δουλεία του τελευταίου οδήγησε στην δημιουργία της EMPA (**E**xtended **M**arkovian **P**rocess **A**lgebra) [12], στην οποία εκτός από τις ενέργειες που καθυστερούσαν με βάση κάποια εκθετική κατανομή – όπως στην TIPP και την PEPA – ενσωματώθηκαν οι έννοιες των στιγμιαίων ενεργειών (immediate actions), του μη-ντετερμινισμού (non-deterministic), των προτεραιοτήτων (priorities) και των βαρών (weights).

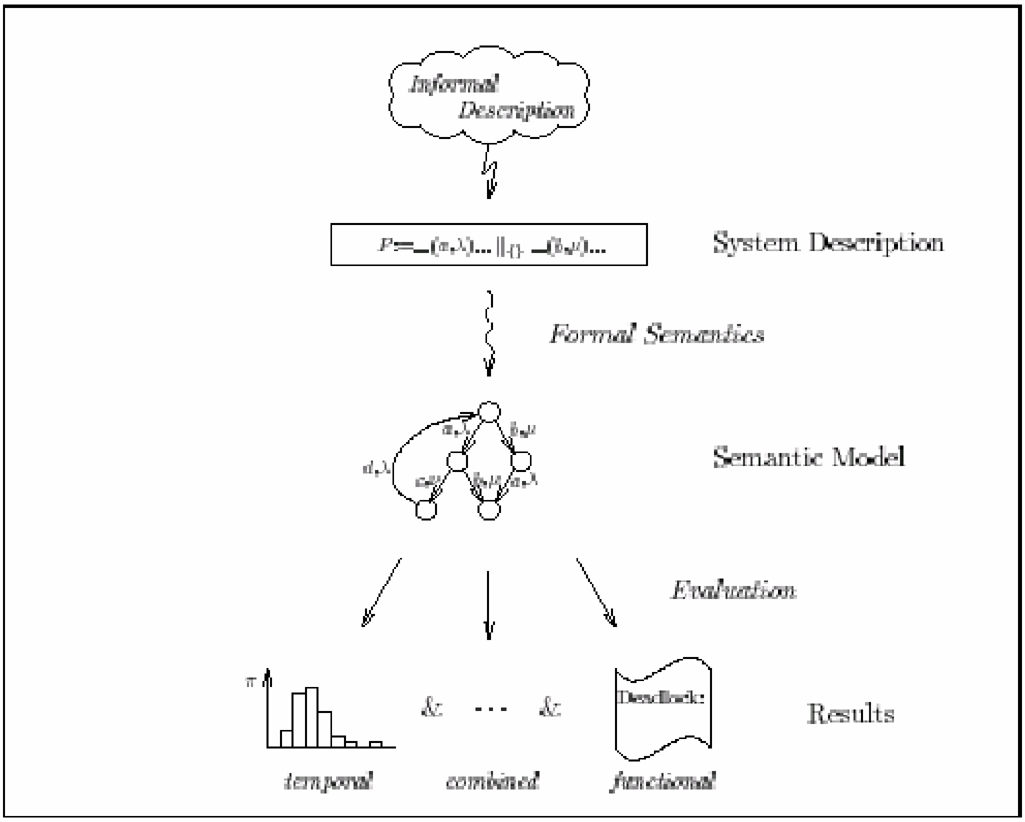
Όλες αυτές οι προσεγγίσεις είχαν ως κοινό σημείο το γεγονός ότι το υπόβαθρο της σημασιολογίας τους συνδεόταν στενά με τις αλυσίδες συνεχούς-χρόνου του Markov (continuous-time Markov chains), με τη προσθήκη ετικετών για τις ενέργειες. Σε αντίθεση με τα κλασσικά συστήματα ουρών και τις στοχαστικές μεταβλητές στα Petri nets, οι στοχαστικές άλγεβρες TIPP, PEPA και EMPA επιτρέπουν τον προσδιορισμό και την δημιουργία πολύπλοκων αλυσίδων Markov με τρόπο συνθετικό δηλαδή χωρίζοντας το σύστημα σε επί μέρους κομμάτια, αναλύοντας τα και συνδέοντας στο τέλος.

Στη συνέχεια θα κάνουμε μία μικρή αναφορά στις τρεις πιο κύριες στοχαστικές άλγεβρες διεργασιών, την TIPP, την PEPA και την EMPA.

2.3.1. TIPP - TImed Processes and Performance analysis

Βασική ιδέα της TIPP είναι η ανάπτυξη μίας στοχαστικής άλγεβρας διεργασιών με την ενσωμάτωση τυχαίων μεταβλητών στις κλασσικές προσεγγίσεις των αλγεβρών.

Πιο αναλυτικά, ο τρόπος ανάπτυξης συνοψίζεται στο πιο κάτω σχήμα:



Σχήμα 2 : Διαδικασία Μοντελοποίησης

Στην άλγεβρα αυτή θα λέγαμε ότι ακολουθείται ο τρόπος σκέψης των κλασσικών αλγεβρών διεργασιών, με την περιγραφή της συμπεριφοράς του συστήματος από μία ‘αφαιρετική’ γλώσσα (abstract language) – system description – έχοντας ως υπόβαθρο ένα γράφο μεταβάσεων των καταστάσεων (state-transition graph), ο οποίος αντιπροσωπεύει το ακριβές νόημα της διεργασίας που μελετούμε (semantic model). Πλέον, αντί να λαμβάνεται υπόψη μόνο η λειτουργική συμπεριφορά του συστήματος, προσθέτουμε και πληροφορίες που σχετίζονται με τον χρόνο: κάθε δραστηριότητα περιγράφεται από ένα ζεύγος που περιλαμβάνει παραμέτρους για τον τύπο της και την διάρκειά της. Το κύριο χαρακτηριστικό όμως των διαφόρων χρονικών παραμέτρων είναι το γεγονός ότι βασίζονται πάνω σε στοχαστικές υποθέσεις, π.χ. Markovian, semi-Markovian. Ως αποτέλεσμα, αυτές οι πρόσθετες πληροφορίες στην σημασιολογία της νέας άλγεβρας επέτρεψαν την αξιολόγηση αρκετών χαρακτηριστικών του συστήματος όπως:

- Λειτουργική Συμπεριφορά (π.χ. αδιέξοδα ή ζωτικότητα)
- Χρονική Συμπεριφορά (ρυθμός επίδοσης ή χρόνος αναμονής)
- Συνδυασμένες Ιδιότητες (όπως η διάρκεια καθορισμένων γεγονότων που εκτελούνται σειριακά, η πιθανότητα να βρεθεί το σύστημα σε αδιέξοδο ή εκτός χρονικών ορίων κτλ.)

Τέλος, να αναφέρουμε πως παρόλο που υπήρχε η πρόθεση, αρχικά, να χρησιμοποιηθούν στην TIPP αυθαίρετες κατανομές, τελικά η όλη φιλοσοφία και θεωρία της γλώσσας οδήγησαν στο να εξελιχθεί σε μία στοχαστική άλγεβρα διεργασιών που υποστήριζε εκθετικές κατανομές. Η TIPP πολλές φορές αναφέρεται και ως Μαρκοβιανή Άλγεβρα Διεργασιών.

2.3.2. PEPA - Process Algebra for Performance Modelling

Η PEPA αναπτύχθηκε κυρίως για να ερευνήσει το πώς τα συνθετικά στοιχεία των αλγεβρών διεργασιών μπορούσαν να έχουν επίδραση πάνω στην πρακτική υλοποίησης της μοντελοποίησης της επίδοσης.

Η αξιολόγηση της επίδοσης ασχολείται με την περιγραφή, την ανάλυση και την βελτιστοποίηση της δυναμικής συμπεριφοράς των υπολογιστών και των επικοινωνιακών συστημάτων. Αυτό περιλαμβάνει την διερεύνηση της ροής των δεδομένων και τον έλεγχο των πληροφοριών μέσα και μεταξύ των διαφόρων τμημάτων του συστήματος. Ο τελικός στόχος είναι η κατανόηση της συμπεριφοράς του συστήματος και ο εντοπισμός των στοιχείων εκείνων που σχετίζονται με την επίδοση.

Στη μοντελοποίηση της επίδοσης, το μοντέλο του συστήματος χρησιμοποιείται ακριβώς για τον πιο πάνω λόγο, δηλαδή για να εντοπιστούν τα βασικά χαρακτηριστικά του συστήματος που θα βοηθήσουν στην αναπαραγωγή της επίδοσης. Ακολούθως, η μελέτη της επίδοσης θα επικεντρωθεί πάνω σε αντικειμενικούς στόχους και θα διερευνηθούν αρκετές εναλλακτικές λύσεις – κυρίως αλλάζοντας τις μεταβλητές στις παραμέτρους του συστήματος. Το μοντέλο θα αξιολογηθεί για να καθοριστεί η συμπεριφορά του και η απόδοσή του με βάση τις τρέχουσες τιμές των παραμέτρων. Η αξιολόγηση γίνεται με την βοήθεια ενός συνόλου από εξισώσεις με ορισμένες αναλυτικές τεχνικές ή με την με την βοήθεια της προσομοίωσης του μοντέλου. Ακριβώς εδώ εμπλέκεται και η PEPA. Τα αναλυτικά μοντέλα στηρίζονται πάνω σε στοχαστικά μοντέλα τα οποία υποστηρίζει η γλώσσα αυτή - στην ουσία θα λέγαμε ότι οι στοχαστικές διεργασίες που χρησιμοποιούνται είναι συνεχείς-Μαρκοβιανές – και έτσι η χρησιμοποίηση της PEPA για κατασκευή μοντέλων συστημάτων παρείχε αρκετά οφέλη και κίνητρα.

2.3.3. EMPA - Extended Markovian Process Algebra

Η ανάπτυξη της EMPA επηρεάστηκε σε μεγάλο βαθμό από την στοχαστικά χρονική άλγεβρα διεργασιών MTIPP και την PEPA, καθώς και από τον φορμαλισμό των γενικευμένων στοχαστικών Petri nets GSPNs. Ως συνέπεια αυτών, στην EMPA κυριαρχούν τριών ειδών ενεργειών:

- Εκθετικά χρονικές ενέργειες (που πηγάζουν από την MTIPP και PEPA)
- Στιγμιαίες ενέργειες/ γεγονότα που χαρακτηρίζονται από κάποιο βάρος και βαθμό προτεραιότητας (κάτι ανάλογο με αυτό που επικρατεί στα GSPNs)
- Παθητικές ενέργειες (passive actions) (κάτι αντίστοιχο με αυτό που επικρατεί στην PEPA)

Οι εκθετικά χρονικές ενέργειες, περιγράφουν δραστηριότητες που σχετίζονται με την επίδοση. Οι στιγμιαίες ενέργειες με βαθμό προτεραιότητας και βάρος, μοντελοποιούν λογικά γεγονότα καθώς και δραστηριότητες που δεν σχετίζονται με την επίδοση αλλά είναι χρήσιμες στον να εκφράσουν είτε επιλογές που εξαρτώνται από κάποια προτεραιότητα ή πιθανοτικές επιλογές. Τέλος, οι παθητικές ενέργειες μοντελοποιούν δραστηριότητες που είναι έτοιμες να πραγματοποιήσουν κάποιο συγχρονισμό με εκθετικό χρόνο αναμονής ή στιγμιαίες δραστηριότητες και χρησιμοποιούνται για να εκφράσουν μη-ντετερμινιστικές επιλογές.

2.4. Τυχαίες Μεταβλητές και Συναρτήσεις Κατανομών

Η εισαγωγή των τυχαίων μεταβλητών στις άλγεβρες διεργασιών σκοπό είχε να χαρακτηρίζει την οποιαδήποτε διάρκεια ενέργειας (activity duration) στα παράλληλα συστήματα.

Με τον όρο τυχαία μεταβλητή (ενός πειράματος), εννοούμε μία συνάρτηση της οποίας το πεδίο ορισμού είναι ένας χώρος δειγμάτων και το πεδίο τιμών είναι κάποιο σύνολο πραγματικών αριθμών. Έτσι, όταν λέμε ότι η έκβαση ενός πειράματος είναι **s**, η τυχαία μεταβλητή συμβολίζεται με **X(s).** Η τιμής μίας τυχαίας μεταβλητής ποικίλει από δοκιμή σε δοκιμή.

Υπάρχουν δύο είδη τυχαίων μεταβλητών, οι διακριτές (discrete random variables) και οι συνεχείς (continuous random variables). Η διακριτή τυχαία μεταβλητή είναι αυτή που λαμβάνει μόνο ένα διακριτό σύνολο τιμών. Πιο συγκεκριμένα, η τυχαία μεταβλητή X είναι μία διακριτή τυχαία μεταβλητή εάν μπορεί να πάρει μόνο ένα πεπερασμένο αριθμό τιμών σε οποιοδήποτε διάστημα παρατήρησης. Εάν ωστόσο, η μεταβλητή αυτή μπορεί να πάρει οποιαδήποτε τιμή σ' ένα διάστημα παρατήρησης, τότε η X ονομάζεται συνεχής τυχαία μεταβλητή.

