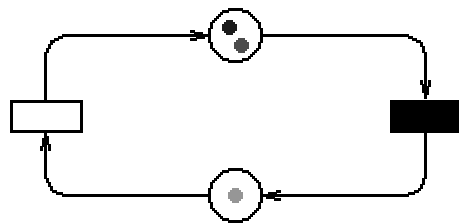


ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣ/ΝΙΚΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μοντελοποίηση και Προσομοίωση με Χρωματισμένα Δίκτυα Petri

Πτυχιακή Εργασία



Όνοματεπώνυμο : Αβραμίδου Ευτέρπη
ΑΕΜ : 476
Επιβλέπων : κ. Παναγιώτης Κατσαρός

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. Εισαγωγή.....	7
2. Εισαγωγή στις Βασικές Έννοιες των Δικτύων Petri.....	9
2.1. Ιστορία των Δικτύων Petri.....	9
2.2. Βασικές Έννοιες της Θεωρίας Γράφων.....	10
2.3. Φορμαλισμός των Δικτύων Petri.....	11
2.3.1. Βρόγχοι Αυτοεπανάληψης και Καθαρά Δίκτυα Petri.....	14
2.3.2. Υποδίκτυα ενός Δικτύου Petri.....	14
2.3.3. Μονοπάτια.....	14
2.3.4. Συνδεσμικότητα (Connectivity).....	15
2.3.5. Κυκλώματα.....	15
2.4. Γραμμική Αλγεβρική Προσέγγιση των Δικτύων Petri.....	15
2.5. Μαρκάρισμα κι Εκτέλεση ενός Δικτύου Petri.....	16
2.5.1. Μαρκάρισμα.....	17
2.5.2. Εκτέλεση.....	17
2.5.3. Ακολουθία Πυροδοτήσεων.....	20
2.6. Ιδιότητες Δικτύων Petri.....	20
2.6.1. Προσιτότητα (Reachability).....	21
2.6.2. Περατότητα (Boundedness) και Ασφάλεια (Safety).....	21
2.6.3. Διάρκεια (Liveness).....	22
2.6.4. Αντιστρεψιμότητα (Reversibility) και Αρχική Κατάσταση (Home State).....	24
2.6.5. Δικαιοσύνη (Fairness) και Κάλυψη (Coverability).....	24
2.6.6. Επιμονή (Persistence).....	25
2.6.7. Απόσταση Συγχρονισμού (Synchronic Distance).....	25
2.6.8. Συντηρητικότητα (Conservation).....	25
2.6.9. Συνέπεια (Consistence).....	26
2.7. Μοντελοποίηση με τη Βοήθεια των Δικτύων Petri.....	26
2.7.1. Σύγκρουση (Conflict).....	26
2.7.2. Ταυτοχρονισμός (Concurrency) και Συγχρονισμός (Synchronisation).....	27
2.7.3. Σύγχυση (Confusion).....	28
2.8. Επεκτάσεις των Δικτύων Petri.....	30
2.9. Μέθοδοι Ανάλυσης.....	31
2.9.1. Μέθοδος Ανάλυσης μέσω Προσιτότητας.....	32

2.9.2.	Μέθοδος Ανάλυσης μέσω Αμετάβλητων Διανυσμάτων.....	33
2.9.3.	Μέθοδος Ανάλυσης βάσει της Κατηγορίας του Δικτύου Petri.....	34
2.9.4.	Μέθοδοι Ανάλυσης μέσω Απλοποίησης και Σύνθεσης.....	34
3.	Κατηγορίες Δικτύων Petri.....	35
3.1.	Δίκτυα Petri Χαμηλού Επιπέδου.....	35
3.2.	Δίκτυα Petri Υψηλού Επιπέδου.....	37
3.2.1.	Κατηγορίες δικτύων Petri Υψηλού Επιπέδου.....	38
3.3.	Χρονικές Εκδόσεις Δικτύων Petri.....	40
3.4.	Δίκτυα Petri με Χρονισμένες Μεταβάσεις.....	44
3.4.1.	Χρονισμένα Δίκτυα Petri.....	45
3.4.2.	Χρονικά Δίκτυα Petri.....	49
3.4.3.	Μνήμη.....	51
3.4.4.	Πολλαπλή Ενεργοποίηση (Multiple Enabling).....	53
3.5.	Δίκτυα με Χρονισμένες Θέσεις.....	55
3.6.	Στοχαστικές Εκδόσεις.....	56
3.6.1.	Στοχαστικά Δίκτυα Petri.....	56
3.6.2.	Γενικευμένα Στοχαστικά Δίκτυα Petri.....	60
3.7.	Δίκτυα Petri με Ουρές.....	63
4.	Χρωματισμένα Δίκτυα Petri.....	65
4.1.	Εισαγωγή στα Χρωματισμένα Δίκτυα Petri.....	65
4.1.1.	Χρωματισμένα Δίκτυα Petri και Δίκτυα Θέσης-Μετάβασης	78
4.2.	Προσομοίωση των Χρωματισμένων Δικτύων Petri.....	79
4.2.1.	Τμήματα Κώδικα (Code Segments).....	84
4.3.	Ανάλυση του Χώρου Καταστάσεων των Χρωματισμένων Δικτύων Petri.....	88
4.4.	Χρωματισμένα Δίκτυα Petri με Χρόνο.....	98
4.5.	Ιεραρχικά Χρωματισμένα Δίκτυα Petri.....	109
4.5.1.	Γράφος Ιεραρχίας Σελίδων.....	107
4.5.2.	Θέσεις Συγχώνευσης.....	109
4.6.	Συμπεράσματα.....	110

5. CPN Tools.....	113
5.1. Τι είναι το πρόγραμμα CPN Tools ;.....	113
5.2. Ξεκινώντας με το CPN Tools.....	113
5.2.1. Η Διεπιφάνεια.....	113
5.2.2. Το Ευρετήριο.....	114
5.2.3. Οι Παλέτες.....	115
5.2.4. Τα Μενού Μαρκαρισμάτων.....	117
5.3. Σχεδιασμός ενός Χρωματισμένου Δικτύου Petri.....	118
5.3.1. Δημιουργία ενός Νέου Δικτύου.....	118
5.3.2. Άνοιγμα ενός Δικτύου.....	119
5.3.3. Αποθήκευση ενός Δικτύου.....	120
5.3.4. Εικόνα ενός Δικτύου.....	121
5.3.5. Άνοιγμα μιας Σελίδας.....	123
5.3.6. Δημιουργία Θέσεων, Μεταβάσεων και Τόξων.....	126
5.3.7. Σύνταξη Δηλώσεων.....	127
5.3.8. Εισαγωγή Επιγραφών.....	130
5.4. Ιεραρχικά Δίκτυα.....	130
5.4.1. Ετικέτες Ιεραρχίας.....	131
5.4.1.1. Ετικέτες Υποσελίδας.....	131
5.4.1.2. Ετικέτες Συνόλου Συγχώνευσης.....	131
5.4.1.3. Ετικέτες Τύπου Πύλης.....	132
5.4.2. Δημιουργία Μεταβάσεων Αντικατάστασης.....	132
5.4.2.1. Από Πάνω-προς-τα-Κάτω Ανάπτυξη.....	132
5.4.2.3. Από Κάτω-προς-τα-Πάνω Ανάπτυξη.....	134
5.4.3. Πλοήγηση μεταξύ των Υποσελίδων και των Υπερσελίδων.....	137
5.4.4. Θέσεις Συγχώνευσης.....	137
5.4.4.1. Δημιουργία Θέσεων Συγχώνευσης.....	138
5.4.4.2. Μέλη ενός Συνόλου Συγχώνευσης.....	138
5.5. Ανάλυση ενός Χρωματισμένου Δικτύου Petri.....	139
5.5.1. Συντακτικός Έλεγχος.....	139
5.5.2. Προσομοίωση.....	143
5.5.2.1. Αναφορά Προσομοίωσης.....	144
5.5.3. Ανάλυση του Χώρου Καταστάσεων.....	146
5.5.3.1. Είσοδος στο Εργαλείο του Χώρου Καταστάσεων.....	146
5.5.3.2. Υπολογισμός του Χώρου Καταστάσεων και του SCC-γράφου.....	148
5.5.3.3. Αποθήκευση της Αναφοράς του Χώρου Καταστάσεων.....	148
5.5.3.4. Υποβολή Ερωτημάτων.....	149
5.5.4. Το Δείπνο των Φιλοσόφων.....	150

ΚΕΦΑΛΑΙΟ 1^ο

Εισαγωγή

Τα δίκτυα Petri (Petri Nets/PNs) είναι μια τυπική και γραφικά ελκυστική γλώσσα που είναι κατάλληλη για τη μοντελοποίηση συστημάτων με χαρακτηριστικά σύγχρονης και ασύγχρονης εκτέλεσης, κατανομής και παραλληλίας και μη ντετερμινιστικής/στοχαστικής συμπεριφοράς. Ουσιαστικά η γλώσσα αυτή, είναι μια γενίκευση της θεωρίας αυτομάτων, τέτοια ώστε να μπορεί να εκφρασθεί η έννοια των ταυτόχρονα εκτελούμενων γεγονότων. Ως γραφικό εργαλείο, τα δίκτυα Petri μπορούν να χρησιμοποιηθούν σαν βοήθεια οπτικής μορφής παρόμοια με τα διαγράμματα ροής και τα μπλοκ διαγράμματα, ενώ ως μαθηματικό εργαλείο είναι δυνατό να κατασκευαστούν εξισώσεις κατάστασης, αλγεβρικές εξισώσεις ή άλλα μαθηματικά μοντέλα που ρυθμίζουν τη συμπεριφορά των συστημάτων.

Τα δίκτυα Petri αναπτύχθηκαν στις δεκαετίες του '60 και του '70 και σύντομα αναγνωρίστηκαν ως μία από τις πιο κατάλληλες και αξιόπιστες γλώσσες για την περιγραφή και την ανάλυση του συγχρονισμού, της επικοινωνίας, των συγκρούσεων και του διαμοιρασμού πόρων μεταξύ των ταυτόχρονων διεργασιών. Μέχρι σήμερα τα δίκτυα Petri έχουν χρησιμοποιηθεί με επιτυχία στη μοντελοποίηση και στην προσομοίωση κατανομημένων και παράλληλων συστημάτων, στην εκτίμηση της απόδοσης συστημάτων, σε πρωτόκολλα επικοινωνίας καθώς και σε ανεκτικά σε σφάλματα συστήματα.