Προχωρώντας πιο βαθιά, θα λέγαμε ότι όλες οι τυχαίες μεταβλητές έχουν μία συσσωρευτική συνάρτηση κατανομής (cumulative distribution function) η οποία περιγράφει την πιθανοτική τους συμπεριφορά. Αναλυτικότερα, η συνάρτηση αυτή δίνει την πιθανότητα μίας τυχαίας μεταβλητής **X** να είναι μικρότερη ή ίση από μία τιμή **x**, για κάθε τιμή του x. συμβολίζουμε αυτή την πιθανότητα με:

$$P(X \leq x) \text{ για } -\infty < x < \infty$$

Είναι φανερό ότι αυτή η πιθανότητα είναι συνάρτηση της μεταβλητής x και όχι της τυχαίας μεταβλητής X, ωστόσο εξαρτάται από την αντιστοίχηση της X. Για απλούστευση γράφουμε τον συμβολισμό:

$$F_X(x) = P(X \leq x)$$

Η $F_X(x)$ ονομάζεται και διαφορετικά ως συνάρτηση κατανομής (distribution function) και έχει τις εξής ιδιότητες:

- Είναι φραγμένη μεταξύ μηδέν και ένα

- Είναι μία μονότονη μη-φθίνουσα συνάρτηση του x, δηλ

$F_X(x_1) \le F_X(x_2)$ εάν $x_1 < x_2$

Συχνά, είναι χρήσιμη μία εναλλακτική περιγραφή της κατανομής πιθανότητας της τυχαίας μεταβλητής X. αυτή είναι η παράγωγος της συνάρτησης κατανομής, η οποία γράφεται:

$$f_X(x) = \frac{d}{dx} F_X(x)$$

και ονομάζεται συνάρτηση πυκνότητας πιθανότητας (probability density function). Το όνομα αυτό προέρχεται από το ότι η πιθανότητα του γεγονότος $x_1 < X \le x_2$ ισούται με:

$$P(x_1 < X \le x_2) = P(X \le x_2) - P(X \le x_1) = F_X(x_2) - F_X(x_1) = \int_{x_1}^{x_2} f_X(x)dx$$

Η συνάρτηση πυκνότητας πιθανότητας ικανοποιεί τα εξής:

- Η συνολική πιθανότητα για όλες τις πιθανές τιμές της συνεχούς τυχαίας μεταβλητής X είναι 1

- Δεν μπορεί να είναι αρνητική.

Τελειώνοντας θα αναφερθούμε σε μια συγκεκριμένη κατηγορία τυχαίων κατανομών, αυτή των εκθετικών κατανομών: Μια τυχαία μεταβλητή X καλείται *εκθετικά κατανεμημένη* αν και μόνο αν

$$P(X \le x) = 1 - e^{-\lambda t} \text{ για x } \ge 0\text{, διαφορετικά } P(X \le x) = 0$$

Το λ ονομάζεται ο συντελεστής της κατανομής και χαρακτηρίζει πλήρως την κατανομή. Η τάξη των εκθετικών κατανομών ικανοποιεί κάποιες σημαντικές ιδιότητες οι οποίες δικαιολογούν την ευρεία χρήση της στην περιοχή της εκτίμησης επίδοσης συστημάτων. Αρχικά η μέση τιμή μιας εκθετικής κατανομής με λ είναι 1/λ και επιπλέον είναι η πιο κατάλληλη μεταβλητή για να περιγράψει ένα τυχαίο φαινόμενο του οποίου είναι γνωστή μόνο η μέση τιμή. Τέλος η εκθετική κατανομή είναι η μόνη κατανομή που ικανοποιεί την ιδιότητα έλλειψης μνήμης:

Μια συνεχής κατανομή X ικανοποιεί την *ιδιότητα έλλειψης μνήμης* αν για κάθε t, t'> 0

$$P(X \leq t + t' \mid X > t) = P(X \leq t')$$

2.5. Στοχαστικές Άλγεβρες και Συνεχείς Κατανομές

Έχουν γίνει διάφορες προσπάθειες ενσωμάτωσης συνεχών κατανομών σε στοχαστικές άλγεβρες διεργασιών. Ωστόσο, η μελέτη για το ποιες είναι κατάλληλες για να χρησιμοποιηθούν οδήγησε τους επιστήμονες σε περαιτέρω έρευνες. Ένα από τα κυριότερα αντικείμενα της μελέτης αυτής, ήταν ένα πολύ σημαντικό αξίωμα των αλγεβρών διεργασιών, γνωστό ως Expansion law. Στη συνέχεια. θα αναλύσουμε το αξίωμα αυτό όπως και το πώς συνδέεται με τις διάφορες κατανομές.

Expansion Law

Ένα πολύ γνωστό μαθηματικό μοντέλο με τη βοήθεια του οποίου δίνεται η σημασιολογία στις παραδοσιακές άλγεβρες διεργασιών, είναι το σύστημα μεταβάσεων με ετικέτες (label transition system). Συγκεκριμένα, τα συστήματα μεταβάσεων είναι γράφοι των οποίων οι κόμβοι συμβολίζουν τις καταστάσεις ενός συστήματος και οι ακμές τις μεταβάσεις που μπορούν να γίνουν από μια κατάσταση σε μια άλλη. Οι μεταβάσεις χαρακτηρίζονται από ετικέτες οι οποίες συμβολίζουν τις διάφορες ενέργειες και περιγραφούν τον τρόπο με τον οποίο το σύστημα πραγματοποιεί μετάβαση από την μία κατάσταση στην άλλη. Με αυτό τον τρόπο, η σύνδεση δύο παράλληλων ανεξάρτητων μεταξύ τους ενεργειών, αντιμετωπίζεται ως επιλογή (choice) μεταξύ των δύο πιθανόν σειριακών γεγονότων. Έτσι, στην παράλληλη σύνδεση, οι ενέργειες του ενός μέρους (component) παρεμβάλλονται με αυτές του άλλου – interleaving semantics. Ως αποτέλεσμα των πιο πάνω, η παράλληλη σύνδεση δύο ανεξάρτητων ενεργειών (η οποία συμβολίζεται με τον όρο ||) μπορεί να αναχθεί σε μία επιλογή μεταξύ των δύο (η οποία συμβολίζεται με τον όρο +), όπως φαίνεται και στο πιο κάτω παράδειγμα:

$$\alpha ;p \parallel b;q = \alpha ;(p \parallel b;q) + b;(\alpha ; p\parallel q)$$

όπου α, b ενέργειες και p, q διεργασίες. O κανόνας αυτός, γνωστός ως Expansion Law, είναι ευρύτατα αποδεχτός και όπως αποδεικνύεται, ο ρόλος του είναι πολύ σημαντικός στις άλγεβρες διεργασιών για σκοπούς επαλήθευσης (verification purpose).

Εκθετικές κατανομές και Expansion Law

Η σημασιολογία των Μαρκοβιανών Αλγεβρών Διεργασιών είναι ορισμένη με βάση μίας επέκτασης των συστημάτων μετάβασης. Η κομψότητα της ιδιότητας της έλλειψης μνήμης (memoryless property) των εκθετικών κατανομών, επιτρέπει την ομαλή μετάβαση σε συνθήκες παρεμβολής και έτσι το παράδειγμα που αναφέραμε πιο πάνω μπορεί να γραφτεί ως:

$$\lambda \to p \parallel \mu \to q = \lambda \to (p \parallel \mu \to q) + \mu \to (\lambda \to p \parallel q)$$

όπου λ και μ οι ρυθμοί καθυστέρησης προτού ξεκινήσουν οι διεργασίες p και q αντίστοιχα. Σύμφωνα με την ιδιότητα αυτή, αν υποθέσουμε ότι η καθυστέρηση της διεργασίας p (με ρυθμό λ) 'τελειώσει' πρώτη, τότε το γεγονός αυτό δεν έχει καμία επίδραση στην κατανομή της υπολειπόμενης καθυστέρησης της διεργασίας q – με άλλα λόγια ο υπολειπόμενος χρόνος ωσότου η αρχική καθυστέρηση της διεργασίας q τελειώσει συνεχίζει να κατανέμεται εκθετικά με ρυθμό μ. Το ίδιο ισχύει και για το αντίστροφο (δηλαδή αν 'τελειώσει' πρώτα η καθυστέρηση της διεργασίας q).

Βλέπουμε λοιπόν ότι με τις χρήση των εκθετικών κατανομών τα οποιαδήποτε προβλήματα λύνονται εύκολα με την βοήθεια της ιδιότητας της έλλειψης μνήμης.

Γενικές Κατανομές και Expansion Law

Στην περίπτωση όπου οι καθυστερήσεις των ενεργειών δεν ακολουθούν την εκθετική κατανομή, αλλά ακολουθούν γενικές κατανομές, τότε δημιουργείται πρόβλημα. Με βάση το παράδειγμα μας, ας υποθέσουμε ότι οι καθυστερήσεις των ενεργειών ακολουθούν τις γενικές κατανομές F και G. Τότε έχουμε:

$$F \to p \parallel G \to q \neq F \to (p \parallel G \to q) + G \to (F \to p \parallel q)$$

Ο λόγος για τον οποίο ισχύει αυτή η ανισότητα δεν είναι άλλος από την απουσία της ιδιότητας της έλλειψης μνήμης από τις γενικές κατανομές. Για παράδειγμα, μετά από την καθυστέρηση που προκαλείται από την F για τη διεργασία p, ο χρόνος που έχει παρέλθει προτού ολοκληρωθεί η $G \rightarrow q$ πρέπει να ληφθεί σοβαρά υπόψη αν θέλουμε να είμαστε σωστοί στον υπολογισμό της υπολειπόμενης καθυστέρησης έως ότου ξεκινήσει η διεργασία q.

Συμπέρασμα

Αρκετές προσπάθειες έχουν γίνει για να ενσωματωθούν τυχαίες μεταβλητές με γενικές κατανομές [13]. Ωστόσο η προσέγγιση αυτή 'υπέφερε' από το γεγονός πως οι γενικές κατανομές οδηγούσαν σε Στοχαστικές Άλγεβρες Διεργασιών με μη ελεγχόμενη δομή και αδύνατη ανάλυση.

Η χρησιμοποίηση των εκθετικών κατανομών έλυσε αρκετά προβλήματα. Τα μοντέλα με εκθετικές κατανομές αποτέλεσαν την βάση των σημερινών μεθοδολογιών για αξιολόγηση της επίδοσης των συστημάτων, αφού οι κατανομές αυτές επέτρεπαν την ακριβή περιγραφή αρκετών καταστάσεων πραγματικού χρόνου.

3. EPACSR

Σκοπός της μελέτης μας, είναι η παρουσίαση μιας άλγεβρας διεργασιών, που στόχο έχει να συνδυάσει όλα τα πλεονεκτήματα των πιο πάνω διεργασιών (ACSR, PACSR, MAC). Πιο συγκεκριμένα, το μοντέλο που θα παρουσιάσουμε, θα είναι μια μία διαδικασιακή άλγεβρα, η οποία θα αποτελεί προέκταση της PACSR – η οποία επεκτείνει την ACSR – με την εισαγωγή στοχαστικών διαδικασιών.

Στην PACSR υπάρχουν ενέργειες (actions) που χρειάζονταν κάποιο συγκεκριμένο χρόνο για να εκτελεστούν προτού το σύστημα προχωρήσει σε κάποια διεργασία, π.χ. $A : P$. Στην ACSR, μία ενέργεια εκτελείται για κάποιο διακριτό χρόνο u και ύστερα συνεχίζει το σύστημα με κάποια διεργασία, π.χ. $A^u : P$.

Η καινοτομία που εισάγουμε, έχει να κάνει με τον χρόνο που διαρκούν αυτές οι χρονικές ενέργειες (timed actions). Στο μοντέλο αυτό, κάθε χρονική ενέργεια καθυστερεί κατά κάποιο χρόνο t, ο οποίος ακολουθεί στοχαστική κατανομή με ρυθμό

λ. με άλλα λόγια, μια ενέργεια θα τερματίσει μεσα σε t μονάδες χρόνου με πιθανότητα ($\boldsymbol{1-e^{-\lambda t}}$).

3.1. Η σύνταξη του μοντέλου

Για την σύνταξη του μοντέλου, όπως είναι φυσικό, θα χρησιμοποιηθούν όροι και έννοιες και από τα τρία μοντέλα που προαναφέρθηκαν.

Ορίζουμε ως ***Proc*** το σύνολο όλων των διεργασιών και χρησιμοποιούμε τα ***P, Q, R*** και ***S*** για να δηλώνουμε διεργασίες. Μια διεργασία εξελίσσεται πραγματοποιώντας ενέργειες. Έτσι δηλώνουμε ως ***Act*** το σύνολο όλων των ενεργειών και χρησιμοποιούμε τους ελληνικούς χαρακτήρες ***α*** και ***β*** για να δηλώνουμε ενέργειες. Όπως και στις PACSR και ACSR, οι ενέργειες χωρίζονται σε δύο κατηγορίες:

- Σε ενέργειες που χρειάζονται κάποιο χρόνο για να εκτελεστούν και που έχουν πρόσβαση σ' ένα σύνολο από πόρους – ***timed actions***
- Σε στιγμιαίες ενέργειες ή στιγμιαία γεγονότα, που δεν καταναλίσκουν χρόνο για την εκτέλεση τους και μοντελοποιούν ενδο-διαδικασιακό συγχρονισμό (σχετίζονται με την επικοινωνία δύο διεργασιών) – ***instantaneous events***

Όπως και στην PACSR, ο κάθε πόρος χρησιμοποιείται με κάποια πιθανότητα, η οποία δίνει τον ρυθμό με τον οποίο ο πόρος δεν ανταποκρίνεται λόγω βλάβης. Επίσης γίνεται αναφορά και στον στοχαστικό χρόνο που διαρκεί κάθε χρονική ενέργεια. Στην άλγεβρα αυτή διακρίνουμε τρεις έννοιες:

Χρονικές Ενέργειες – Timed Actions. Θεωρούμε ότι το σύστημα αποτελείται από ένα πεπερασμένο σύνολο σειριακά επαναχρησιμοποιήσιμων πόρων, ***Res***. Χρησιμοποιούμε το ***r*** για να δηλώνουμε τους πόρους που είναι διαθέσιμοι και το $\overline{\boldsymbol{r}}$ για πόρους που δεν ανταποκρίνονται (failed resources). Μια ενέργεια A που χρησιμοποιεί πόρους, αντιπροσωπεύει την χρήση ενός υποσυνόλου του συνόλου ***Res***. Η ενέργεια αυτή χαρακτηρίζεται από:

- το σύνολο των πόρων που χρησιμοποιεί – $\rho(A)$
- την συνάρτηση $\pi_A(\mathbf{r}) : \mathbf{Res}$, όπου $\pi_A(\mathbf{r})$ η προτεραιότητα του πόρου r, κάθε φορά που εκτελείται η ενέργεια A, δεδομένου ότι $r \in \rho(A)$.

Οι ενέργειες, αφού καταναλίσκουν χρόνο, δηλώνονται ως A^{λ}, γεγονός που υποδηλώνει την εκτέλεση της ενέργειας A για χρόνο που ακολουθεί στοχαστική κατανομή με ρυθμό λ. Μία σημαντική παρατήρηση, είναι το γεγονός πως η στοχαστική κατανομή αναφέρεται για την ενέργεια A και όχι για τους πόρους που χρησιμοποιεί.

Ένα παράδειγμα μίας στοχαστικής ενέργειας A^{λ} είναι ένα σύνολο από ζεύγη $\{(r_1,p_1),(r_2,p_2),....,(r_n,p_n)\}^{\lambda}$, όπου ως r_i δηλώνεται ο πόρος (είτε είναι διαθέσιμος είτε όχι) και ως p_i ο αντίστοιχος βαθμός προτεραιότητας του πόρου για την συγκεκριμένη ενέργεια. Είναι σημαντικό να αναφέρουμε ότι στο πιο πάνω σύνολο θα εμφανίζονται μόνο πόροι που ανήκουν στην ενέργεια A και ο κάθε πόρος θα εμφανίζεται από μία φορά.

Στιγμιαία Γεγονότα – Instantaneous Events. Τα ‘γεγονότα’, όπως και στις PACSR και ACSR θα αποτελέσουν και εδώ το βασικό μηχανισμό για συγχρονισμό. Ουσιαστικά εδώ δεν υπάρχουν διαφορές με την PACSR και θα γίνει απλώς ία αναφορά.

Ένα γεγονός δηλώνεται από το ζεύγος (α, ρ) όπου α η ετικέτα του γεγονότος και ρ η προτεραιότητά του. Οι ετικέτες προέρχονται από το σύνολο $L = \mathcal{L} \cup \overline{\mathcal{L}} \cup \{\tau\}$ όπου α είναι η ετικέτα και $\overline{\alpha}$ η αλληλενέργεια. Η ειδική ετικέτα τ, ονομάζεται εσωτερική ενέργεια και αναφέρεται μόνο όταν δύο γεγονότα με ετικέτες α και $\overline{\alpha}$ εκτελούνται παράλληλα και υποδηλώνει συγχρονισμό.

Χρησιμοποιούμε τους όρους ***a, b, c*** για να δηλώνουμε γεγονότα. Επίσης, όταν δύο γεγονότα εκτελούνται παράλληλα και συγχρονισμένα, τότε η προτεραιότητα του τελικού γεγονότος (resulting composition event) είναι ίση με το άθροισμα των δύο αρχικών γεγονότων.

Πιθανότητες Πόρων – Resource Probabilities. Όπως και στην PACSR έτσι και σε αυτό το μοντέλο, ο κάθε πόρος σχετίζεται με μία πιθανότητα, που δηλώνει τον ρυθμό με τον οποίο ο πόρος είναι χρησιμοποιήσιμος. Συγκεκριμένα, για κάθε $r \in \mathrm{Res}$ δηλώνουμε ως $\rho(r) \in [0,1]$ την πιθανότητα ο πόρος να μπορεί να ανταποκριθεί και ως $\rho(\bar{r}) = 1 - \rho(r)$ την πιθανότητα ο πόρος να μην μπορεί να χρησιμοποιηθεί.

Για παράδειγμα, μία διεργασία $\{(cpu, 1)\}^{\lambda} : P$ με *ρ(cpu)=2/3,* σημαίνει ότι ο πόρος cpu με πιθανότητα 2/3 θα είναι διαθέσιμος και έτσι η διεργασία μπορεί με προτεραιότητα βαθμού 1 να την χρησιμοποιήσει για χρόνο που ακολουθεί εκθετική κατανομή με ρυθμό λ, για να εξελιχθεί στο τέλος σε P.

3.1.1. Διεργασίες

Η πιο κάτω γραμματική δηλώνει την σύνταξη του νέου μοντέλου μας:

$$\mathrm{P} ::= \mathbf{0} \mid \mathbf{A}^{\lambda} : \mathrm{P} \mid (\mathrm{e}, \mathrm{n}).\mathrm{P} \mid \mathrm{P} + \mathrm{P} \mid \mathrm{P} \parallel \mathrm{P} \mid \mathrm{P} \setminus \mathrm{F} \mid \mathrm{rec}\ \mathbf{X.P} \mid \mathbf{X}$$

Η διεργασία **0** δεν εκτελεί καμία ενέργεια. Ακολούθως, υπάρχουν δύο προκαθορισμένοι τελεστές, που αντιστοιχούν στα δύο είδη ενεργειών που έχουμε αναφέρει. Ο πρώτος, $\mathbf{A}^{\lambda} : \mathrm{P}$, εκτελεί μία στοχαστική ενέργεια που καταναλώνει πόρους A^{λ}, και διαρκεί για χρόνο με εκθετική κατανομή λ, και ακολούθως εξελίσσεται σε P. Ο δεύτερος, $(\mathrm{e}, \mathrm{n}).\mathrm{P}$, εκτελεί ένα στιγμιαίο γεγονός α και ακολούθως εξελίσσεται σε P. Και πάλι αναφέρουμε ότι τα στιγμιαία γεγονότα δεν καταναλίσκουν χρόνο για την εκτέλεση τους. Υπάρχουν όμως περιπτώσεις όπου ο διαχωρισμός μεταξύ των δύο τύπων ενεργειών, δεν θα είναι αναγκαίος, δηλαδή θα αναφερόμαστε και για τους δύο. Έτσι σε αυτές τις περιπτώσεις θα χρησιμοποιούμε τον ελληνικό χαρακτήρα '**α**' για να αναφερόμαστε γενικά για ενέργειες, π.χ. **α:P** όταν το α είναι στοχαστική ενέργεια και **α.P** όταν το α είναι στιγμιαίο γεγονός. Η διεργασία **P+Q** αντιπροσωπεύει μία μη-ντετερμινιστική επιλογή: οποιαδήποτε διεργασία από τις δύο μπορεί να επιλεγεί για να εκτελεστεί. Η διεργασία **P||Q** δηλώνει την παράλληλη εκτέλεση των δύο διεργασιών. Όπως θα δούμε και πιο κάτω, δύο διεργασίες μπορούν να εκτελεστούν παράλληλα και ανεξάρτητα η μία από την άλλη. Ο τελεστής περιορισμού **P \ F,** με $\mathbf{F} \not\subset \mathcal{L}$,, περιορίζει την συμπεριφορά της διεργασίας P. Εδώ, κανένα γεγονός με ετικέτα που να ανήκει στο F, δεν μπορεί να εκτελεστεί. Τέλος η διεργασία rec **X.P** δηλώνει αναδρομή.

3.2. Σημασιολογία του μοντέλου

Η σημασιολογία της QoSPA δίνεται μέσω δύο συστημάτων μεταβάσεων που ορίζουν την πιθανοτική και τη μη-ντετερμινιστική συμπεριφορά διεργασιών. Για να γίνει αυτό χρησιμοποιούμε ως αντικείμενα ερμηνείας σχηματισμούς ζεύγη της μορφής (P, W) που αντιπροσωπεύουν μία διεργασία P στον ‘κόσμο’ W, όπου το W συλλαμβάνει την κατάσταση στην οποία βρίσκονται οι πόροι του συστήματος (αν έχουν υποστεί πιθανοτική βλάβη ή όχι). Επομένως χωρίζουμε το σύνολο όλων των σχηματισμών σε δύο κλάσεις:

$$S_n = \{(P,W) \mid imr(P) \subseteq W\}, \text{ και}$$

$$S_p = \{(P,W) \mid imr(P) \not\subset W\},$$

όπου η συνάρτηση imr(P) επιστρέφει το σύνολο όλων των πόρων που απαιτούνται για χρήση από τη διεργασία P μέσα στην επόμενη χρονική στιγμή (ακριβής ορισμός μπορεί να δοθεί επαγωγικά στη δομή της P).

Σύμφωνα με αυτή την θεωρία, ο αρχικός σχηματισμός μιας διεργασίας P είναι ο (P, ∅), όπου δηλαδή ο κόσμος της διεργασίας είναι κενός. Για να υπολογίσουμε τον κόσμο της διεργασίας και τους διαθέσιμους πόρους προς αυτήν, πρέπει να εκτελεστεί μια πιθανοτική μετάβαση ως εξής:

$$\boxed{\dfrac{(P,W) \in S_p, \quad B = \{r \text{ or } \bar{r} \mid r \in imr(P)\}, \quad \pi = p(A)}{(P,W) \xrightarrow{\pi} (P,B)}}$$

Αφού γίνει αυτό, το κόσμος W δεν αλλάζει εκτός και αν πραγματοποιηθεί κάποια χρονική ενέργεια. Σε αυτό το νέο σχηματισμό οι διαθέσιμοι πόροι είναι γνωστοί και είναι δυνατόν να συνεχίσουμε με την εκτέλεση όλων των δυνατών ενεργειών. Ο κόσμος W δεν αλλάζει παρά μόνο αν πραγματοποιηθεί κάποια χρονική ενέργεια. Τότε, όπως θα δούμε και στους κανόνες που ακολουθούν, το σύστημα επαναφέρεται στον κενό κόσμο από όπου πιθανοτικές μεταβάσεις μπορούν και πρέπει να ξαναγίνουν για περαιτέρω εξέτασης του συστήματος.

Ένα παράδειγμα πιθανοτικών μεταβάσεων δίνεται από την πιο κάτω διεργασία. Έστω ότι έχουμε τους πόρους r_1 και r_2 και ισχύουν $p(r_1)$ = ½ και $p(r_2)$ = ¼. Τότε η διεργασία $P = \{(r_1,2),(\overline{r_2},3)\}:Q$ έχει τις εξής μεταβάσεις:

$$(P,0) \xrightarrow{1/8} (P,\{r_1,r_2\}) \qquad (P,0) \xrightarrow{1/8} (P,\{\overline{r_1},r_2\})$$

$$(P,0) \xrightarrow{3/8} (P,\{r_1,\overline{r_2}\}) \qquad (P,0) \xrightarrow{3/8} (P,\{\overline{r_1},\overline{r_2}\})$$

Στον Πίνακα 1 δίνουμε τους κανόνες που υπάρχουν για κάθε τελεστή της άλγεβρας. Πιο κάτω εξηγούμε τους κανόνες:

- Οι δύο κανόνες **ActI** και **ActT** για τους προκαθορισμένους τελεστές, θεωρούνται αξιώματα. Ο κανόνας **ActI** αναφέρεται για τα στιγμιαία γεγονότα και ο κανόνας **ActT** για τις στοχαστικές ενέργειες. Είναι φανερό πως στα στιγμιαία γεγονότα ο κόσμος W διατηρείται, ενώ, όπως αναφέραμε και πιο πάνω στις στοχαστικές ενέργειες μηδενίζεται.