Αρχικά, τα δίκτυα Petri χρησιμοποιήθηκαν για τη μελέτη των ποιοτικών ιδιοτήτων (qualitative properties) συστημάτων που χαρακτηρίζονται από ταυτοχρονισμό και συγχρονισμό. Η χρήση τους για ποιοτική (quantitative) ανάλυση των συστημάτων απαιτούσε την εισαγωγή χρονικών προδιαγραφών στα βασικά μοντέλα δικτύων Petri. Επίσης, σε πάρα πολλά κατανομημένα και ταυτόχρονα συστήματα, ο χρόνος παίζει σημαντικό ρόλο. Η σωστή λειτουργία πολλών συστημάτων εξαρτάται από το χρόνο που διαρκούν συγκεκριμένες δράσεις και οι διαφορετικές αποφάσεις που παίρνονται κατά τη σχεδίαση ενός συστήματος, μπορεί να έχουν σημαντική επίδραση στην απόδοση του συστήματος. Έτσι, για να μπορούν τα δίκτυα Petri να μελετούν θέματα απόδοσης και να εξετάζουν τόσο την ποιότητα (πχ, τη χρονική καθυστέρηση), όσο και την ποσότητα μια υπηρεσίας (πχ, τη ρυθμοαπόδοση) που παρέχεται από το σύστημα, εισήχθηκε η έννοια του χρόνου στο μοντέλο των δικτύων Petri.

Παρ'όλα αυτά, οι προσπάθειες να χρησιμοποιηθούν τα δίκτυα Petri στην πράξη, αποκάλυψαν δύο σημαντικά μειονεκτήματα. Πρώτα απ'όλα δεν υπήρχαν κατάλληλες δομές για την αναπαράσταση δεδομένων κι έτσι τα μοντέλα πολλές φορές γινότουσαν υπερβολικά μεγάλα, γιατί η διαχείριση όλων των δεδομένων έπρεπε να αναπαρασταθεί απ'ευθείας από τη δομή του δικτύου. Δεύτερον, δεν υπήρχαν ιεραρχικές δομές κι έτσι ήταν αδύνατον να κατασκευαστεί ένα μεγάλο μοντέλο με τη βοήθεια ενός συνόλου ξεχωριστών υπομοντέλων με καλά ορισμένες διεπιφάνειες.

Η ανάπτυξη των δικτύων Petri υψηλού επιπέδου (high level Petri nets) στο τέλος της δεκαετίας του '70 και των ιεραρχικών δικτύων Petri (hierarchical Petri nets) στο τέλος της δεκαετίας του '80 έδωσε λύση σε αυτά τα δύο σοβαρά προβλήματα. Τα Χρωματισμένα δίκτυα Petri (Coloured Petri nets) αποτελούν μια από τις πιο γνωστές κατηγορίες των δικτύων Petri υψηλού επιπέδου. Τα Χρωματισμένα δίκτυα Petri ενσωματώνουν και τη δόμηση δεδομένων και την ιεραρχική αποσύνθεση, χωρίς να αλλοιώνουν τις αξίες των πρωτότυπων δικτύων Petri.

Η ανάπτυξη των Χρωματισμένων δικτύων Petri οφείλεται στην επιθυμία να αναπτυχθεί μια γλώσσα μοντελοποίησης βιομηχανικής ισχύος, που να είναι ταυτόχρονα, καλά ορισμένη σε θεωρητικό επίπεδο και πολύπλευρη, ώστε να μπορεί να χρησιμοποιηθεί πρακτικά σε συστήματα του μεγέθους και της πολυπλοκότητας, που συναντώνται στα τυπικά βιομηχανικά έργα. Για το επιτύχουμε αυτό, συνδύσαμε τη δύναμη των δικτύων Petri με τη δύναμη των γλωσσών προγραμματισμού. Τα δίκτυα Petri παρέχουν τις κατάλληλες δομές για την περιγραφή του συγχρονισμού των ταυτόχρονων διεργασιών, ενώ μια γλώσσα προγραμματισμού παρέχει τις κατάλληλες δομές για τον ορισμό των τύπων δεδομένων και τη διαχείριση των τιμών τους.

Τα Χρωματισμένα δίκτυα Petri παρέχουν ένα πλαίσιο εργασίας για την κατασκευή και την ανάλυση των κατανεμημένων και των ταυτόχρονων συστημάτων. Το μοντέλο ενός Χρωματισμένου δικτύου Petri ενός συστήματος περιγράφει τις καταστάσεις στις οποίες μπορεί να βρίσκεται ένα σύστημα και τις μεταβάσεις μεταξύ των διαφόρων καταστάσεων. Τα Χρωματισμένα δίκτυα Petri έχουν χρησιμοποιηθεί σε μια μεγάλη ποικιλία εφαρμογών, όπως είναι τα πρωτόκολλα επικοινωνίας, οι εφαρμογές που βασίζονται στην τεχνολογία του World Wide Web, τα λειτουργικά συστήματα, ο σχεδιασμός συστημάτων υλικού και λογισμικού, τα συστήματα ήχου/εικόνας και τα στρατιωτικά συστήματα.

Η οργάνωση του κειμένου της διπλωματικής εργασίας γίνεται ως εξής : Στο Κεφάλαιο 2 γίνεται μια πρώτη γνωριμία με τα δίκτυα Petri και περιγράφονται οι βασικές έννοιες, οι ιδιότητες, οι δομές που μοντελοποιούν και οι μέθοδοι ανάλυσης τους. Στο Κεφάλαιο 3 παρουσιάζονται οι δύο βασικές κατηγορίες των δικτύων Petri, τα δίκτυα Petri χαμηλού επιπέδου και τα δίκτυα Petri υψηλού επιπέδου καθώς και τα χαρακτηριστικά τους. Στη συνέχεια περιγράφονται αναλυτικά οι χρονικές και οι στοχαστικές εκδόσεις των δικτύων Petri. Στο Κεφάλαιο 4 παρουσιάζονται αναλυτικά τα Χρωματισμένα δίκτυα Petri με τη βοήθεια ενός παραδείγματος ενός απλού πρωτοκόλλου επικοινωνίας. Περιγράφονται επίσης τα Χρωματισμένα δίκτυα Petri με χρονικές προδιαγραφές και τα Χρωματισμένα δίκτυα Petri με ιεραρχική δομή. Τέλος, στο Κεφάλαιο 5 παρουσιάζεται το πιο γνωστό εργαλείο λογισμικού που χρησιμοποιεί Χρωματισμένα δίκτυα Petri για τη μοντελοποίηση και την ανάλυση συστημάτων, το CPN Tools. Στο τέλος του κεφαλαίου αυτού γίνεται εφαρμογή του εργαλείου CPN Tools στο πρόβλημα του δείπνου των φιλοσόφων (dining philosophers problem).

ΚΕΦΑΛΑΙΟ 2^ο

Εισαγωγή στις Βασικές Έννοιες των Δικτύων Petri

2.1 Ιστορία των Δικτύων Petri

Τα δίκτυα Petri γεννήθηκαν το 1962, στη Φυσικομαθηματική Σχολή του Τεχνικού Πανεπιστημίου του Ντάρμστατ στη Γερμανία, από τη διδακτορική διατριβή του Carl Adam Petri, “Kommunikation mit Automaten”, που είχε ως αντικείμενο την επικοινωνία μεταξύ αυτόματων μηχανών. Χρησιμοποιώντας ένα δίκτυο, περιέγραψε την τυπική σχέση μεταξύ των γεγονότων σε ένα σύστημα υπολογιστών. Ήταν η πρώτη φορά που διατυπώθηκε μια γενική θεωρία για τα διακριτά παράλληλα συστήματα.

Η δουλειά του C.A.Petri τράβηξε την προσοχή του A. W. Holt και άλλων επιστημόνων από το Information System Theory Project of Applied Research, Inc. στις Ηνωμένες Πολιτείες. Εκείνοι πρότειναν ότι τα δίκτυα Petri θα μπορούσαν να χρησιμοποιηθούν για την μοντελοποίηση και την ανάλυση συστημάτων με ταυτόχρονα γεγονότα. Επίσης, το Computational Structure Group του MIT, καθοδηγούμενο από τον καθηγητή J. B. Dennis, πρόσεξε την δουλειά του Petri, ενώ στην δεκαετία του '70 έγιναν και πολλές διδακτορικές διατριβές πάνω στο αντικείμενο.

Από τα τέλη της δεκαετίας του '70, τα δίκτυα Petri αποτέλεσαν μια ιδιαίτερος ενεργή περιοχή, κυρίως στην Ευρώπη. Ετήσια συνέδρια γύρω από τις Εφαρμογές και την Θεωρία των δικτύων Petri γίνονται από το 1979, ενώ το πρώτο βιβλίο με συλλογή μελετών και κειμένων γύρω από τα δίκτυα εκδόθηκε το 1981 [Peterson, 1981] και το δεύτερο, με ιδιαίτερο ενδιαφέρον για την Ευρώπη, το 1985 [Resig, 1985]. Οι περισσότερες μελέτες αφορούσαν σε συστήματα επεξεργασίας δεδομένων που απασχολούσαν τους ανθρώπους της κοινωνίας των ηλεκτρονικών υπολογιστών.

Στις αρχές τις δεκαετίας του '80, αναλυτές με βάσεις μηχανικών άρχισαν να ασχολούνται με την χρήση δικτύων Petri σε εφαρμογές μηχανικών, και κυρίως σε Αυτόματα Συστήματα Κατεργασιών. Στην πορεία ανακαλύφθηκε ότι τα δίκτυα Petri ήταν ένα ισχυρό εργαλείο στην περιγραφή **συστημάτων που εξαρτώνται από γεγονότα (event driven systems)**. Αυτά τα συστήματα μπορεί να ήταν ασύγχρονα, να περιέχουν σειριακές ή παράλληλες εργασίες, να περιλαμβάνουν συγκρούσεις, αμοιβαίο αποκλεισμό, και να μην είναι ντετερμινιστικά. Τέτοια συστήματα χαρακτηρίζονται ως συστήματα διακριτών γεγονότων ή **Δυναμικά Συστήματα Διακριτών Γεγονότων (Discrete Event Dynamic Systems, DEDS)**. Πολλά βιβλία, κυρίως προς τα τέλη της δεκαετίας του '80, ασχολήθηκαν με εφαρμογές των δικτύων Petri σε προβλήματα μηχανικών, είτε αυτά ήταν η μοντελοποίηση και η σύνθεση μεθόδων για τον έλεγχο συστημάτων κατεργασιών μέσω διακριτών γεγονότων, είτε η γενικότερη χρήση δικτύων Petri στην ανάλυση, την αποτίμηση των επιδόσεων, και τον έλεγχο συστημάτων παραγωγής.

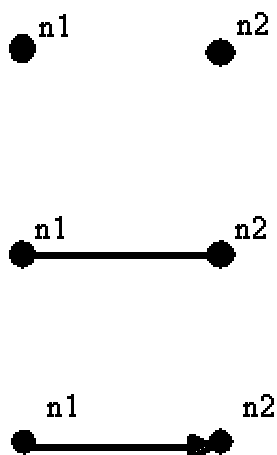
2.2 Βασικές Έννοιες της Θεωρίας Γράφων

Εφόσον τα δίκτυα Petri αποτελούν μια ειδική κατηγορία γράφων, η παρουσίαση τους θα αρχίσει με μια μικρή αναφορά στις βασικές έννοιες από τη θεωρία γράφων. Ένας γράφος αποτελείται από 2 τύπους στοιχείων, τους κόμβους και τις ακμές. Επίσης, χαρακτηριστικός για κάθε γράφο, είναι και ο τρόπος με τον οποίο συνδέονται οι κόμβοι και οι ακμές μεταξύ τους. Ο επίσημος ορισμός των γράφων είναι :

Ορισμός 1 : Ένας γράφος G ορίζεται ως μια τριάδα $G = (V, E, \phi)$ όπου V το μη κενό σύνολο των κόμβων του γράφου, E το σύνολο των ακμών του γράφου και ϕ η συνάρτηση που συνδέει το σύνολο των ακμών E στο σύνολο των ζευγών των στοιχείων του V .

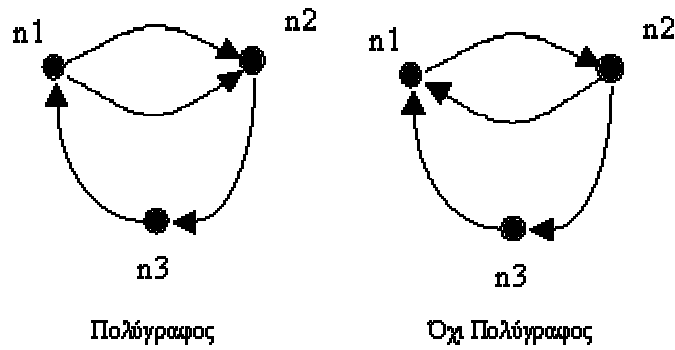
Σε ένα γράφο G στον οποίον ανήκουν οι κόμβοι i και j συμβολίζουμε με το ζεύγος (i, j) την ακμή που ενώνει τα i και j . Αν το ζεύγος των κόμβων που συνδέονται μεσω μιας ακμής είναι διατεταγμένο, δηλαδή έχει οριστεί η σειρά των μελών του ζεύγους των ακμών (i, j) , τότε η ακμή είναι κατευθυνόμενη και μπορεί να αντικατασταθεί από ένα βέλος το οποίο θα δείχνει την κατεύθυνση. Αν όλες οι ακμές του γράφου είναι κατευθυνόμενες, τότε ο γράφος ονομάζεται **κατευθυνόμενος γράφος**.

Τρία παραδείγματα γράφων φαίνονται στο Σχήμα 1. Ο πρώτος γράφος που αποτελείται μόνο από δύο ασύνδετους κόμβους, μπορεί να θεωρηθεί και ως κατευθυνόμενος και ως μη κατευθυνόμενος. Ο δεύτερος γράφος είναι μη κατευθυνόμενος, ενώ ο τρίτος είναι κατευθυνόμενος



Σχήμα 1: Παραδείγματα γράφων

Σε ένα γράφο, δύο κόμβοι που συνδέονται με μια ακμή ονομάζονται **γειτονικοί κόμβοι (adjacent nodes)**. Όταν ένας γράφος περιέχει παράλληλες ακμές, δηλαδή ακμές που συνδέουν το ίδιο ζευγάρι κόμβων και που, εάν είναι κατευθυνόμενες, έχουν την ίδια κατεύθυνση, τότε ο γράφος ονομάζεται **πολύγραφος (multigraph)**. Στο Σχήμα 2 φαίνονται δύο γράφοι από τους οποίους, ο πρώτος είναι πολύγραφος ενώ ο δεύτερος δεν είναι, γιατί οι πολλαπλές ακμές που συνδέουν τους κόμβους $n1$, $n2$ δεν έχουν την ίδια κατεύθυνση.



Σχήμα 2: Παραδείγματα πολυγράφων

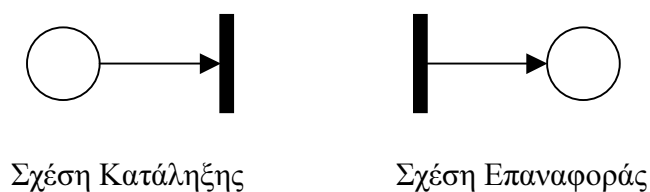
2.3 Φορμαλισμός των Δικτύων Petri

Τα δίκτυα Petri είναι κατευθυνόμενοι διμερείς γράφοι. Ο όρος “διμερείς” σημαίνει ότι έχουν δύο είδη κόμβων. Διαφορετικά σύμβολα χρησιμοποιούνται για να διακρίνονται τα δύο είδη. Ο πρώτος τύπος κόμβων ονομάζεται **θέση (place)** και ο δεύτερος **μετάβαση (transition)**. Οι θέσεις αναπαριστώνται με κύκλους ενώ οι μεταβάσεις αναπαριστώνται με κατακόρυφες γραμμές (μπάρες). Οι ακμές ενός δικτύου Petri είναι πάντα κατευθυνόμενες και ονομάζονται **τόξα (arcs)**. Στο Σχήμα 3 φαίνονται τα σύμβολα.



Σχήμα 3: Βασικά συστατικά των δικτύων Petri

Στους διμερείς γράφους μια ακμή μπορεί να συνδέσει μόνο δύο κόμβους που ανήκουν σε διαφορετικό τύπο. Έτσι, στα Δίκτυα Petri τα τόξα είτε ξεκινούν από μία θέση και καταλήγουν σε μία μετάβαση, ορίζοντας μια σχέση κατάληξης, είτε ξεκινούν από μία μετάβαση και καταλήγουν σε μία θέση, ορίζοντας μια σχέση επαναφοράς. Θέσεις που μέσω των τόξων οδηγούν σε μία μετάβαση, λέγονται **θέσεις εισόδου (input places)** για αυτήν την μετάβαση ενώ θέσεις στις οποίες καταλήγουν τα τόξα από μία μετάβαση λέγονται **θέσεις εξόδου (output places)** για αυτήν τη μετάβαση. Στο Σχήμα 4 φαίνονται οι σχέσεις κατάληξης και επαναφοράς.



Σχήμα 4: Σχέσεις στα δίκτυα Petri

Σε ένα δίκτυο Petri οι μεταβάσεις αναπαριστούν τα γεγονότα ή τις εργασίες που λαμβάνουν χώρα όπως εξωτερικά γεγονότα, ολοκλήρωση υπολογιστικών διαδικασιών, μετάδοση πακέτου δεδομένων, μετάδοση μηνύματος, λογικές προτάσεις που γίνονται αληθείς κτλ. Οι θέσεις εισόδου μιας μετάβασης αναπαριστούν προϋποθέσεις για το γεγονός που αντιστοιχεί στη μετάβαση ή απαραίτητα δεδομένα εισόδου, ενώ οι θέσεις εξόδου μιας μετάβασης αναπαριστούν αποτελέσματα του γεγονότος που αντιστοιχεί στη μετάβαση ή λογικές προτάσεις που έπονται της μετάβασης.

Ο επίσημος ορισμός των δικτύων Petri είναι:

Ορισμός 2 : Ένα δίκτυο Petri είναι ένας διμερής κατευθυνόμενος γράφος και ορίζεται από τη διατεταγμένη τετράδα $PN = (P, T, I, O)$, όπου :

(I) $P = \{p_1, p_2, \dots, p_n\}$, $n > 0$, είναι το πεπερασμένο σύνολο όλων των θέσεων του δικτύου Petri.

(II) $T = \{t_1, t_2, \dots, t_m\}$, $m > 0$, είναι το πεπερασμένο σύνολο όλων των μεταβάσεων του δικτύου Petri, όπου $P \cup T \neq \emptyset$ και $P \cap T = \emptyset$.

(III) $I : P \times T \rightarrow \mathbb{N}$, είναι η συνάρτηση εισόδου που ορίζει το σύνολο των προσανατολισμένων τόξων από τις P στις T , όπου $\mathbb{N} = \{0, 1, 2, \dots\}$

(IV) $O : P \times T \rightarrow \mathbb{N}$, είναι η συνάρτηση εξόδου που ορίζει το σύνολο των προσανατολισμένων τόξων από τις T στις P .

Το δίκτυο Petri που ορίστηκε παραπάνω, ονομάζεται **δίκτυο Θέσης-Μετάβασης (Place-Transition net / PT-net)** και είναι το πιο διαδεδομένο μοντέλο δικτύων Petri. Τα τέσσερα στοιχεία (P, T, I, O) αποτελούν την δομή του δικτύου Petri που ορίζεται από ένα προσανατολισμένο γραφικό δίκτυο. Όταν $I(p, t) > 1$, τότε ο αριθμός $I(p, t)$ ονομάζεται και **βάρος (weight)** του τόξου ή των τόξων από το p στο t . Το ίδιο ισχύει και για το $O(p, t)$ το οποίο ονομάζεται βάρος του τόξου ή των τόξων από το t στο p . Το I και το O παριστάνουν δύο πίνακες $n \times s$ θετικών ακέραιων στοιχείων.

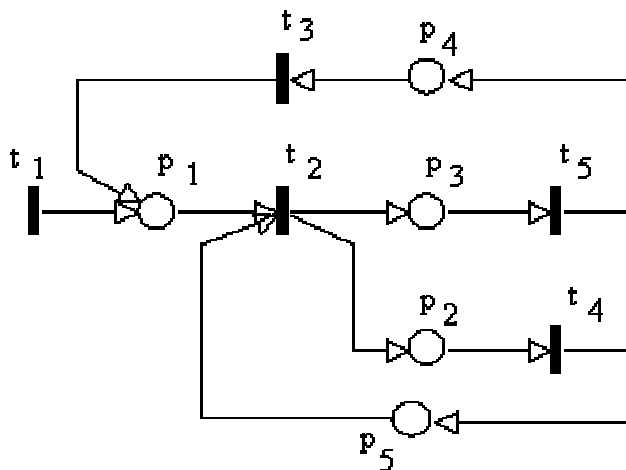
Ένα δίκτυο Petri μπορεί εναλλακτικά να οριστεί ως (P, T, F, W) , όπου F είναι ένα υποσύνολο $\{P \times T\} \cup \{T \times P\}$, που αναπαριστά το σύνολο όλων των τόξων, ενώ το $W : F \rightarrow \mathbb{N}$ αναπαριστά την πολλαπλότητα ή το βάρος των τόξων.