  Έτσι, με βάση το παράδειγμα που δώσαμε πιο πάνω και τον κανόνα ActT ισχύει:

  $$(P,\{r_1,\overline{r_2}\}) \xrightarrow{\{(r_1,2),(\overline{r_2},3)\}^{\lambda}} (Q,0)$$

  ενώ οι διεργασίες:

  $$(P,\{r_1,r_2\}^{\lambda}),\ (P,\{\overline{r_1},r_2\}^{\lambda}),\ (P,\{\overline{r_1},\overline{r_2}\}^{\lambda})$$

  δεν έχουν μεταβάσεις

- Οι κανόνες **ChoiceL** και **ChoiceR**, είναι κανόνες μη-ντετερμινιστικής επιλογής. Είναι οι ίδιοι και για τα δύο ήδη ενεργειών, δηλαδή όπου **α** μπορεί να είναι A^{λ} για στοχαστικές ενέργειες ή **e** για στιγμιαία γεγονότα.

  Ως παράδειγμα, έστω η διεργασία $P = (e,2).P_1 + \{(r_1,5)\}^{1.3} : P_2$. Τότε η P μπορεί να επιλέξει να εκτελέσει το γεγονός (e,2) και να γίνει P_1 ή να εκτελέσει την στοχαστική ενέργεια $\{(r_1,5)\}^{1.3}$ και να γίνει P_2. (Για χάρη απλότητας δεν συμπεριλάβαμε τον κόσμο στον οποίο βρίσκεται η διεργασία).

- Για την παράλληλη εκτέλεση διεργασιών ορίζουμε τέσσερις κανόνες. Οι πρώτοι τρεις **ParIL, ParIR** και **ParCom** είναι παρόμοιοι με την PACSR και αναφέρονται στα στιγμιαία γεγονότα. Συγκεκριμένα, **ParIL** και **ParIR** δείχνουν ότι όταν δύο γεγονότα πρόκειται να εκτελεστούν παράλληλα, εκτελούνται ανεξάρτητα μεταξύ τους. Για το συγχρονισμό, υπάρχει ο κανόνας **ParCom**, με προϋπόθεση ότι αν η P εκτελεί ένα γεγονός e τότε η Q θα εκτελεί την αλληλενέργεια $\overline{e}$. Αυτό επιτρέπει διαδοχικά γεγονότα να εκτελούνται την ίδια χρονική στιγμή. Όπως φαίνεται και από τον κανόνα, ο συγχρονισμός αποτελεί μία εσωτερική ενέργεια **τ** με προτεραιότητα τον άθροισμα των επιμέρους προτεραιοτήτων.

  Ο τέταρτος κανόνας αναφέρεται στην παράλληλη εκτέλεση δύο στοχαστικών ενεργειών με διαφορετικούς ρυθμούς εκτέλεσης. Όπως βλέπουμε η διεργασία $A_1^{\lambda} : P$ μπορεί να εκτελεστεί και να ολοκληρωθεί ανεξάρτητα από την διεργασία $A_2^{\mu} : Q$ και το αντίθετο. Επίσης, η συνθήκη $\rho(A_1) \cap \rho(A_2) = 0$ δηλώνει ότι κάθε πόρος είναι σειριακά επαναχρησιμοποιήσιμος και ότι μόνο μία διεργασία μπορεί να χρησιμοποιήσει ένα συγκεκριμένο πόρο κάθε φορά.

  Ως παράδειγμα για τους πιο πάνω κανόνες, θεωρούμε τις πιο κάτω διεργασίες:

  $$\mathrm{St} = (\mathrm{in},3).\mathrm{A} + \{(\overline{\mathrm{out}},5)\}^{0.6} : \mathrm{B} \text{ και } \mathrm{Fs} = (\overline{\mathrm{in}},4).\mathrm{C} + \{(\mathrm{send},7)\}^{0.8} : \mathrm{D}$$

  Η διεργασία St || Fs μπορεί να έχει τις πιο κάτω μεταβάσεις:

  $\mathrm{St} \,||\, \mathrm{Fs} \xrightarrow{(\mathrm{in},3)} \mathrm{A} \,||\, \mathrm{Fs}$ από **ParIL**

  $\mathrm{St} \,||\, \mathrm{Fs} \xrightarrow{(\overline{\mathrm{in}},4)} \mathrm{St} \,||\, \mathrm{C}$ από **ParIR**

  $\mathrm{St} \,||\, \mathrm{Fs} \xrightarrow{(\tau,7)} \mathrm{A} \,||\, \mathrm{C}$ από **ParCom**

  $\mathrm{St} \,||\, \mathrm{Fs} \xrightarrow{(\overline{\mathrm{out}},\mathrm{send})^{0.6}} \mathrm{B} \,||\, \mathrm{Fs}$ από **ParT**

  $\mathrm{St} \,||\, \mathrm{Fs} \xrightarrow{(\overline{\mathrm{out}},\mathrm{send})^{0.8}} \mathrm{St} \,||\, \mathrm{D}$ από **ParT**

- Οι κανόνες για περιορισμό, **ResT** και **ResI** αναφέρονται σε περιπτώσεις όπου ένα υποσύνολο των στιγμιαίων γεγονότων δεν μπορεί να χρησιμοποιηθεί από το σύστημα.. Ο κανόνας **ResI** εκφράζει το γεγονός ότι μπορούν να χρησιμοποιηθούν γεγονότα που βρίσκονται εκτός του υποσυνόλου που είναι σε περιορισμό. Ο κανόνας **ResT** τονίζει το γεγονός ότι οι στοχαστικές ενέργειες παραμένουν ανεπηρέαστες από οποιοδήποτε είδος περιορισμού.

Έτσι, με βάση τις διεργασίες St και Fs, που αναφέραμε πιο πάνω, η $(\mathrm{St} \,||\, \mathrm{Fs}) \setminus \{\mathrm{in}\}$, έχει μόνο τις πιο κάτω μεταβάσεις:

$$(\mathrm{St} \,||\, \mathrm{Fs}) \setminus \{\mathrm{in}\} \xrightarrow{(\tau,7)} (\mathrm{A} \,||\, \mathrm{C}) \setminus \{\mathrm{in}\}$$

$$(\mathrm{St} \,||\, \mathrm{Fs}) \setminus \{\mathrm{in}\} \xrightarrow{(\overline{\mathrm{out}},\, \mathrm{send})^{0.6}} (\mathrm{B} \,||\, \mathrm{Fs}) \setminus \{\mathrm{in}\}$$

$$(\mathrm{St} \,||\, \mathrm{Fs}) \setminus \{\mathrm{in}\} \xrightarrow{(\overline{\mathrm{out}},\, \mathrm{send})^{0.8}} (\mathrm{St} \,||\, \mathrm{D}) \setminus \{\mathrm{in}\}$$

- Τέλος ο τελεστής **(recX.P, B)** δηλώνει αναδρομή. Η σημασιολογία του τελεστή αυτού δίνεται από τον κανόνα **Rec** και ο όρος P[recX.P / X] αντικαθιστά τον όρο «recX.P / X» κάθε φορά που πραγματοποιείται το X στη P.

ActI $(e.P,B) \xrightarrow{e} (P,B)$ **ActT** $\left(A^{\lambda} : P,B\right) \xrightarrow{A^{\lambda}} (P,0)$, if $\rho(A) \subseteq B$

ChoiceL $\dfrac{(P,B) \xrightarrow{\alpha} (P',B')}{(P+Q,B) \xrightarrow{\alpha} (P',B')}$ **ChoiceR** $\dfrac{(Q,B) \xrightarrow{\alpha} (Q',B')}{(P+Q,B) \xrightarrow{\alpha} (Q',B')}$

ParT $\dfrac{(P,B) \xrightarrow{A_1^{\lambda}} (P',B'),(Q,B) \xrightarrow{A_2^{\mu}} (Q',B')}{\begin{array}{c}(P \,||\, Q,B) \xrightarrow{(A_1|A_2)^{\lambda}} (P' \,||\, Q,B') \\ (P \,||\, Q,B) \xrightarrow{(A_1|A_2)^{\mu}} (P \,||\, Q',B')\end{array}}$ if $\rho(A_1) \cap \rho(A_2) = \varnothing$

$$\textbf{ParIL}\ \frac{(P,B)\xrightarrow{e}(P',B')}{(P\,||\,Q,B)\xrightarrow{e}(P',B')} \qquad \textbf{ParIR}\ \frac{(Q,B)\xrightarrow{e}(Q',B')}{(P\,||\,Q,B)\xrightarrow{e}(Q',B')}$$

$$\textbf{ParCom}\ \frac{(\mathrm{P},\mathrm{B})\xrightarrow{(\mathrm{a},\mathrm{p})}(\mathrm{P}',\mathrm{B}'),\ (\mathrm{Q},\mathrm{B})\xrightarrow{(\bar{\mathrm{a}},\mathrm{q})}(\mathrm{Q}',\mathrm{B}')}{(\mathrm{P}\,||\,\mathrm{Q},\mathrm{B})\xrightarrow{(\tau,\mathrm{p+q})}(\mathrm{P}'\,||\,\mathrm{Q}',\mathrm{B}')}$$

$$\textbf{ResT}\ \frac{(P,B)\xrightarrow{A^{\lambda}}(P',B')}{(P\setminus F,B)\xrightarrow{A^{\lambda}}(P'\setminus F,B)} \qquad \textbf{ResI}\ \frac{(P,B)\xrightarrow{(e,n)}(P',B')}{(P\setminus F,B)\xrightarrow{(e,n)}(P'\setminus F,B')}\ (\mathrm{e},\bar{\mathrm{e}}\notin F)$$

$$\textbf{Rec}\ \frac{(P[recX.P/X],B)\xrightarrow{\alpha}(P',B')}{(recX.P,B)\xrightarrow{\alpha}(P',B')}$$

Πίνακας 1: Σημασιολογία της EPACSR

Όπως φαίνεται από τη σημασιολογία της EPACSR διεργασίες γραμμένες στην άλγεβρα μπορούν να μεταφραστούν σε μαθηματικά αντικείμενα που αποτελούν συστήματα μεταβάσεων με πιθανοτικές, μη-ντετερμινιστικές και στοχαστικές μεταβάσεις. Στα επόμενα κεφάλαια θα περιγράψουμε μεθόδους ανάλυσης για το μοντέλο.

4. Ισχυρή Δυπροσομοίωση

Ένα βασικό κομμάτι για την ανάπτυξη της θεωρίας αλγεβρών διεργασιών έχει αφιερωθεί στη μελέτη κατάλληλων εννοιών ισοδυναμιών συμπεριφοράς. Τέτοιες ισοδυναμίες λαμβάνονται από διάφορες ερμηνείες του τι αποτελεί συμπεριφορά μιας διεργασίες, αφού δύο διεργασίες ορίζονται ως ισοδύναμες αν και μόνο αν παρουσιάζουν πανομοιότυπη συμπεριφορά. Υπάρχει μία πλειάδα τέτοιων ισοδυναμιών που διαχωρίζονται μεταξύ τους σύμφωνα με τη δύναμη και γνώση που χρειάζεται ένας παρατηρητής για να ξεχωρίσει δύο διεργασίες. Στα πλαίσια του προγράμματος εστιάσαμε την προσοχή μας σε μία τέτοια ισοδυναμία, την ισχυρή δυπροσομοίωση (strong bisimulation) μια από τις επικρατέστερες ισοδυναμίες που έχουν προταθεί στην βιβλιογραφία, και μάλιστα την πρώτη από αυτές [2, 23]. Δύο διεργασίες ορίζονται ως ισοδύναμες βάσει της σχέσης ισχυρής δυπροσομοίωσης, αν και μόνο αν η κάθε μια μπορεί να προσομοιώσει τη συμπεριφορά της άλλης βήμα προς βήμα. Η σχέση αυτή είναι πολύ σημαντική γιατί διατηρεί πληροφορίες σχετικά με τη δομή και την ύπαρξη αδιεξόδων ανάμεσα σε ισοδύναμες διεργασίες. Σχέσεις ισοδυναμίας μπορούν να χρησιμοποιηθούν για να αποδείξουν ότι ένα σύστημα ικανοποιεί τις προδιαγραφές του και ότι δύο συστήματα είναι ισοδύναμα. Τέλος, ένα σημείο πολύ χρήσιμο για τη αυτοματοποιημένη ανάλυση συστημάτων, είναι ότι σχέσεις ισοδυναμίας μπορούν να χρησιμοποιηθούν για την ελαχιστοποίηση ενός συστήματος μεταβάσεων, συνενώνοντας όλες τις ισοδύναμες καταστάσεις σε μία και μοναδική κατάσταση, εφαρμόζοντας με αυτό τον τρόπο κάποιου είδους συμπίεσης του συστήματος.