Αν ισχύει $I(p, t) = k$ (ή $O(p, t) = k$), τότε υπάρχουν k προσανατολισμένα τόξα που συνδέουν την θέση p με την μετάβαση t (ή την μετάβαση t με την θέση p). Αν ισχύει $I(p, t) = 0$ (ή $O(p, t) = 0$), τότε δεν υπάρχουν προσανατολισμένα τόξα που συνδέουν την θέση p με την μετάβαση t (ή την μετάβαση t με την θέση p). Ένα μοναδικό τόξο υπάρχει αν $k = 1$. Για τις περιπτώσεις όπου $k > 1$, είτε υπάρχουν k παράλληλα τόξα που συνδέουν μια θέση (ή μια μετάβαση) με μια μετάβαση (ή μια θέση), ή ένα τόξο το οποίο όμως έχει ως επιγραφή την πολλαπλότητα ή το βάρος k όταν χρησιμοποιείται.

Όταν τα βάρη όλων των τόξων ενός δικτύου Petri είναι ίσα με τη μονάδα ($k = 1$), το δίκτυο Petri ονομάζεται **Κοινό δίκτυο Petri (Ordinary Petri net)**. Το Κοινό δίκτυο Petri ορίζεται ως μια υποκλάση του Δικτύου Θέσης-Μετάβασης με θέσεις άπειρης

χωρητικότητας και μοναδιαία βάρη τόξων, αλλά έχει την ίδια δύναμη μοντελοποίησης με ένα δίκτυο Θέσης-Μετάβασης. Είναι πιο βολικό για ανάλυση, αλλά λιγότερο αποδοτικό.

Στο Σχήμα 5 φαίνεται ένα παράδειγμα ενός δικτύου Petri το οποίο συμβολίζουμε με PN1.



Σχήμα 5: Το δίκτυο Petri PN1

Η δομή του Δικτύου Petri PN1, δηλαδή η τετράδα που το ορίζει, μπορεί να παρουσιαστεί ως εξής :

$$P = \{p1, p2, p3, p4, p5\}$$

$$T = \{t1, t2, t3, t4, t5\}$$

$$\begin{array}{ccccc} I(p1, t1) = 0 & I(p2, t1) = 0 & I(p3, t1) = 0 & I(p4, t1) = 0 & I(p5, t1) = 0 \\ I(p1, t2) = 1 & I(p2, t2) = 0 & I(p3, t2) = 0 & I(p4, t2) = 0 & I(p5, t2) = 1 \\ I(p1, t3) = 0 & I(p2, t3) = 0 & I(p3, t3) = 0 & I(p4, t3) = 1 & I(p5, t3) = 0 \\ I(p1, t4) = 0 & I(p2, t4) = 1 & I(p3, t4) = 0 & I(p4, t4) = 0 & I(p5, t4) = 0 \\ I(p1, t5) = 0 & I(p2, t5) = 0 & I(p3, t5) = 1 & I(p4, t5) = 0 & I(p5, t5) = 0 \end{array}$$

$$\begin{array}{ccccc} O(t1, p1) = 1 & O(t2, p1) = 0 & O(t3, p1) = 1 & O(t4, p1) = 0 & O(t5, p1) = 0 \\ O(t1, p2) = 0 & O(t2, p2) = 1 & O(t3, p2) = 0 & O(t4, p2) = 0 & O(t5, p2) = 0 \\ O(t1, p3) = 0 & O(t2, p3) = 1 & O(t3, p3) = 0 & O(t4, p3) = 0 & O(t5, p3) = 0 \\ O(t1, p4) = 0 & O(t2, p4) = 0 & O(t3, p4) = 0 & O(t4, p4) = 0 & O(t5, p4) = 1 \\ O(t1, p5) = 0 & O(t2, p5) = 0 & O(t3, p5) = 0 & O(t4, p5) = 1 & O(t5, p5) = 0 \end{array}$$

Το σύνολο όλων των θέσεων εισόδου της μετάβασης t συμβολίζεται με $\bullet t$ και ονομάζεται **προσύνολο (preset)** της t , ενώ το σύνολο των θέσεων εξόδου της μετάβασης t συμβολίζεται με $t\bullet$ και ονομάζεται **μετασύνολο (postset)** της t . Ανάλογα για τις θέσεις ισχύουν : το προσύνολο $\bullet p$ ορίζει το σύνολο όλων των μεταβάσεων εισόδου της θέσης p , ενώ το μετασύνολο $p\bullet$ ορίζει το σύνολο όλων των μεταβάσεων εξόδου της θέσης p . Όταν μία θέση έχει κενό προσύνολο ονομάζεται **αρχική θέση (start place)**, ενώ όταν έχει κενό μετασύνολο ονομάζεται **τελική θέση (final place)**. Για το δίκτυο Petri PN1 του Σχήματος 5 ισχύουν οι ακόλουθες σχέσεις :

$$\bullet p1 = \{t1, t3\} \quad \text{και} \quad t2\bullet = \{p2, p3\}$$

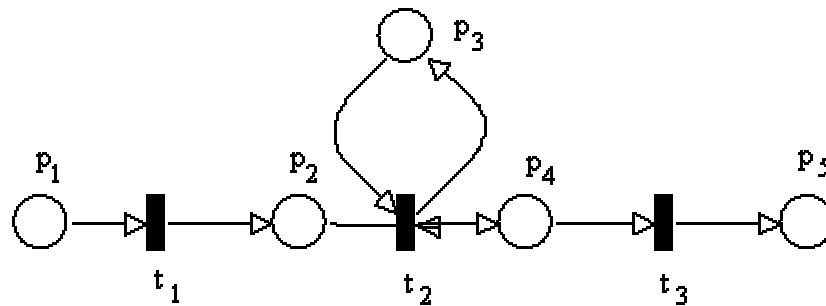
για το προσύνολο του p_1 και το μετασύνολο του t_2 .

Ορίζοντας το σύνολο των θέσεων, το σύνολο των μεταβάσεων, και τα προσύνολα και τα μετασύνολα είτε όλων των θέσεων ή όλων των μεταβάσεων, έχουμε μια ισοδύναμη παρουσίαση της δομής των δικτύων Petri. Τα προσύνολα και τα μετασύνολα είναι πολύ χρήσιμα στην περιγραφή αλγορίθμων για την ανάλυση των δικτύων Petri.

2.3.1 Βρόγχοι Αυτοεπανάληψης και Καθαρά Δίκτυα Petri

Μία θέση p και μία μετάβαση t αποτελούν έναν **κλειστό βρόγχο (self loop)**, εάν η θέση p είναι ταυτόχρονα θέση εισόδου και θέση εξόδου για τη μετάβαση t . Ένα δίκτυο Petri θα είναι **καθαρό (pure)** αν δεν περιέχει κλειστούς βρόγχους.

Το δίκτυο Petri PN1 του Σχήματος 5 είναι καθαρό, ενώ το δίκτυο Petri PN2 του Σχήματος 6 δεν είναι, γιατί περιέχει το κλειστό βρόγχο (t_2, p_3).



Σχήμα 6: Το δίκτυο Petri PN2 με το κλειστό βρόγχο (t_2, p_3)

2.3.2 Υποδίκτυα ενός Δικτύου Petri

Το **υποδίκτυο (subnet)** ενός δικτύου Petri $PN = (P, T, I, O)$ είναι ένα δίκτυο Petri $PN_s = (P_s, T_s, I_s, O_s)$ τέτοιο ώστε : $P \supseteq P_s$, $T \supseteq T_s$ και I_s, O_s είναι οι περιορισμοί του I και του O για $P_s \times T_s$ και $T_s \times P_s$, αντίστοιχα.

Ο κλειστός βρόγχος του PN2 στο Σχήματος 6 είναι ένας πιθανός υπογράφος :

$$PN_{21} = (P_1, T_1, I_1, O_1)$$

όπου $P_1 = \{p_3\}$, $T_1 = \{t_2\}$, $I(p_3, t_2) = 1$ και $O(t_2, p_3) = 1$.

2.3.3 Μονοπάτια

Το **μονοπάτι (path)** είναι ένα σύνολο από k κόμβους και $k-1$ τόξα, για κάποιο ακέραιο k , τέτοιο ώστε το i -οστό τόξο ενώνει είτε τον i -οστό κόμβο με τον $i+1$ -οστό κόμβο, είτε τον $i+1$ -οστό κόμβο με τον i -οστό κόμβο. Ένα μονοπάτι είναι

κατευθυνόμενο, αν για όλα τα $i = 1, 2, \dots, k$, το i -οστό τόξο ενώνει είτε τον i -οστό κόμβο με τον $i+1$ -οστό κόμβο.

Ένα μονοπάτι στο οποίο κανένα τόξο δεν διασχίζεται παραπάνω από μια φορά ονομάζεται **απλό μονοπάτι (simple path)**, ενώ ένα μονοπάτι στο οποίο κανένας κόμβος δεν διασχίζεται παραπάνω από μια φορά ονομάζεται **στοιχειώδες μονοπάτι (elementary path)**. Είναι φανερό ότι όλα τα στοιχειώδη μονοπάτια είναι και απλά, αλλά το αντίστροφο δεν ισχύει.

Στο Σχήμα 6 το μονοπάτι $(p_1, t_1, p_2, t_2, p_4, t_3, p_5)$ είναι στοιχειώδες μονοπάτι και ταυτόχρονα απλό. Το μονοπάτι $(p_1, t_1, p_2, t_2, p_3, t_2, p_4, t_3, p_5)$ είναι απλό μονοπάτι αλλά όχι στοιχειώδες, αφού η θέση t_2 διασχίζεται δύο φορές.

2.3.4 Συνδεσμικότητα (Connectivity)

Ένα δίκτυο Petri είναι **συνδεδεμένο (connected)**, αν και μόνο αν υπάρχει ένα μονοπάτι – όχι απαραίτητα κατευθυνόμενο – από έναν κόμβο σε οποιοδήποτε άλλον κόμβο, ενώ είναι **ισχυρά συνδεδεμένο (strongly connected)**, αν και μόνο αν υπάρχει ένα κατευθυνόμενο μονοπάτι από έναν κόμβο σε οποιοδήποτε άλλον κόμβο.

Το δίκτυο στο Σχήμα 5, PN1, είναι συνδεδεμένο αλλά όχι ισχυρά συνδεδεμένο : δεν υπάρχει κατευθυνόμενο μονοπάτι από τη θέση p_1 στη μετάβαση t_1 για παράδειγμα. Ένα δίκτυο με τους ίδιους κόμβους και με τις ίδιες συνδέσεις με το PN1, εκτός από τη t_1 και τη σύνδεση της με τη p_1 , θα ήταν ισχυρά συνδεδεμένο. Το δίκτυο PN2 στο Σχήμα 6 είναι επίσης συνδεδεμένο και, όπως φαίνεται ξεκάθαρα, όχι ισχυρά συνδεδεμένο.

Όταν μία μετάβαση έχει μόνο εξερχόμενα τόξα, ονομάζεται **μετάβαση πηγής (source transition)**, ενώ όταν έχει μόνο εισερχόμενα τόξα ονομάζεται **μετάβαση καταβόθρας (sink transition)**. Έαν ένα δίκτυο έχει τουλάχιστον μία μετάβαση πηγής ή μία μετάβαση απόθεσης, τότε δεν μπορεί να είναι ισχυρά συνδεδεμένο.

2.3.5 Κυκλώματα

Ένα **κατευθυνόμενο κύκλωμα (directed circuit)** είναι ένα κατευθυνόμενο μονοπάτι από έναν κόμβο πίσω στον εαυτό του και ένα **κατευθυνόμενο στοιχειώδες κύκλωμα (directed elementary circuit)** είναι ένα κατευθυνόμενο κύκλωμα στο οποίο κανένας κόμβος δεν εμφανίζεται παραπάνω από μια φορά.

Το δίκτυο PN1 στο Σχήματος 5 έχει δύο κατευθυνόμενα στοιχειώδη κυκλώματα : το $p_1 - t_2 - p_3 - t_5 - p_4 - t_3 - p_1$ και το $t_2 - p_2 - t_4 - p_5 - t_2$.

Τα κατευθυνόμενα στοιχειώδη κυκλώματα παίζουν σημαντικό ρόλο στην ανάλυση απόδοσης των μοντέλων των συστημάτων διακριτών γεγονότων που περιγράφονται από τα δίκτυα Petri.

2.4 Γραμμική Αλγεβρική Προσέγγιση των Δικτύων Petri

Στη προηγούμενη παράγραφο, τα δίκτυα Petri παρουσιάστηκαν χρησιμοποιώντας μια προσέγγιση επηρεασμένη από τη θεωρία γράφων. Τα δίκτυα Petri μπορούν επίσης να περιγραφούν με όρους από την αριθμητική ακεραίων. Η ικανότητα της παράστασης των μοντέλων και γραφικά και με όρους από την αριθμητική ακεραίων, τα κάνει ιδιαίτερα ελκυστικά ως εργαλεία σχεδίασης με τη βοήθεια υπολογιστή, όπου ο χρήστης μπορεί να αλληλεπιδρά γραφικά με τα δίκτυα σε κάποια οθόνη, ενώ την ίδια στιγμή μπορεί να χρησιμοποιεί μια πλειάδα από αλγορίθμους για να υποστηρίξει την ανάλυση.

Η τοπολογική δομή ενός καθαρού δικτύου Petri μπορεί να παρασταθεί από έναν πίνακα ακεραίων C , που ονομάζεται **πίνακας προοπτώσεων ή πίνακας ροής (incidence or flow matrix)**. Ο C είναι ένας $n \times m$ πίνακας, του οποίου οι n γραμμές αντιστοιχούν στις θέσεις του δικτύου και οι m στήλες στις μεταβάσεις του. Ο C ορίζεται ως εξής :

$$C_{ij} = O(t_j, p_i) - I(p_i, t_j) \text{ για } 1 \leq i \leq n, 1 \leq j \leq m.$$

Ο παραπάνω ορισμός περιορίζεται μόνο στα καθαρά δίκτυα Petri, γιατί στα μη-καθαρά δίκτυα Petri υπάρχει ένα πρόβλημα με την έννοια ότι, οι κλειστοί βρόγχοι δεν μπορούν να παρασταθούν στον πίνακα προοπτώσεων : το 1 και το -1 αναιρούν το ένα το άλλο και δίνουν μηδέν στον πίνακα, με αποτέλεσμα να χαθεί η ένδειξη της ύπαρξης του κλειστού βρόγχου.

Η συνάρτηση εισόδου I και η συνάρτηση εξόδου O μπορούν να ξανασηματιστούν από τον πίνακα C με τον παρακάτω εύκολο τρόπο :

$$O(t_j, p_i) = \max \{ C_{ij}, 0 \} \text{ και } I(p_i, t_j) = \min \{ C_{ij}, 0 \}$$

Ο πίνακας προοπτώσεων του δικτύου Petri PN_1 δίνεται παρακάτω :

$$C_1 = \begin{bmatrix} 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & -1 & 0 & 1 & 0 \end{bmatrix}$$

2.5 Μαρκάρισμα κι Εκτέλεση ενός Δικτύου Petri

Τα δίκτυα Petri δε θα ήταν πολύ χρήσιμα, αν ήταν απλώς διαγράμματα τα οποία θα περιέγραφαν τις σχέσεις μεταξύ των αντικειμένων που αναπαρίστανται από τους κόμβους. Ένα βασικό χαρακτηριστικό των δικτύων Petri είναι ότι μπορούν να εκτελεστούν. Έτσι μπορεί κάποιος να παρατηρήσει τις αλληλεπιδράσεις μεταξύ των συστατικών ενός δικτύου Petri και να μελετήσει τη δυναμικότητα του συστήματος που μοντελοποιείται από αυτό.

Οι θέσεις, οι μεταβάσεις και τα τόξα κάνουν το δίκτυο Petri ένα προσανατολισμένο γράφημα και αποτελούν τη δομή του δικτύου. Για να περιγραφεί η δυναμική του δικτύου Petri εισάγεται ένα τέταρτο αντικείμενο, το οποίο ονομάζεται **μάρκα (token)** και αναπαρίσταται από ένα μικρό, γεμάτο κύκλο ●. Κάθε θέση μπορεί να έχει καμία μάρκα ή θετικό αριθμό από μάρκες. Οι μάρκες μέσα σε κάποια θέση μπορούν να σημαίνουν τον αριθμό των πόρων, να υποδηλώνουν αν μια συνθήκη είναι αληθής, ή να δηλώνουν αν μια εργασία είναι σε λειτουργία. Αυτό εξαρτάται από το τι υποδηλώνει η θέση στην οποία βρίσκονται.

Χωρητικότητα μιας θέσης ονομάζεται ο μέγιστος αριθμός από μάρκες που μπορεί να περιέχει. Οι θέσεις μπορούν να περιέχουν αυθαίρετο αριθμό από μάρκες και οπότε να μιλάμε για **δίκτυα άπειρης χωρητικότητας (infinite capacity nets)** ή να έχουν περιορισμένη χωρητικότητα και οπότε να μιλάμε για **δίκτυα πεπερασμένης χωρητικότητας (finite capacity nets)**.

2.5.1 Μαρκάρισμα

Το **μαρκάρισμα (marking)** ενός δικτύου Petri PN, συμβολίζεται με M και είναι μια συνάρτηση $P \rightarrow N$, όπου $N = \{0, 1, 2, \dots\}$, η οποία αναθέτει ένα μη-αρνητικό ακέραιο αριθμό σε κάθε θέση του δικτύου. Ένα μαρκάρισμα μπορεί να αναπαρασταθεί με ένα n-διάστατο ακέραιο διάνυσμα, του οποίου τα στοιχεία αντιστοιχούν στις θέσεις του δικτύου. Έτσι, ο i-οστός όρος του διανύσματος παριστάνει τον αριθμό από μάρκες στην i-οστή θέση. Το διάνυσμα μαρκάριατος παριστάνει την κατάσταση του δικτύου Petri, δηλαδή η διανομή από μάρκες στις θέσεις του δικτύου ορίζει την κατάσταση του. Η κατάσταση του συστήματος αλλάζει όταν αλλάζει η διανομή από μάρκες. Το αρχικό μαρκάρισμα του δικτύου συμβολίζεται με M_0 .

Στα Σχήματα 5 και 6 δεν υπάρχουν μάρκες και η αρχική κατάσταση αυτών των δικτύων δηλώνεται από τα ακόλουθα μηδενικά διανύσματα :

$$PN1: M_0 = [0 \ 0 \ 0 \ 0 \ 0]_T \text{ και } PN2: M_0 = [0 \ 0 \ 0 \ 0 \ 0]_T$$

2.5.2 Εκτέλεση

Όταν όλες οι θέσεις εισόδου μιας μετάβασης διαθέτουν ικανό αριθμό από μάρκες, το γεγονός που υποδεικνύεται από την μετάβαση μπορεί να πραγματοποιηθεί. Αυτό αποκαλείται **πυροδότηση μετάβασης (transition firing)**. Η πυροδότηση αλλάζει την διανομή από μάρκες στις θέσεις, δηλώνοντας την αλλαγή στην κατάσταση του συστήματος. Ένα απλό τόξο από μια θέση σε μια μετάβαση υποδεικνύει ότι η μετάβαση απαιτεί μια μάρκα σε αυτήν τη θέση για να πυροδοτήσει. Σε έναν πολύγραφο, η πολλαπλότητα του τόξου υποδεικνύει τον αριθμό από μάρκες που πρέπει να περιέχει η θέση, ώστε να πυροδοτήσει η μετάβαση.

Οι κανόνες εκτέλεσης των δικτύων Petri είναι οι εξής :

- (i) Μια μετάβαση $t \in T$ ενεργοποιείται, όταν και μόνο όταν :

$$M(p) \geq I(p,t), \forall p \in P,$$

όπου $M(p)$ είναι το μαρκάρισμα της θέσης p.

- (ii) Η μετάβαση t που ενεργοποιείται σε ένα μαρκάρισμα M , πυροδοτεί και ως αποτέλεσμα προκύπτει ένα νέο μαρκάρισμα M' , όπου:
- $$M'(p) = M(p) - I(p, t) + O(p, t), \forall p \in P.$$

Ο πρώτος από τους κανόνες εκτέλεσης είναι ο κανόνας ενεργοποίησης, ενώ ο δεύτερος είναι ο κανόνας πυροδότησης.