Σκοπός του κεφαλαίου αυτού είναι η μελέτη και ο ορισμός της ισχυρής δυπροσομοίωσης καθώς επίσης και η εύρεση ενός ικανού αλγορίθμου υπολογισμού της. Για τον σκοπό αυτό θα ορίσουμε στην επόμενη ενότητα το μοντέλο πάνω στο οποίο θα στηριχθούμε για την εξαγωγή των σχέσεων, στην συνέχεια θα ορίσουμε την ισχυρή δυπροσομοίωση και ακολούθως θα ασχοληθούμε με την διατύπωση του αλγορίθμου.

4.1. Το μοντέλο

Το μοντέλο μας δίνεται από την πιο κάτω πλειάδα:

$$S = < S_n, S_p, S_{st}, Act, \rightarrow_n, \rightarrow_p, \rightarrow_{st}, s_0 >$$

όπου

S_n : είναι το σύνολο των μη-ντετερμινιστικών καταστάσεων

S_p : είναι το σύνολο των πιθανοτικών καταστάσεων

S_{st} : είναι το σύνολο των στοχαστικών καταστάσεων

$\mathrm{Act} = \mathrm{Act1} \cup \mathrm{Act2}$: είναι το σύνολο των ετικετών των ενεργειών, όπου

- **Act1**: Τα στιγμιαία γεγονότα, που όπως μαρτυρεί και η ονομασία, εκτελούνται στιγμιαία χωρίς να καταναλίσκουν χρόνο
- **Act2**: Οι στοχαστικές ενέργειες, όπου για την εκτέλεση τους καταναλίσκεται χρόνος τ που ακολουθεί στοχαστική κατανομή με ρυθμό λ

$\rightarrow_{\mathrm{n}} \subset \mathrm{S_n} \times \mathrm{Act1} \times (\mathrm{S_n} \cup \mathrm{S_p} \cup \mathrm{S_{st}})$ αντιπροσωπεύει τις μη-ντετερμινιστικές μεταβάσεις, όπου ο όρος Act1 αναφέρεται στα στιγμιαία γεγονότα

$\rightarrow_{p} \subset S_p \times (0,1] \times (S_n \cup S_{st})$ αντιπροσωπεύει τις πιθανοτικές μεταβάσεις

$\rightarrow_{\mathrm{st}} \subset \mathrm{S_{st}} \times \mathrm{Act2} \times (\mathrm{S_n} \cup \mathrm{S_{st}} \cup \mathrm{S_p})$ αντιπροσωπεύει τις στοχαστικές μεταβάσεις, όπου ο όρος Act2 αναφέρεται στις στοχαστικές ενέργειες

s_0 **:** είναι η αρχική κατάσταση

Στην συνέχεια θα γράφουμε και θα χρησιμοποιούμε τα εξής:

- S για το σύνολο $S_n \cup S_p \cup S_{st}$ και θα αντιπροσωπεύεται από τα P και $\mathrm{P'}$
- α και b για το σύνολο Act

4.2. Ισχυρή δυπροσομοίωση

Προτού δώσουμε τον ορισμό της ισχυρής δυπροσομοίωση για το πιο πάνω μοντέλο θα ορίσουμε τρεις νέες συναρτήσεις, δεδομένου ότι $\mathrm{P}, \mathrm{P'} \in \mathrm{S}, \mathrm{C} \subseteq \mathrm{S}$:

1. $$pr(P,P') = \begin{cases} \pi, & \text{if } \mathrm{P} \xrightarrow{\pi}_p P' \\ 1, & \text{if } \mathrm{P} = \mathrm{P'}, \mathrm{P'} \in (\mathrm{S_n} \cup S_{st}) \\ 0, & \text{otherwise} \end{cases}$$

2. $$\mu(P,C) = \sum\nolimits_{P' \in C} pr(P,P')$$

3. $$\gamma(P,\alpha,C) = \sum_i \{\lambda \mid P \xrightarrow{\alpha^{\lambda}}_i P', P' \in C\}$$

Μία σχέση ισοδυναμίας $R \subseteq S \times S$ είναι ισχυρή δυπροσομοίωση αν, οποτεδήποτε ισχύει $(P, Q) \in R$ τότε:

1. για κάθε $\alpha \in Act$, αν $(P,Q) \in S_n$ και $P \xrightarrow{\alpha}_n P'$ τότε $Q \xrightarrow{\alpha}_n Q'$ και $(P', Q') \in R$,

2. για κάθε $C \in S/R$, μ(P, C) = μ(Q, C),

3. για κάθε $C \in S/R$, γ(P, α, C) = γ(Q, α, C).

Δύο καταστάσεις (διεργασίες) είναι ισχυρά ισοδύναμες (strong bisimulation equivalent), P ~ Q, αν υπάρχει ισχυρή δυπροσομοίωση R τέτοια ώστε να ισχύει $(P, Q) \in R$.

Προφανώς ο ορισμός αυτός απαιτεί δύο ισοδύναμες διεργασίες να προσομοιώνουν η μία η άλλη τόσο όσο αφορά την μη-ντετερμινιστική συμπεριφορά (clause 1), όσο και ως προς τη συνολική πιθανότητα και συνολικό στοχαστικό rate μετακίνησης σε άλλες ισοδύναμες καταστάσεις (clauses 2 και 3).

4.3. Ο Αλγόριθμος

Στην ενότητα αυτή θα δώσουμε ένα αλγόριθμο για την ισχυρή δυπροσομοίωση. Για τον σκοπό αυτό η ενότητα θα χωριστεί σε τρεις υποενότητες. Αρχικά θα ορίσουμε ξεχωριστά διαδικασίες για μία τις μη-ντετερμινιστικές, στοχαστικές και πιθανοτικές καταστάσεις τις οποίες τελικά θα συνδυάσουμε για να πάρουμε ολοκληρωμένο τον αλγόριθμο. Ξεκινούμε με τη γενική περιγραφή του αλγόριθμου.

Οι περισσότεροι αλγόριθμοι που σχετίζονται με την δυπροσομοίωση εφαρμόζονται σε συστήματα με διακριτό αριθμό καταστάσεων. Η μέθοδος που θα χρησιμοποιήσουμε βασίζεται στην τεχνική του διαχωρισμού (partition technique) [24, 25, 33]. Αυτό συνεπάγεται πως ξεκινώντας από ένα αρχικό διαχωρισμό των καταστάσεων, ο οποίος αποτελείται από μία μοναδική κλάση που περιέχει όλες τις καταστάσεις, οι κλάσεις εκλεπτύνονται (refined) έως ότου ο τελικός διαχωρισμός αντιστοιχίσει σε ισοδυναμία δυπροσομοίωσης. Το τελικό αποτέλεσμα, δηλ οι κλάσεις που προκύπτουν, αποτελεί το μεγαλύτερο δυνατό σύνολο δυπροσομοίωσης αφού περιέχει ένα ελάχιστο αριθμό κλάσεων ισοδυναμίας.

Για την τεχνική εκλέπτυνσης ενός διαχωρισμού (partition), η χρησιμοποίηση ενός νέου όρου, του σχίσματος (splitter), παίζει καθοριστικό ρόλο (όπως θα δούμε στην συνέχεια, θα ορίσουμε τρία είδη σχισμάτων, ανάλογα με το αν η κατάσταση που εξετάζουμε είναι μη-ντετερμινιστική, στοχαστική ή πιθανοτική). Το **μη-ντετερμινιστικό σχίσμα** ορίζεται ως ένα ζεύγος (α, $C_{σχ}$) το οποίο αποτελείται από μία ενέργεια (action) α και την κλάση $C_{σχ}$. Κατά την διάρκεια της εκλέπτυνσης η κλάση C χωρίζεται με βάση το σχίσμα στις υποκλάσεις C_1 και C_2, έτσι ώστε:

- η υποκλάση C_1 να περιέχει όλες εκείνες τις καταστάσεις από την κλάση C οι οποίες: μπορούν να πραγματοποιήσουν μεταβάσεις με ενέργεια α (α-transitions), και να οδηγηθούν σε καταστάσεις που ανήκουν στην κλάση $C_{σχ}$ και είναι πιθανοτικές καταστάσεις όπου με πιθανότητα 'ένα' (p = 1) πάνε σε κατάσταση που μπορεί να πραγματοποιήσει μετάβαση με ενέργεια α.
- η υποκλάση C_2 να περιέχει τις υπόλοιπες καταστάσεις

Στη συνέχεια θα δώσουμε τον αλγόριθμο για την ισχυρή δυπροσομοίωση. Όπως θα δούμε, ο αλγόριθμος χρησιμοποιεί ένα δυναμικό σύνολο από σχίσματα, το οποίο ονομάζεται **Splitters** και το σύνολο των κλάσεων που ονομάζεται **Partition**. Η πολυπλοκότητα του αλγορίθμου είναι της τάξης ***O(m log n)***, όπου n ο αριθμός των καταστάσεων και m ο αριθμός των μεταβάσεων.

1.Αρχικοποίηση

Partition := {S}

/ αρχικός διαχωρισμός αποτελείται από μία κλάση που περιέχει όλες τις καταστάσεις /

Splitters := Act x Partition

/ όλα τα πιθανά ζεύγη ενεργειών και κλάσεων πρέπει να θεωρηθούν ως σχίσματα /

2.Κύριος Βρόγχος

While *(Splitters $\neq$ 0)*

choose *splitter (a,* $C_{σχ}$) $\in$ *Splitters*

forall $C \in$ *Partition* *split (C, a,* $C_{\sigma\chi}$, Partition, Splitters)

/ όλες οι κλάσεις (περιλαμβανόμενης και της $C_{\sigma\chi}$) διαχωρίζονται /

Splitters: = Splitters - (a, $C_{\sigma\chi}$)

/ το σχίσμα που χρησιμοποιήθηκε αφαιρείται από το σύνολο των Splitters /

4.3.1. Διαδικασία για καταστάσεις με μη-ντετερμινιστική συμπεριφορά.

Απομένει τώρα να καθορίσουμε την συνάρτηση ***split***. Σκοπός της είναι να διαχωρίσει μία κλάση C, χρησιμοποιώντας το ζεύγος *(a,* $C_{\sigma\chi}$) ως μη-ντετερμινιστικό σχίσμα. Στην περίπτωση που αυτό πραγματοποιηθεί, τότε η κλάση C διαχωρίζεται στις υποκλάσεις C_1 και C_2.

procedure *split (C, a,* $C_{\sigma\chi}$, Partition, Splitters)

$$C_1 := \{P \mid P \in C \wedge \exists Q : (P \xrightarrow{\alpha} Q \wedge Q \in C_{\sigma\chi}) \mid P \in C \cap Sp \wedge \sum\{\pi \mid \mathrm{P} \xrightarrow{\pi} \mathrm{P}' \wedge \mathrm{P}' \xrightarrow{\alpha}\} = 1\}$$

/* υπολογισμός υποκλάσης C_1 */

/*στην υποκλάση αυτή μπαίνουν και οι πιθανοτικές καταστάσεις όπου με πιθανότητα 'ένα' πάνε σε καταστάσεις που μπορούν να πραγματοποιήσουν μεταβάσεις με ενέργεια α*/

if $(C_1 \neq C \wedge C_1 \neq 0)$

/ ο αλγόριθμος προχωράει μόνο όταν η κλάση C χρειάζεται να διαχωριστεί /

$C_2 = C - C_1$

/ η C_2 είναι συμπληρωματική της C_1 /

Partition: = Partition $\cup \{C_1, C_2\} - \{C\}$

Splitters: = Splitters $\cup (\mathrm{Act}\, x \{C_1, C_2\}) - \mathrm{Act}\, x \{C\}$

/ενημερώνονται τα σύνολα των Partition και Splitters/

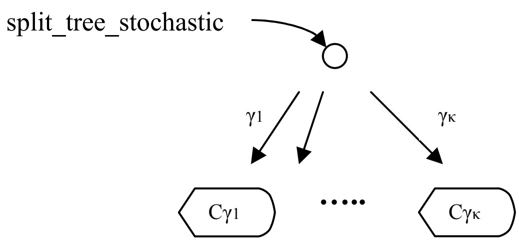
Όπως φαίνεται και από τον αλγόριθμο, η όλη διαδικασία τερματίζει έως ότου το σύνολο των Splitters αδειάσει. Για να φτάσουμε έως εδώ όμως, ο αλγόριθμος, με διαδοχικές επαναλήψεις, ελέγχει όλες τις κλάσεις με σκοπό την εύρεση – αν υπάρχει – του κατάλληλου σχίσματος, ούτως ώστε να προχωρήσει στην διαδικασία της διάσπασης. Θα εξηγήσουμε τον πιο πάνω αλγόριθμο διαμέσου δύο παραδειγμάτων.