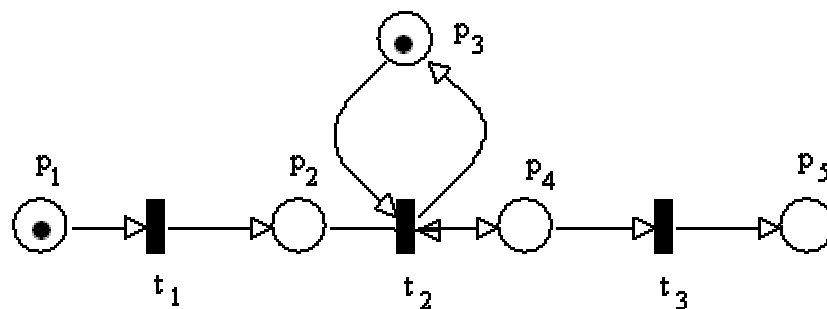
Ο κανόνας ενεργοποίησης λέει ότι αν όλες οι θέσεις εισόδου μιας μετάβασης t περιέχουν τον απαραίτητο αριθμό από μάρκες για την πυροδότηση της μετάβασης, τότε η μετάβαση ενεργοποιείται και ονομάζεται **ενεργοποιημένη (enabled)**. Μόνο ενεργοποιημένες μεταβάσεις μπορούν να πυροδοτήσουν. Αυτό σημαίνει ότι αν οι συνθήκες σχετικά με την πραγματοποίηση ενός γεγονότος ικανοποιούνται πλήρως, τότε το γεγονός λαμβάνει χώρα. Σημειώνεται ότι, από πλευράς λογικής, οι συνθήκες αυτές δημιουργούν μια σχέση AND.

Ο κανόνας πυροδότησης λέει ότι μια ενεργοποιημένη μετάβαση t πυροδοτεί ή ισοδύναμα ένα γεγονός συμβαίνει. Η πυροδότηση αυτή μπορεί να εξεταστεί σε δύο διαφορετικά στάδια. Στο πρώτο στάδιο, αφαιρείται ο αριθμός των μαρκών που χρειάζονται από την κάθε μία θέση εισόδου, και ο αριθμός αυτός ισούται με το άθροισμα των τόξων που συνδέουν την θέση εισόδου με την μετάβαση t (ή ισοδύναμα με το άθροισμα των πολλαπλοτήτων τους). Στην παραπάνω εξίσωση, αυτό δηλώνεται από το $-I(p, t)$. Σε δεύτερο στάδιο, τοποθετούνται οι μάρκες στην κάθε μία θέση εξόδου από την μετάβαση t , και ο αριθμός των μαρκών ισούται με το άθροισμα των τόξων από την t στις συνδεόμενες θέσεις εξόδου (ή ισοδύναμα με το άθροισμα των πολλαπλοτήτων τους). Στην παραπάνω εξίσωση, αυτό δηλώνεται από το $+O(p, t)$.

Το δίκτυο Petri PN3 του Σχήματος 7 έχει αρχική κατάσταση που δηλώνεται από το ακόλουθο διάνυσμα μαρκαρίσματος :

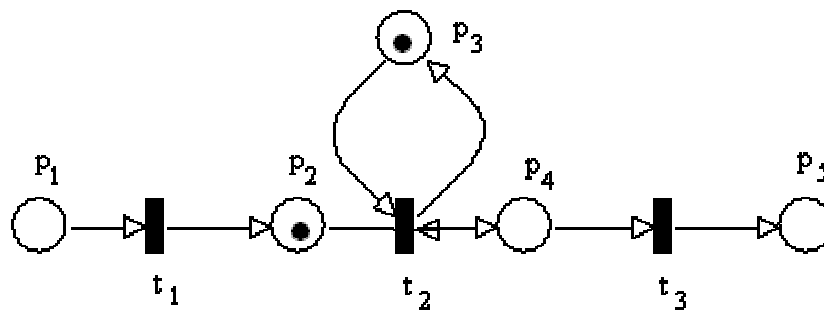
$$PN3 : M_0 = [1 \ 0 \ 1 \ 0 \ 0]_T$$

Η μετάβαση t_1 είναι ενεργοποιημένη, ενώ η t_2 δεν είναι, γιατί μία τις θέσεις εισόδου της δεν περιέχει μάρκα.



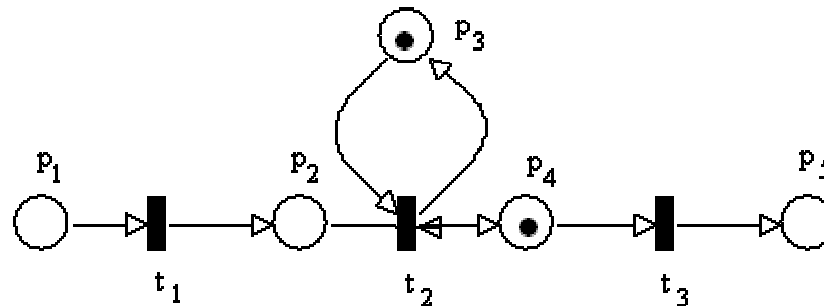
Σχήμα 7: Το δίκτυο Petri PN3 (αρχική κατάσταση)

Όταν η μετάβαση t_1 πυροδοτεί, η μάρκα αφαιρείται από τη θέση p_1 και μια νέα μάρκα εμφανίζεται στη θέση p_2 , όπως φαίνεται στο Σχήμα 8. Στα δίκτυα Petri η λειτουργία της πυροδότησης δεν μετακινεί μάρκες διαμέσου του δικτύου. Η μάρκα στη θέση p_1 δεν μετακινήθηκε στη θέση p_2 .



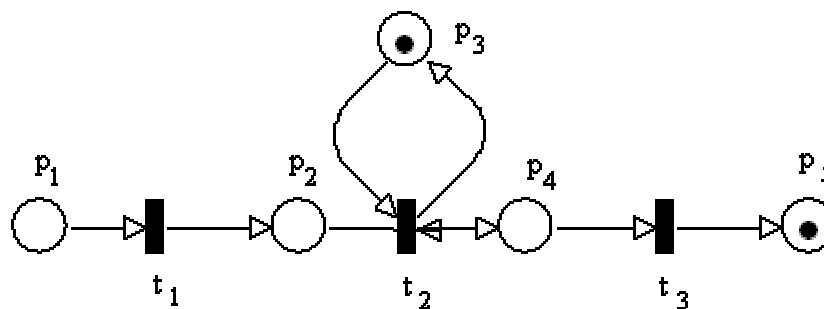
Σχήμα 8: Η κατάσταση του PN3 μετά από 1 πυροδότηση

Τώρα, η μετάβαση t_2 είναι ενεργοποιημένη, αφού υπάρχει μία μάρκα σε κάθε μία από τις θέσεις εισόδου, p_2 και p_3 . Όταν η μετάβαση t_2 πυροδοτήσει, οι μάρκες στις θέσεις p_2 και p_3 αφαιρούνται και νέες μάρκες παράγονται στις θέσεις p_3 και p_4 , όπως φαίνεται στο Σχήμα 9. Το μαρκάρισμα της θέσης p_3 δεν μεταβάλλεται μετά την πυροδότηση της μετάβασης t_2 , αφού τα p_3 και t_2 σχηματίζουν ένα κλειστό βρόγχο.



Σχήμα 9: Η κατάσταση του PN3 μετά από 2 πυροδοτήσεις

Η μετάβαση t_3 είναι ενεργοποιημένη τώρα και πυροδοτεί. Η μάρκα αφαιρείται από τη θέση p_4 και μια νέα μάρκα γεννιάται στη θέση p_5 , όπως φαίνεται στο Σχήμα 10.



Σχήμα 10: Η κατάσταση του PN3 μετά από 3 πυροδοτήσεις

Καμία άλλη μετάβαση δεν είναι ενεργοποιημένη και η εκτέλεση του δικτύου ολοκληρώνεται. Η παραπάνω διαδικασία αναφέρεται σαν **παιχνίδι μαρκών (token game)**.

Με δίκτυα Petri είναι δυνατή η παράσταση τόσο ανοικτών όσο και κλειστών συστημάτων. **Κλειστό σύστημα** θεωρείται εκείνο που μία φορά του δίνεται η αρχική ώθηση με τη διανομή από τις μάρκες και αναλογικά παράγει αιώνια. **Ανοιχτό σύστημα** θεωρείται εκείνο που κάθε παραγωγή αντιστοιχεί σε μία αρχική ώθηση. Το δίκτυο Petri PN3 των Σχημάτων 7-10, μοντελοποιεί ένα ανοικτό σύστημα, αφού του δόθηκε μια αρχική ώθηση (οι μάρκες στις θέσεις p1 και p3) και μετά από τρεις πυροδοτήσεις, η εκτέλεση του ολοκληρώθηκε και σταμάτησε να παράγει μάρκες. Για να αρχίσει μια νέα παραγωγή από μάρκες πρέπει να δοθεί μια νέα αρχική ώθηση.

Επίσης υπάρχουν δύο τρόποι πυροδότησης μεταβάσεων για δίκτυα Petri πεπερασμένης χωρητικότητας : η **αυστηρή πυροδότηση** και η **ασθενής πυροδότηση**. Στην αυστηρή πυροδότηση, μια ενεργοποιημένη μετάβαση πυροδοτεί αν και μόνο αν, μετά την πυροδότησή της ο αριθμός από μάρκες σε κάθε θέση στο μετασύνολό της δεν υπερβαίνει τη χωρητικότητα της θέσης. Στην ασθενή πυροδότηση: μια ενεργοποιημένη μετάβαση πυροδοτεί ανεξάρτητα από τη χωρητικότητα των θέσεων του μετασυνόλου της.

2.5.3 Ακολουθία Πυροδοτήσεων

Η **ακολουθία πυροδοτήσεων (firing sequence)** των μεταβάσεων t_1, t_2, \dots, t_s θα συμβολίζεται ως εξής :

$$\sigma_s = t_1 \bullet t_2 \bullet \dots \bullet t_{s-1} \bullet t_s$$

Το σύνολο όλων των ακολουθιών πυροδοτήσεων συμβολίζεται με T^* .

Για το δίκτυο PN3, το παράδειγμα των Σχημάτων 7-10, η ακολουθία πυροδοτήσεων είναι η $\sigma_3 = t_1 \bullet t_2 \bullet t_3$. Για το παράδειγμα του Σχήματος 1.5, PN1, οι μόνες πιθανές ακολουθίες πυροδοτήσεων είναι οι σειρές δυνάμεων της t_1 , οι οποίες συμβολίζονται με t_1^m , όπου το m είναι ένας θετικός ακέραιος.

2.6 Ιδιότητες Δικτύων Petri

Τα δίκτυα Petri κατέχουν ένα μεγάλο αριθμό ιδιοτήτων. Οι ιδιότητες διακρίνονται σε δύο είδη, στις ιδιότητες που αφορούν τη συμπεριφορά του συστήματος και ονομάζονται **ιδιότητες συμπεριφοράς (behavioral properties)** και σε αυτές που αφορούν τη δομή του συστήματος και ονομάζονται **ιδιότητες δομής (structural properties)**. Οι ιδιότητες συμπεριφοράς εξαρτώνται από το αρχικό μαρκάρισμα του δικτύου, ενώ στον αντίποδα, οι ιδιότητες δομής εξαρτώνται μόνο από την δομή ή την τοπολογία του δικτύου. Οι ιδιότητες συμπεριφοράς είναι πιο σημαντικές από πλευράς πρακτικότητας και περιλαμβάνουν την **προσιτότητα (reachability)**, την **περατότητα (boundedness)**, την **ασφάλεια (safety)**, τη **διάρκεια (liveness)**, την **αντιστρεψιμότητα (reversibility)**, την **οικεία κατάσταση (home state)**, τη **δικαιοσύνη (fairness)**, την **κάλυψη (coverability)**, την **επιμονή (persistence)** και την **απόσταση συγχρονισμού (synchronic distance)**. Οι ιδιότητες δομής περιλαμβάνουν τη **συντηρητικότητα (conservation)**, τη **δομική περατότητα (structural boundedness)**, τη **δομική διάρκεια (structural liveness)** και τη **συνέπεια (consistence)**.

2.6.1 Προσιτότητα (Reachability)

Δεδομένου ενός αρχικού μαρκάριατος M_0 ενός δικτύου Petri, το μαρκάρισμα M είναι προσιτό από το μαρκάρισμα M_0 , εφόσον υπάρχει κάποια ακολουθία πυροδοτήσεων που οδηγεί από το M_0 στο M . Το σύνολο όλων των πιθανών προσιτών μαρκαρισμάτων από το M_0 , ονομάζεται **σύνολο προσιτότητας (reachability set)** και συμβολίζεται με $R(M_0)$. Επίσης, το μαρκάρισμα M' λέγεται ότι είναι άμεσα προσιτό από το M , εφόσον η πυροδότηση μιας ενεργοποιημένης μετάβασης στο μαρκάρισμα M οδηγήσει στο μαρκάρισμα M' .

Το σύνολο προσιτότητας του δικτύου PN3, δεδομένου του αρχικού μαρκάριατος $M_0 = [1\ 0\ 1\ 0\ 0]^T$, όπως φαίνεται στο Σχήμα 7, αποτελείται από τα ακόλουθα τρία μαρκαρίσματα :

$$\text{Σχήμα 8: } M_1 = [0\ 1\ 1\ 0\ 0]^T$$

$$\text{Σχήμα 9: } M_2 = [0\ 0\ 1\ 1\ 0]^T$$

$$\text{Σχήμα 10: } M_3 = [0\ 0\ 1\ 0\ 1]^T$$

Το σύνολο προσιτότητας είναι ένα σημαντικό εργαλείο στη θεωρητική ανάπτυξη των δικτύων Petri και η ανάλυση μέσω προσιτότητας αποτελεί μια μαθηματική μέθοδο ανάλυσης τους.

Το **πρόβλημα της προσιτότητας (reachability problem)** σε ένα δίκτυο Petri σχετίζεται με την διαπίστωση εάν ένα δεδομένο μαρκάρισμα m ανήκει στο σύνολο όλων εκείνων των μαρκαρισμάτων που είναι προσιτά από το αρχικό μαρκάρισμα M_0 . Έχει αποδειχτεί ότι πρόκειται για αποφασίσιμο πρόβλημα, αν και απαιτεί τουλάχιστον εκθετικό χρόνο και χώρο για να απαντηθεί. Στην περίπτωση όμως που δεν εμφανίζονται συγκρούσεις, το πρόβλημα λύνεται σε πολυωνυμικό χρόνο.

Η ιδιότητα της προσιτότητας βοηθά το σχεδιαστή στο να αποφασίσει εάν λαμβάνει χώρα μια συγκεκριμένη διαμόρφωση-κατάσταση σε ένα σύστημα. Πρόκειται για μια πολύ σημαντική ιδιότητα σε **συστήματα κρίσιμα σε ασφάλεια (safety critical systems)**, όταν οι συνθήκες που οδηγούν σε μια δεδομένη κρίσιμη διαμόρφωση δεν πρέπει, για λόγους ασφάλειας, ποτέ να ικανοποιηθούν. Σε συστήματα με χρονικές απαιτήσεις ο έλεγχος της προσιτότητας μπορεί να δείξει καταστάσεις όπου λαμβάνει χώρα παραβίαση χρονικών προθεσμιών.

2.6.2 Περατότητα (Boundedness) και Ασφάλεια (Safety)

Σε ένα δίκτυο Petri με $PN = (P, T, I, O)$, με αρχικό μαρκάρισμα M_0 και με σύνολο προσιτότητας το $R(M_0)$, μια θέση $p \in P$ είναι k -πεπερασμένη, αν $M(p) \leq k$, για κάθε μαρκάρισμα M που ανήκει στο $R(M_0)$, όπου k είναι ένας θετικός ακέραιος. Δηλαδή μια θέση p είναι k -πεπερασμένη, αν ο αριθμός από μάρκες που περιέχει δεν ξεπερνά τον αριθμό k για οποιοδήποτε προσιτό μαρκάρισμα M από το M_0 . Το PN είναι **k -πεπερασμένο (k -bounded)** αν κάθε θέση p του δικτύου είναι k -πεπερασμένη. Η ιδιότητα μπορεί να εξασφαλίσει ότι ο αριθμός των καταστάσεων που μπορεί να βρεθεί ένα σύστημα είναι πεπερασμένος.

Ασφαλές ονομάζεται ένα δίκτυο Petri όταν είναι 1-πεπερασμένο δηλαδή $k = 1$.

Σε πολλές περιπτώσεις δεν μας ενδιαφέρει η ακριβής τιμή του k . Έτσι, μια θέση p ονομάζεται πεπερασμένη αν είναι k -πεπερασμένη για κάποιο k , και το PN ονομάζεται **πεπερασμένο (bounded)** αν όλες οι θέσεις είναι πεπερασμένες. Το PN είναι **δομικά πεπερασμένο (structurally bounded)**, αν είναι πεπερασμένο για κάθε αρχικό μαρκάρισμα M_0 .

Ιδιότητες, όπως η περατότητα και η ασφάλεια ενός δικτύου, καθορίζουν εάν ο αριθμός των μαρκών σε κάθε θέση, είναι πεπερασμένος ή όχι. Ανάλογα λοιπόν με τον τρόπο εφαρμογής ενός μοντέλου δικτύων Petri σε ένα υπολογιστικό σύστημα, οι μάρκες είναι δυνατόν να αποτελούν αφαιρετικές δομές που αντιστοιχούν σε οντότητες όπως οι πόροι, οι διεργασίες κλπ. Επομένως, η ιδιότητα της περατότητας μπορεί να συσχετιστεί με την επαλήθευση πιθανής “**υπερχείλισης**” (“**overflow**”) των οντοτήτων ενός συστήματος, ενώ η ιδιότητα της ασφάλειας εγγυάται ότι δεν θα γίνει καμία προσπάθεια εκκίνησης μιας υπό-εκτέλεσης διεργασίας. Η αρχή της περατότητας ερμηνεύεται ως σταθερότητα ενός συστήματος.

2.6.3 Διάρκεια (Liveness)

Μία μετάβαση t είναι διαρκής αν για ένα οποιαδήποτε μαρκάρισμα $M \in R(M_0)$, υπάρχει μια ακολουθία πυροδοτήσεων, η οποία συνεπάγεται ένα μαρκάρισμα που ενεργοποιεί την t . Με άλλα λόγια, η t είναι διαρκής αν για ένα οποιοδήποτε μαρκάρισμα $M \in R(M_0)$, υπάρχει μια ακολουθία πυροδοτήσεων, η οποία συνεπάγεται ένα μαρκάρισμα που συμπεριλαμβάνει την t . Ένα δίκτυο Petri είναι **διαρκές (live)** αν κάθε μετάβαση σε αυτό είναι διαρκής, δηλαδή αν κάθε μετάβαση μπορεί να πυροδοτήσει, ξεκινώντας από το αρχικό μαρκάρισμα M_0 . Ένα δίκτυο Petri είναι **δομικά διαρκές (structurally live)** αν υπάρχει ένα διαρκές αρχικό μαρκάρισμα M_0 .