4.3.2. Διαδικασία για καταστάσεις με στοχαστική και πιθανοτική συμπεριφορά

Στην ενότητα αυτή θα λάβουμε υπόψη την $3^{η}$ προδιαγραφή που δώσαμε για τον ορισμό της ισχυρής δυπροσομοίωσης, πως για όλες τις ισοδύναμες κλάσεις C της σχέσης ισοδυναμίας και για όλες τις ενέργειες α, πρέπει να ισχύει η σχέση:

$$\gamma(P , \alpha , C) = \gamma(Q , \alpha , C)$$

Ο βασικός αλγόριθμος είναι ο ίδιος με αυτόν που δώσαμε στην προηγούμενη ενότητα. Το μόνο που χρειάζεται είναι η τροποποίηση της συνάρτηση split. Για τον σκοπό αυτό, η νέα συνάρτηση split' χρησιμοποιεί μία δομή-δεδομένων, την split_tree_stochastic, η οποία φαίνεται στο πιο κάτω σχήμα:



Σκοπός της split' είναι να χωρίσει τις καταστάσεις με βάση την γ-τιμή τους. Κατά την διάρκεια της εκλέπτυνσης, όταν η κλάση C διαχωρίζεται από κάποιο **στοχαστικό σχήμα** *(a,* $C_{\sigma\chi}$), όπου τώρα $a \in A^{\lambda}$, πιθανός να δημιουργούνται περισσότερες των δύο υποκλάσεων $C\gamma_1, C\gamma_2, \ldots, C\gamma_\kappa$. Έτσι η κλάση C διαχωρίζεται με τέτοιο τρόπο ώστε το cumulative rate $\gamma(P,\alpha,C_{\sigma\chi}) = \gamma_j$ να είναι το ίδιο για όλες τις καταστάσεις P που ανήκουν στην ίδια υποκλάση $C\gamma_j$. Η $C\gamma_j$ αποτελεί ένα φύλλο (leaf) στην δομή split_tree_stochastic, όπως φαίνεται και στο σχ5. Να αναφέρουμε ότι, όπως και στις μη-ντετερμινιστικές καταστάσεις, έτσι και εδώ ελέγχεται η περίπτωση όπου μία πιθανοτική κατάσταση με πιθανότητα 'ένα' (p=1) πραγματοποιεί μετάβαση σε στοχαστική κατάσταση, γεγονός που διαμορφώνει καταλλήλως την συνάρτηση split’. Η νέα συνάρτηση split’ έχει ως ακολούθως:

procedure *split’ (C, a,* $C_{\sigma\chi}$, Partition, Splitters)

forall $P \in C$

$\gamma := \gamma(P, \alpha, C_{\sigma\chi})$

/* υπολογίζεταί το cumulative stochastic rate από την κατάσταση P στην $C_{\sigma\chi}$ */

if (γ = 0) **and** $\{P \mid P \in C \cap Sp \wedge \sum\{\pi \mid P \xrightarrow{\pi} P' \wedge P' \xrightarrow{\alpha}\} = 1\}$

$\gamma := \gamma(P', \alpha, C_{\sigma\chi})$

/*οι πιθανοτικές καταστάσεις P όπου με πιθανότητα ‘ένα’ πάνε σε καταστάσεις P΄ που μπορούν να πραγματοποιήσουν μεταβάσεις με ενέργεια α, έχουν το ίδιο γ με τις καταστάσεις P’, έτσι ώστε να μην διαχωριστούν */

insert (split_tree_stochastic, P , γ)

/ η κατάσταση P εισάγεται στην δομή δεδομένων */*

/ τώρα η δομή περιέχει k-φύλλα* $C\gamma_1, C\gamma_2, \ldots, C\gamma_\kappa$ */

if (k > 1)

/* το πρόγραμμα προχωράει αν η C μπορεί να διασπαστεί σε k > 1 υποκλάσεις */

Partition: = Partition $\cup \{C\gamma_1, C\gamma_2, \ldots, C\gamma_k\} - \{C\}$

Splitters: = Splitters $\cup (Act \times \{C\gamma_1, C\gamma_2, \ldots, C\gamma_k\}) - Act \times \{C\}$

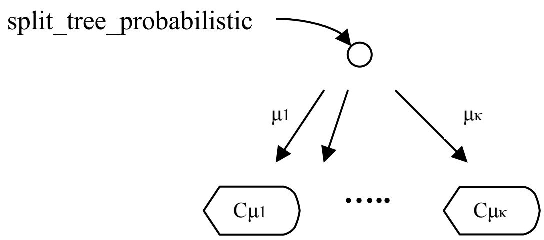
/ενημερώνονται τα σύνολα των Partition και Splitters/

Η χρονική πολυπλοκότητα του αλγορίθμου είναι της τάξης του O(**m**log**n**) ενώ η χωρική πολυπλοκότητα είναι της τάξης του O(**m** + **n**), όπου n ο αριθμός των καταστάσεων και m ο αριθμός των μεταβάσεων.

Για τις πιθανοτικές καταστάσεις η συνάρτηση split’ διαφοροποιείται, με αποτέλεσμα να έχουμε την split’’. Ως αρχική διαφορά, είναι το γεγονός πως για τις πιθανοτικές καταστάσεις λαμβάνεται υπόψη η 2η προδιαγραφή που δώσαμε για τον ορισμό της ισχυρής δυπροσομοίωσης, ότι δηλ για όλες τις ισοδύναμες κλάσεις C της σχέσης ισοδυναμίας R ισχύει:

$$\mu(P, C) = \mu(Q, C)$$

Έτσι η συνάρτηση split’’ διαφέρει σε σχέση με την split’ όσον αφορά την δομή δεδομένων που χρησιμοποιεί, η οποία ονομάζεται split_tree_probabilistic, όπως φαίνεται και στο σχήμα:



Τέλος, μία άλλη διαφορά εντοπίζεται στην επιλογή του σχίσματος. Τώρα χρησιμοποιούμε **πιθανοτικά σχίσματα** της μορφής (- , $C_{\sigma\chi}$), τα οποία δηλώνουν πως οι καταστάσεις μίας κλάσης θα πραγματοποιούν μετάβαση σε καταστάσεις της κλάσης $C_{\sigma\chi}$ με κάποιο άλλο τρόπο – πιθανοτικό – και όχι με κάποια ενέργεια, γι αυτό και ο συμβολισμός {-}. Εδώ αξίζει να αναφέρουμε ότι μία κατάσταση μη-πιθανοτική, έχει σύμφωνα με τον ορισμό μ=1. Έτσι, μία πιθανοτική κατάσταση P, η οποία με πιθανότητα ‘ένα’ πάει σε κατάσταση $P' \in C_{\sigma\chi}$, θα έχει μ=1 με αποτέλεσμα οι P και P΄ να μπουν στην ίδια υποκλάση χωρίς να διαχωριστούν.

Πιο κάτω δίνουμε τον αλγόριθμο της συνάρτησης split’’, καθώς και ένα παράδειγμα για καλύτερη κατανόηση.

procedure *split’’ (C,* $C_{\sigma\chi}$, Partition, Splitters’)

forall $P \in C$

$\mu := \mu(P, C_{\sigma\chi})$

/* υπολογίζεταί το cumulative probabilistic rate από την κατάσταση P στην $C_{\sigma\chi}$ */

insert (split_tree_probabilistic, P , μ)

/ η κατάσταση P εισάγεται στην δομή δεδομένων */*

/ τώρα η δομή περιέχει k-φύλλα* $C\mu_1, C\mu_2, \ldots, C\mu_\kappa$ */

if (k > 1)

/* το πρόγραμμα προχωράει αν η C μπορεί να διασπαστεί σε k > 1 υποκλάσεις */

Partition: = Partition $\cup \{C\mu_1, C\mu_2, \ldots, C\mu_k\} - \{C\}$

Splitters: = Splitters $\cup (- x \{C\mu_1, C\mu_2, \ldots, C\mu_k\}) - (- x \{C\})$

/ενημερώνονται τα σύνολα των Partition και Splitters/

4.3.3. Ο ολοκληρωμένος αλγόριθμος

Συνδυάζοντας τα στοιχεία των δύο προηγούμενων ενοτήτων θα δώσουμε ολοκληρωμένο τον αλγόριθμο:

1.Αρχικοποίηση

Partition: = {S}

/*αρχικός διαχωρισμός αποτελείται από μία κλάση που περιέχει όλες τις καταστάσεις */

Splitters: = (Act x Partition) ∪ ({-} x Partition)

/*όλα τα πιθανά πρέπει να θεωρηθούν ως σχίσματα */

2.Κύριος Βρόγχος

While *(Splitters ≠ 0)*

choose *splitter (a,* $C_{\sigma\chi}$ *)* $\in$ *Splitters*

/ το splitter μπορεί να είναι μη-ντετερμινιστικό, στοχαστικό ή πιθανοτικό (α = {-}) */*

case $\alpha \in$ Act1

forall *C* $\in$ *Partition* *split (C, a,* $C_{\sigma\chi}$, Partition, Splitters)

Splitters: = Splitters - (a, $C_{\sigma\chi}$)

case $\alpha \in$ Act2

forall *C* $\in$ *Partition* *split' (C, a,* $C_{\sigma\chi}$, Partition, Splitters)

Splitters: = Splitters - (a, $C_{\sigma\chi}$)

case $\alpha = \{-\}$

forall *C* $\in$ *Partition* *split'' (C,* $C_{\sigma\chi}$, Partition, Splitters)

Splitters: = Splitters - ({-} , $C_{\sigma\chi}$)

Ισχυρή δυπροσομοίωση και συστήματα αξιωμάτων

Σε σχέση με την ισχυρή δυπροσομοίωση, έχουμε μελετήσει συστήματα αξιωμάτων για ένα υποσύνολο της EPACSR την EACSR. Εκτενής αναφορά υπάρχει στο Παράρτημα του παραδοτέου ([26]).

5. Αλυσίδες Markov και Μοντελοέλεγχος

Η αξιολόγηση της απόδοσης είναι η τεχνική η οποία προσπαθεί να δώσει απαντήσεις σε ερωτήματα που αφορούν την απόδοση και την αξιοπιστία ενός συστήματος. Χαρακτηριστικά παραδείγματα τέτοιων ερωτημάτων είναι:

- Ποια είναι η συχνότητα που παρουσιάζει ένα σύστημα ασταθή συμπεριφορά;
- Πόσους πελάτες μπορεί να εξυπηρετήσει ένας server;
- Είναι η σωστή και έγκαιρη παράδοση των πακέτων επιτυχημένη σε βαθμό μεγαλύτερο του 90% του συνόλου;

Οι τεχνικές που αναπτύχθηκαν για να δώσουν απαντήσεις σε τέτοιας φύσεως ερωτήματα ανήκουν σε δύο μεγάλες κατηγορίες:

1. Στις τεχνικές που στηρίζονται στις μετρήσεις (measurement-based techniques)

2. Στις τεχνικές που βασίζονται στην μοντελοποίηση (model-based techniques)

Παρόλο που οι τεχνικές που στηρίζονται στις μετρήσεις είναι πολύ σημαντικές και ακριβείς, αφού πραγματοποιούνται στο πραγματικό σύστημα ή τουλάχιστον σε κάποιο πρωτότυπο, δεν υποστηρίζονται αρκετά λόγω του μεγάλου κόστους και του γεγονότος ότι σε αρκετές περιπτώσεις κρίνονται ακατάλληλες. Γι αυτό τον λόγο περιοριζόμαστε στην μοντελοποίηση των συστημάτων. Η μοντελοποίηση συστημάτων συνήθως συνεπάγεται τη διαδικασία της απόσπασης (abstraction), δηλαδή, της απλοποίησης της περιγραφής ενός συστήματος διατηρώντας μόνο ένα περιορισμένο αριθμό από τις αρχικές λεπτομέρειες, οι οποίες όμως είναι ικανές στο να μας βοηθήσουν στις μετρήσεις μας για την αξιολόγηση της απόδοσης [2].

Οι αλυσίδες Μαρκόβ συνεχούς χρόνου, CTMCs, αποτελούν ένα πάρα πολύ γνωστό μοντέλο το οποίο είναι αρκετά χρήσιμο για δύο λόγους: Πρώτον, γιατί το μοντέλο αυτό είναι αρκετά γενικό για τη μοντελοποίηση αρκετών φυσικών συστημάτων και δεύτερον, γιατί η θεωρία των αλυσίδων Μαρκόβ παρέχει αριθμητικές μεθόδους για την αξιολόγηση των μέτρων επίδοσης αυτού του μοντέλου. Μπορούν να θεωρηθούν ως ένα σύστημα μεταβάσεων με ετικέτες, όπου οι ετικέτες – ρυθμοί των εκθετικών κατανομών – απεικονίζουν την ταχύτητα με την οποία το σύστημα μετακομίζει από μία κατάσταση σε άλλη [27,28]. Οι τυπικές μετρήσεις που αφορούν την απόδοση σε CTMCs βασίζονται στις:

- Steady-state probabilities
- Transient-state probabilities

Οι steady-state probabilities αφορούν την συμπεριφορά του συστήματος μακροπρόθεσμα ή αλλιώς όταν το σύστημα φτάσει σε ισορροπία. Οι Transient-state probabilities απλώς θεωρούν ότι το σύστημα βρίσκεται σε κάποια καθορισμένη χρονική στιγμή t. Σύγχρονοι αριθμητικοί αλγόριθμοι μας επιτρέπουν τον υπολογισμό και των δύο ειδών πιθανοτήτων με γρήγορο και εύκολο τρόπο [29].

Αφού λοιπόν δημιουργηθεί το μοντέλο CTMC, ως επόμενο βήμα ορίζουμε την αξιολόγηση των μετρήσεων/ζητημάτων που μας αφορούν με την απαιτούμενη ακρίβεια. Για τον σκοπό αυτό θα χρησιμοποιήσουμε την χρονική λογική CSL για τον καθορισμό των διαφόρων τύπων περιορισμού (constraints) και κριτηρίων ορθότητας, αλλά και για να δείξουμε πως οι τεχνικές του μοντελοέλεγχου μπορούν να εφαρμοστούν πλήρως για την αυτόματη ανάλυση αυτών των περιορισμών σε μοντέλα CTMCs [30, 31, 32].

Στο κεφάλαιο αυτό θα επικεντρώσουμε το ενδιαφέρον μας στην χρήση των Μαρκοβιανών μοντέλων για την αξιολόγηση της απόδοσης και της αξιοπιστίας των συστημάτων (κυρίως υπολογιστικών και επικοινωνιακών). Στις επόμενες ενότητες θα κάνουμε μια πιο αναλυτική αναφορά στις Αλυσίδες Μαρκόβ. Συγκεκριμένα θα αναφερθούμε στις Διακριτού Χρόνου Αλυσίδες Μαρκόβ (DTCM – Discrete Time Markov Chains) και ακολούθως στις CTMCs, δίνοντας τους τύπους για τους υπολογισμούς των steady-state και transient-steady πιθανοτήτων στα μοντέλα. Στη συνέχεια θα επεκτείνουμε τις τεχνικές για το μοντέλο της EPACSR το οποίο το οποίο βρίσκεται ανάμεσα στα CTMC (continuous time Markov chains) και τα DTMC (discrete time Markov chains), αφού περιέχει τόσο πιθανοτικές όσο και στοχαστικές ακμές. Τέλος στην τελευταία ενότητα όπου θα περιγράψουμε την χρονική λογική CSL και τους αλγορίθμους μοντελοέλεγχου.

5.1. Αλυσίδες Μαρκόβ Διακριτού Χρόνου

Οι DTMCs αποτελούν μία κλάση των πιθανοτικών μοντέλων. Στις αλυσίδες αυτές ο χρόνος ‘κυλάει’ με διακριτά βήματα, έτσι ώστε οι μεταγενέστερες χρονικές στιγμές να είναι απαριθμήσιμες (denumerable) και να μπορούν να οριστούν στο σύνολο IN. Έτσι οι DTMCs μπορούν να οριστούν ως $\{X_n, n \in IN\}$, όπου X_n τυχαία μεταβλητή και **n** ο διακριτός χρόνος. Το σύνολο των τιμών των τυχαίων μεταβλητών X_n ορίζεται ως ο σταθερός χώρος I των DTMCs και είναι και αυτός απαριθμήσιμος και πεπερασμένος.

Το κύριο χαρακτηριστικό όμως των DTMCs είναι το γεγονός ότι στις αλυσίδες αυτές εφαρμόζεται η ιδιότητα Μαρκόβ (Markov property). Αυτό σημαίνει πως δοθέντος της παρούσας κατάστασης σε μία DTMC, η μεταγενέστερη εξέλιξη της εξαρτάται αποκλειστικά και μόνο απ' αυτή και καθόλου από τις προηγούμενες. Με άλλα λόγια:

$$\Pr\{X_{n+1} = i_{n+1} \mid X_0 = i_0, ..., X_n = i_n\} = \Pr\{X_{n+1} = i_{n+1} \mid X_n = i_n\} \qquad (1)$$

όπου $X_n = i_n$ δηλώνει ότι το σύστημα την χρονική στιγμή **n** βρίσκεται στην κατάσταση $i_n \in I$. Από την (1) βλέπουμε ότι το μέλλον (n+1) εξαρτάται από την παρούσα κατάσταση (n) και όχι από τις καταστάσεις του παρελθόντος.

Αξίζει επίσης να τονίσουμε ότι οι DTMCs παρουσιάζουν ομοιογένεια στον χρόνο (time homogeneous), πράγμα που σημαίνει ότι αυτό που πραγματικά ενδιαφέρει είναι οι σχετικές διαφορές των διαφορετικών χρονικών στιγμών, δηλαδή ισχύει:

$$\Pr\{X_{n+1} = i \mid X_n = j\} = \Pr\{X_{m+1} = i \mid X_m = j\} \text{ για } n, m \in IN \quad (2)$$

Ορίζουμε λοιπόν την υπό συνθήκη πιθανότητα $p_{j,k}(m,n) = \Pr\{X_n = k \mid X_m = j\}$ για όλα τα *m = 0,1,…,n* η οποία δηλώνει την πιθανότητα να γίνει η μετάβαση από την κατάσταση $j \in I$ την χρονική στιγμή $m \in IN$ στην κατάσταση $k \in I$ την χρονική στιγμή $n \in IN$. Σύμφωνα όμως με την (2), αυτές οι πιθανότητες μετάβασης εξαρτώνται μόνο από την χρονική διαφορά ***l = n-m***. Έτσι ορίζουμε ως $p_{j,k}(l) = \Pr\{X_{m+l} = k \mid X_m = j\}$ την *'πιθανότητα μετάβασης l-βημάτων' (l-step transition probability)*. Η πιθανότητα μετάβασης 1-βήματος γράφεται απλά ως $p_{j,k}$, ενώ η αρχική κατανομή των DTMCs καθορίζεται από το διάνυσμα $\underline{\pi}(0) = (\pi_0(0), ..., \pi_{|I|}(0))$

Με βάση λοιπόν τα πιο πάνω, οι DTMCs ορίζονται πλήρως από τις αρχικές πιθανότητες και τις πιθανότητες μετάβασης 1-βήματος, οι οποίες συνοψίζονται στον πίνακα $P = (p_{i,j})$. Ο πίνακας P ονομάζεται *'πίνακας καταστάσεων πιθανοτικών μεταβάσεων' (state-transition probability matrix)* και ισχύει $0 \le p_{i,j} \le 1$ και $\sum_j p_{i,j} = 1$.

5.2. Συνεχής Χρόνου Αλυσίδες Μαρκόβ

Ως CTMCs ορίζουμε ένα πεπερασμένο σύνολο καταστάσεων όπου οι μεταβάσεις δηλώνονται με ετικέτες. Οι ετικέτες αυτές, $q_{i,j}$, υποδηλώνουν το ρυθμό της εκθετικής κατανομής που αντιπροσωπεύει τον χρόνο παραμονής (ή καθυστέρησης) στην παρούσα κατάσταση i, προτού το σύστημα πραγματοποιήσει μετάβαση στην κατάσταση j.

Για κάθε κατάσταση i, ορίζουμε την μεταβλητή μ_i, η οποία δηλώνει τον ρυθμό της εκθετικής κατανομής που αντιστοιχεί στον χρόνο παραμονής στην κατάσταση i και ισούται με $\mu_i = \sum_{j \neq i} q_{i,j}$. Επίσης ορίζουμε ως $p_{i,j} = \frac{q_{i,j}}{\sum_{k \neq i} q_{i,k}}$ την πιθανότητα, μετά από κάθε καθυστέρηση στην κατάσταση i, να πραγματοποιηθεί η μετάβαση στην κατάσταση j.

Με βάση λοιπόν τα πιο πάνω, ορίζουμε τον πίνακα **Q** με στοιχεία $q_{i,j} = \mu_i p_{i,j}$ για $i \neq j$ και $q_{i,i} = -\sum_{j \neq i} q_{i,j} = -\mu_i$. Όταν λοιπόν ένα σύστημα βρεθεί στην κατάσταση i, για εκείνες τις καταστάσεις j που μπορεί να πραγματοποιηθεί μετάβαση από την i, δηλαδή για $q_{i,j} > 0$, επιλέγεται αντίστοιχα μία τυχαία μεταβλητή, σύμφωνα με την κατανομή $F_{i \to j}(t) = 1 - e^{-q_{i,j}t}$, η οποία δηλώνει την καθυστέρηση στην κατάσταση i προτού πραγματοποιηθεί μετάβαση στην j. Μία από αυτές τις επιλεγμένες καθυστερήσεις θα είναι τελικά και η μικρότερη, με αποτέλεσμα να πραγματοποιηθεί η αντίστοιχη μετάβαση (race condition: the faster one wins).

5.3. Παροδικές και μακροπρόθεσμες πιθανότητες (TP και SSP) για την EPACSR

Λόγω του γεγονότος ότι στην EPACSR συνυπάρχουν στοχαστικές και πιθανοτικές μεταβάσεις, γίνεται εύκολα αντιληπτό πως μόνο ο πίνακας P (βλ DTMCs) μπορεί εύκολα να σχηματιστεί. Για το λόγο αυτό χρησιμοποιήσουμε τεχνικές των Semi-Markov Chains (SMCs) για υπολογισμό των πιθανοτήτων που μας ενδιαφέρουν . Στις SMCs ο χρόνος παραμονής σε κάθε κατάσταση ακολουθεί μία γενική κατανομή $F_i(t)$ με πυκνότητα πιθανότητας $f_i(t)$, σε αντίθεση με τις CTMCs όπου έχουμε εκθετικές κατανομές. Για τον υπολογισμό των SSPs στις SMCs, αρχικά βρίσκουμε λύση για το διακριτό κομμάτι (embedded DTMC) το οποίο χαρακτηρίζεται από τον πίνακα P. Αυτό γίνεται για το λόγο ότι η συμπεριφορά των μεταβάσεων στις SMCs και στις DTMCs είναι η ίδια, αφού με βάση κάποιας πιθανότητας πραγματοποιείται η κάθε μετάβαση. Έτσι, ως πρώτο βήμα, υπολογίζουμε το διάνυσμα $\underline{\upsilon}$ με την βοήθεια του τύπου

$$\underline{\upsilon} = \underline{\upsilon}P \Rightarrow \underline{\upsilon}(I - P) = \underline{0}$$

που δίνει τις μακροπρόθεσμες πιθανότητες σε DTMC. Στη συνέχεια υπολογίζεται ο μέσος χρόνος παραμονής h_i για όλες τις καταστάσεις i, αφού στις SMCs επικρατεί κάποια καθυστέρηση σε κάθε κατάσταση:

$$h_i = \int_0^\infty t\, f_i(t)dt$$

Ακολούθως υπολογίζονται οι SSPs με την ενσωμάτωση των μέσων χρόνων παραμονής ως εξής:

$$\pi_i = \frac{\upsilon_i h_i}{\sum_j \upsilon_j h_j}$$

Οι CTMCs αποτελούν ειδική περίπτωση των SMCs. Λόγω της εκθετικής κατανομής, ο μέσος χρόνος παραμονής είναι ίσος με $h_i = \left|q_{i,i}\right|^{-1}$, δηλ το αντίστροφο του ρυθμού της εκθετικής κατανομής. Άρα λοιπόν, οι SSPs για τις CTMCs μπορούν να υπολογιστούν από τον πιο πάνω τύπο, με την βοήθεια των (embedded) DTMCs όπου $p_{i,j} = \dfrac{q_{i,j}}{\left|q_{i,i}\right|}$.

Περνάμε τώρα στον υπολογισμό των μεταβατικών πιθανοτήτων. Ως γνωστό μία CTMC μπορεί να περιγραφτεί πλήρως από τον πίνακα Q και το αρχικό διάνυσμα πιθανοτήτων $\underline{\pi}(0)$. Αν υποθέσουμε ότι το σύστημα βρίσκεται στην κατάσταση $X(t)$, τότε για $h \to 0$ έχουμε:

$$\Pr\{X(t+h) = j \mid X(t) = i\} = \Pr\{X(h) = j \mid X(0) = i\} = q_{i,j}h + o(h), \qquad i \neq j$$

Δηλώνοντας ως $\pi_i(t) = \Pr\{X(t) = i\}$ την πιθανότητα η κατάσταση, την χρονική στιγμή t, να είναι i, έχουμε:

$$\pi_i(t+h) = \pi_i(t)\Pr\{\text{stay at i during}\,[t,t+h)\} + \sum_{j\neq i}\pi_j(t)\Pr\{\text{go from j to i during}\,[t,t+h)\} =$$

$$= \pi_i(t)(1-\sum_{j\neq i} q_{i,j}h) + (\sum_{j,m\neq i}\pi_j(t)q_{j,m}p_{m,i})h + o(h)$$

όπου στον δεύτερο όρο λάβαμε υπόψη το γεγονός πως όταν πραγματοποιηθεί μία στοχαστική μετάβαση, υπάρχει η δυνατότητα το σύστημα να βρεθεί σε μία πιθανοτική κατάσταση και έτσι, στιγμιαία, να πραγματοποιηθεί ακόμη μία (πιθανοτική) μετάβαση σε μηδενικό χρόνο. Γνωρίζοντας ότι $q_{i,i} = -\sum_{j\neq i} q_{i,j}$, έχουμε:

$$\pi_i(t+h) = \pi_i(t) + (\sum_{j,m}\pi_j(t)q_{j,m}p_{m,i})h + o(h)$$

$$\frac{\pi_i(t+h) - \pi_i(t)}{h} = (\sum_{j,m}\pi_j(t)q_{j,m}p_{m,i})h + o(h)$$

και, παίρνοντας το όριο $h \to 0$ έχουμε:

$$\pi_i'(t) = \sum_{j,m}\pi_j(t)q_{j,m}p_{m,i}$$

όπου σε μορφή πινάκων και διανυσμάτων έχει την μορφή:

$$\underline{\pi}'(t) = \pi(t)QP \qquad \text{δοθέντος του } \underline{\pi}(0)$$

Βλέπουμε λοιπόν ότι οι TPs για τις CTMCs περιγράφονται από ένα σύστημα γραμμικών διαφορικών εξισώσεων. Ως λύση, προτείνουμε την μέθοδο *Uniformisation*, γνωστή και ως μέθοδος του Jensen [28].

5.4. Μοντελοέλεγχος για την EPACSR

Στην ενότητα αυτή θα χρησιμοποιήσουμε την χρονική λογική για εκφράσουμε τους διάφορους περιορισμούς (constraints) στις μετρήσεις που θέλουμε να αξιολογήσουμε καθώς και τις προδιαγραφές ενός συστήματος.

Χρονική λογική

Σύνταξη: Θα χρησιμοποιήσουμε την χρονική λογική **CSL** [30] και τους συμβολισμούς p για την πιθανότητα, με $p \in [0,1]$ και $\trianglelefteq$ για τους τελεστές σύγκρισης $\leq$ και $\geq$. Οι τύποι καταστάσεων (state formulas) στην CSL συντάσσονται με βάση τα πιο κάτω:

$$\Phi ::= \alpha \mid \neg\Phi \mid \Phi \vee \Phi \mid S_{\trianglelefteq}(\Phi) \mid P_{\trianglelefteq}(\Phi)$$

Οι δύο πιθανοτικοί τελεστές S και P απευθύνονται στις μακροπρόθεσμες και παροδικές συμπεριφορές της CTMC αντίστοιχα. Συγκεκριμένα ο τελεστής S αναφέρεται στην πιθανότητα το σύστημα να παραμείνει μακροπρόθεσμα σ' ένα συγκεκριμένο σύνολο καταστάσεων, ενώ ο τελεστής P μας επιτρέπει να αναφερόμαστε στην πιθανότητα εμφάνισης ενός συγκεκριμένου μονοπατιού μέσα στην CTMC. Ο τελεστής $P_{\trianglelefteq}(.)$ αντικαθιστά τους όρους $\exists$και$\forall$, αφού μπορούμε να γράψουμε τον όρο $\exists\varphi$ ως $P_{>0}(\varphi)$ και τον όρο $\forall\varphi$ ως $P_{>1}(\varphi)$, όπου φ κάποιο μονοπάτι.

Για κάποιο συγκεκριμένο διάστημα $I \subseteq R_{\geq 0}$, η σύνταξη της CSL για μονοπάτια (path formulas) έχει ως εξής:

$$\varphi ::= X^{I}\Phi \mid \Phi U^{I}\Phi$$

όπου με X^{I} συμβολίζουμε το επόμενο βήμα (next step) και με U^{I} τον όρο until.

Σημασιολογία: Έστω s μία κατάσταση της CTMC που εξετάζουμε και Φ ιδιότητα/ τύπος κατάστασης (state formula). Τότε ο όρος $s \models \Phi$ είναι σωστός αν η Φ ισχύει στην κατάσταση s. Η σημασιολογία της CSL έχει ως εξής:

- $s \models \alpha$ iff $\alpha \in L(s)$ όπου L(s) το σύνολο των ατομικών προτάσεων
- $s \models \neg\Phi$ iff $s \not\models \Phi$
- $s \models \Phi_1 \vee \Phi_2$ iff $s \models \Phi_1 \vee s \models \Phi_2$
- $s \models S_{\trianglelefteq p}(\Phi)$ if and only if $\sum_{s' \models \Phi} \pi_{s'}(s) \trianglelefteq p$
- $s \models P_{\trianglelefteq p}(\varphi)$ if and only if $\mathrm{Prob}(s,\varphi) \trianglelefteq p$

Ο τύπος $s \models S_{\trianglelefteq p}(\Phi)$ δηλώνει ότι η SSP για το σύνολο των Φ-καταστάσεων ικανοποιεί το όριο $\trianglelefteq p$ και με $\pi_{s'}(s)$ δηλώνεται η SSP.

Τέλος, όσον αφορά τα μονοπάτια, αν ορίσουμε με:

- **σ**, ένα μονοπάτι $s_0 t_0 s_1 t_1 ....$ του μοντέλου με $t_i \in R_{\geq 0}$ τον χρόνο παραμονής στην κατάσταση s_i
- **σ[i]**, την (i+1) – κατάσταση στο σ
- **σ@t**, την κατάσταση που βρισκόμαστε σε χρόνο t

τότε η σημασιολογία που αφορά τα μονοπάτια έχει ως εξής:

- $\sigma \models X^I \Phi$ if and only if $\sigma[1] \models \Phi \wedge \delta(\sigma,0) \in I$
- $\sigma \models \Phi_1 U^I \Phi_2$ if and only if $\exists t \in I. (\sigma@t \models \Phi_2 \wedge \forall u \in [0,t). \sigma@u \models \Phi_1)$

όπου $\delta(\sigma,0) = t_0$ η διάρκεια παραμονής στην αρχική κατάσταση s_0 της σ.

Δεδομένες της ικανότητας υπολογισμού παροδικών και μακροπρόθεσμων πιθανοτήτων του μοντέλου μας ο αλγόριθμος μοντελοελέγχου είναι όμοιος με αυτούς των [31, 32].

6. Συμπεράσματα

Έχουμε περιγράψει τη στοχαστική άλγεβρα διεργασιών EPACSR η οποία έχει δημιουργηθεί για τη μοντελοποίηση συστημάτων που προέρχονται από εφαρμογές σε δικτυακά συστήματα. Η άλγεβρα αυτή επιτρέπει την ιεραρχική περιγραφή συστημάτων στοιχείο εξαιρετικά χρήσιμο για τη συνθετική μοντελοποίηση των πολλαπλών στρωμάτων που πιθανόν να απαρτίζουν ένα δικτυακό σύστημα. Επίσης συνδυάζει τους χρονικούς τελεστές, την πιθανοτική και τη στοχαστική συμπεριφορά. Η σημασιολογία της άλγεβρας διεργασιών δίνεται ως συστήματα μεταβάσεων με μη-ντετερμινιστική, πιθανοτική και στοχαστική συμπεριφορά (συνδυασμός των διακριτών και συνεχών αλυσίδων του Μάρκοβ. Το μοντέλο αυτό δεν έχει μέχρι στιγμής τύχει επεκταμένης μελέτης στην περιοχή των τυπικών μεθόδων ανάλυσης συστημάτων. Εμείς προτείναμε μεθόδους ελέγχου του μοντέλου για (1) έλεγχο εννοιών ισοδυναμίας (ισχυρή δυπροσομοίωση) και (2) μεθόδους υπολογισμού παροδικών και μακροπρόθεσμων πιθανοτήτων και μέσω αυτών μεθόδους μοντελοελέγου συστημάτων υλοποιημένων στο μοντέλο αυτό.

Βιβλιογραφία

1. E. Clarke and R. Kurshnan. *Computer-aided Verification.* IEEE Spectrum, 33(6). IEEE, 1996.
2. R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
3. C. A. R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.
4. J. Baeten and P. Weijland. *Process Algebra*. Volume 18 of Cambridge Tracts in Computer Science. Cambridge University Press, 1990.
5. J. A. Bergstra, A. Ponse and S.A. Smolka, editors. *Handbook of Process Algebra*. Elsevier Science Publishers, 2001.
6. W. J. Fokkink. *Introduction to Process Algebra*. Texts in Theoretical Computer Science, Springer, 2000.
7. G. Lowe. *Probabilistic and prioritized models of timed CSP*. Theoretical Computer Science, 138:315-352, 1995.
8. A. Philippou, O. Sokolsky, R. Cleaveland, I. Lee, S. Smolka. Probabilistic Resource Failure in Real-Time Process Algebra. In Proceedings of *CONCUR'98*, pages 389-404. Springer, 1998.
9. H. Hansson and B. Jonsson. *A logic for reasoning about time and probability*. Formal Aspects of Computing, 6:512—535. 1994.
10. P. Bremond-Gregoire, J. Choi and I. Lee. *A complete axiomatization of finite-state ACSR processes*. Information and Computation 138(2):124--159, 1997.
11. S Schneider. *An operational semantics for timed CSP*. In Information and Computation, 116:193-213, 1995.
12. M. Bernardo and R. Gorrieri, *A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time*. In Theoretical Computer Science, 202(1-2):1-54, 1998.
13. E. Brinksma, H. Hermanns and J.-P. Katoen, editors. Lectures *on Formal Methods and Performance Analysis*, LNCS 2090. Springer, 2001.
14. E. Brinksma and H. Hermanns. Process algebra and Markov chains. In [13].

15. H. Hermanns. Interactive Markov Chains. PhD thesis, U. Erlangen-Nurnberg, 1998.

16. H. Hermanns and J.-P. Katoen. Performance Evaluation := (Process Algebra + Model Checking) Markov Chains. In Proceedings of *CONCUR'01*, pages 59-81.

17. U. Herzog and V. Mertsiotakis. Stochastic process algebra applied to failure modeling. In Proceedings of *PAPM'94*, pages 107-126.

18. J. Hillston. A Compositional Approach to Performance Modelling. Cambridge University Press, 1996.

19. H. Hermanns, U. Herzog, J.-P. Katoen. *Process Algebra for Performance Evaluation*. Theoretical Computer Science, 274:43-87, 2002.

20. G. Priami. Stochastic π-calculus with general distributions. In Proceedings of *PAPM'96*, pages 41-57.

21. S. Gilmore, J. Hillston, R. Holton and M. Rettelbach. *Specification in stochastic process algebra for a robot control problem.* International Journal of Production Res., 34(4):1065-1080, 1996.

22. H. Hermanns, U. Herzog, U. Klehmet, V. Mertsiotakis and M. Siegle. *Compositional performance modeling with the TIPP-Tool*. In Performance Evaluation, 39, 2000.

23. D. Park. Concurrency and automata on infinite sequences. In Proceedings of TCS'81.

24. R. Paige and R. Tarjan. Three *Partition Refinement Algorithms*. SIAM Journal of Computing, 16(6):973-989, 1987.

25. P. Kanellakis and S. A. Smolka. CCS Expressions, Finite State Processes, and Three Problems of Equivalence. Information and Computation, 86:43-68, 1990.

26. K. Christou, G. Hadjipollas and A. Philippou. Exponential Distributions in EACSR. Technical Report TR 2003 – 05, University of Cyprus, 2003.

27. C. Derman. Finite State Markovian Decision Processes. Academic Press, 1970.

28. D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

29. H. Haverkort. Markovian models for performance and dependability evaluation. In *Lecture Notes on Formal Methods and Performance Analysis*. Springer, 2001.

30. C. Baier, J.-P Katoen and H. Hermanns. Approximate symbolic model checking of continuous-time Markov chains. In *CONCUR 1999*, pages 146-162.

31. C. Baier, J.-P Katoen and H. Hermanns. Model checking continuous time Markov chains by transient analysis. In *CAV 2000*, pages 358-372. IEEE 2001.

32. J.-P. Katoen, M. Kwiatkowska, G. Norman and D. Parker. Faster and symbolic CTMC model checking. In *PAPM/PROBMIV* 2001.

33. H. Hermanns and M. Siegle. *Bisimulation algorithms for stochastic process algebras and their BDD-based implementation*. In Formal Methods for Real-Time and Probabilistic Systems, LNCS1601, 1999, pages 244-265.

Παράρτημα A

K. Christou, G. Hadjipollas and A. Philippou. Exponential Distributions in EACSR. Technical Report TR 2003 – 05, University of Cyprus, 2003.