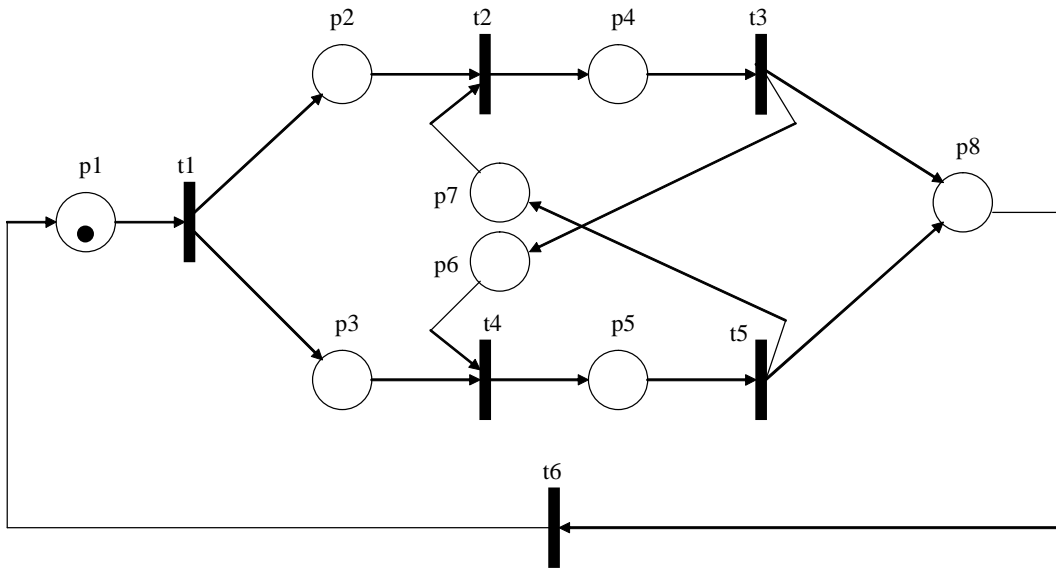
Η ιδιότητα της διάρκειας είναι πολύ περιοριστική και επαληθεύεται δύσκολα. Για το λόγο αυτό ορίζονται διάφορα επίπεδα διάρκειας. Μία μετάβαση t μπορεί να είναι :

- 1) **Νεκρή (dead)**, αν δεν μπορεί να πυροδοτήσει σε καμία ακολουθία πυροδοτήσεων του συνόλου T^* .
- 2) **1-διαρκής (1-live)**, αν μπορεί να πυροδοτήσει τουλάχιστον μία φορά στο σύνολο T^* .
- 3) **2-διαρκής (2-live)**, αν για κάποιον ακέραιο αριθμό K , μπορεί να πυροδοτήσει τουλάχιστον K φορές σε κάποια ακολουθία πυροδοτήσεων του T^* .
- 4) **3-διαρκής (3-live)**, αν πυροδοτεί άπειρες φορές σε κάποια ακολουθία πυροδοτήσεων του συνόλου T^* .

Η ιδιότητα της διάρκειας έχει σχέση με την αποφυγή καταστάσεων όπου παρατηρείται αδιέξοδο λειτουργίας. Ένα δίκτυο Petri περιέχει ένα **αδιέξοδο (deadlock)** αν υπάρχει ένα μαρκάρισμα $M \in R(M_0)$, τέτοιο ώστε μία ή περισσότερες μεταβάσεις να μην ενεργοποιούνται. Ένα τέτοιο μαρκάρισμα ονομάζεται **νεκρό μαρκάρισμα**.

Η διάρκεια ενός δικτύου Petri σημαίνει ότι από ένα οποιοδήποτε μαρκάρισμα M προσιτό από το αρχικό M_0 , είναι απολύτως πιθανό να πυροδοτήσει οποιαδήποτε μετάβαση μέσα στο δίκτυο, ακολουθώντας μία κατάλληλη ακολουθία πυροδοτήσεων. Γι αυτόν τον λόγο, ένα διαρκές δίκτυο Petri, εξασφαλίζει ότι το υπό αναπαράσταση σύστημα δεν εμφανίζει αδιέξοδα λειτουργίας κατά την εκτέλεσή του. Εκτός αν υπάρχει μια διάταξη για εντοπισμό και διόρθωση του αδιεξόδου, μια κατάσταση αδιεξόδου, που συνήθως συμβαίνει σε ένα μικρό υποδίκτυο του δικτύου, μπορεί να διογκωθεί και να επηρεάσει ένα μεγάλο κομμάτι του συστήματος. Αυτό συνήθως συνεπάγεται ολοκληρωτική στασιμότητα του συστήματος.

Στο δίκτυο του Σχήματος 11 υπάρχουν δύο παράλληλες διεργασίες (t_2, t_3) και (t_4, t_5) που αλληλεπιδρούν. Όλο το σύστημα συντονίζεται από τη μετάβαση t_1 .



Σχήμα 11: Παράδειγμα δικτύου Petri με αδιέξοδο

Έστω ότι το παραπάνω δίκτυο Petri έχει αρχικό μαρκάρισμα :

$$M_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]_T$$

Η μετάβαση t_1 είναι ενεργοποιημένη και όταν πυροδοτεί, αφαιρείται η μάρκα της θέσης p_1 και προστίθενται μάρκες στις θέσεις p_2 και p_3 . Το νέο μαρκάρισμα M_1 είναι:

$$M_1 = [0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]_T$$

Εφ'όσον δεν υπάρχουν μάρκες στις θέσεις p_6 και p_7 , και οι δύο μεταβάσεις t_2 και t_3 παρουσιάζουν αδιέξοδο και δεν είναι δυνατή καμία άλλη δραστηριότητα. Αντίθετα, αν το αρχικό μαρκάρισμα του δικτύου είναι το :

$$M_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0]_T$$

ή το συμμετρικό του :

$$M_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]_T,$$

τότε όλες οι μεταβάσεις είναι διαρκείς και δεν υπάρχει αδιέξοδο.

Η ιδιότητα της διάρκειας, όπως προαναφέρθηκε, συνήθως σχετίζεται με την ανίχνευση πιθανών αδιεξόδων λειτουργίας. Ένα τυπικό παράδειγμα κατάστασης αδιεξόδου σε μια εφαρμογή πραγματικού χρόνου συμβαίνει όταν ορισμένες δράσεις της εφαρμογής διακόπτουν τη λειτουργία τους καθώς περιμένουν συνεχώς, προκειμένου να χρησιμοποιήσουν ένα δεσμευμένο πόρο του συστήματος. Μια άλλη περίπτωση αδιεξόδου μπορεί να συμβεί σε μια εφαρμογή που στηρίζεται στην επικοινωνία μέσω ανταλλαγής μηνυμάτων. Οι εφαρμογές αυτές συχνά σχεδιάζονται και υλοποιούνται με βάση το παράδειγμα **πελάτη/εξυπηρετητή (client/server)**. Το αδιέξοδο λαμβάνει χώρα, για παράδειγμα, όταν δεν έχει σταλεί **μήνυμα “κλήσης εισόδου” (“entry call” statement)** προς μια δράση που εκείνη τη στιγμή αναμένει να εκτελέσει μια **εντολή “αποδοχής” (“accept” statement)**. Τότε η δράση που εκτελεί την αποδοχή δεν συνεχίζει τη λειτουργία της, έως ότου διατεθεί το αναμενόμενο μήνυμα. Η διάρκεια είναι μια ιδιότητα που μπορεί να είναι ιδιαιτέρως αυστηρή ώστε να αναπαραστήσει ένα πραγματικό σύστημα ή σενάριο.

2.6.4 Αντιστρεψιμότητα (Reversibility) και Αρχική Κατάσταση (Home State)

Ένα δίκτυο Petri με $PN = (P, T, I, O)$ και αρχικό μαρκάρισμα M_0 καλείται αντιστρέψιμο (**reversible**), εάν για κάθε μαρκάρισμα $M \in R(M_0)$, είναι δυνατό να επιστρέψουμε στο M_0 , δηλαδή όταν το αρχικό μαρκάρισμα είναι πάντοτε προσιτό. Ένα δίκτυο Petri είναι **δομικά αντιστρέψιμο (structurally reversible)**, αν οποιοδήποτε πεπερασμένο αρχικό μαρκάρισμα κάνει το δίκτυο αντιστρεπτό.

Το μαρκάρισμα $M' \in R(M_0)$ ονομάζεται **οικεία κατάσταση (home state)** αν για κάθε μαρκάρισμα $M \in R(M_0)$, το M' είναι προσιτό από το M . Ουσιαστικά, η αντιστρεψιμότητα αποτελεί ειδική περίπτωση της οικείας κατάστασης, αφού αν η ισχύει η οικεία κατάσταση $M' = M_0$, τότε το δίκτυο είναι αντιστρέψιμο.

Η απόδειξη της αντιστρεψιμότητας συνεισφέρει στο να ανιχνευτεί εάν ένα σύστημα επιστρέφει πάντα σε μια ασφαλή κατάσταση, από οποιαδήποτε άλλη κατάσταση εάν βρεθεί. Επειδή πολλά συστήματα απαιτείται να επιστρέφουν από μια κατάσταση αποτυχίας στις επερχόμενες σωστές καταστάσεις, η συγκεκριμένη ιδιότητα είναι πολύ σημαντική για την ανάκαμψη από λάθος ενός συστήματος. Επιπλέον, τα συστήματα πραγματικού χρόνου ανήκουν στη γενικότερη κατηγορία που περιλαμβάνει ταυτόχρονα συστήματα. Η αναπαράσταση ενός ταυτόχρονου συστήματος απαιτεί, για να επιτευχθεί ικανοποιητικά, τη συνεχή παρατήρηση των ταυτόχρονων δράσεων, καθώς και της εκτέλεσής τους, πάντα με γνώμονα το εάν μία δράση επηρεάζει μια άλλη. Τέλος, η ιδιότητα εγγυάται την κυκλική συμπεριφορά του συστήματος, κάτι το οποίο απαιτείται για όλα τα επαναληπτικά συστήματα.

2.6.5 Δικαιοσύνη (Fairness) και Κάλυψη (Coverability)

Ένα δίκτυο Petri με $PN = (P, T, I, O)$, είναι **πεπερασμένα δίκαιο (bounded-fair)**, αν κάθε ζεύγος μεταβάσεων είναι πεπερασμένα δίκαιο, γεγονός που σημαίνει ότι ο μέγιστος αριθμός φορών που οποιαδήποτε από τις δύο μεταβάσεις επιτρέπεται να πυροδοτήσει, ενώ η άλλη δεν πυροδοτεί, είναι πεπερασμένος. Όταν κάθε μετάβαση μπορεί να εμφανίζεται απείρως συχνά σε μια ακολουθία πυροδοτήσεων, τότε το

δίκτυο Petri χαρακτηρίζεται από μια. **χωρίς όρους / καθολική δικαιοσύνη (unconditional / global fairness)**

Η ιδιότητα της δικαιοσύνης των δικτύων Petri μπορεί να συσχετιστεί με την εφαρμογή συγκεκριμένων συνθηκών που ελέγχουν την εξέλιξη (εκτέλεση) ταυτόχρονων δράσεων, και συνεπώς, μπορεί να χρησιμοποιηθεί ώστε να επαληθευτεί αν όλες οι δράσεις της εφαρμογής εν τέλει, λαμβάνουν χώρα και εκτελούνται κατά τη διάρκεια της λειτουργίας του συστήματος.

Ένα μαρκάρισμα M ενός δικτύου Petri με $PN = (P, T, I, O)$ και αρχικό μαρκάρισμα M_0 , καλύπτεται αν υπάρχει άλλο μαρκάρισμα M' , τέτοιο ώστε για κάθε θέση p να ισχύει : $M'(p) \geq M(p)$. Η **κάλυψη** σχετίζεται με τη διάρκεια τύπου 1 (1-live).

2.6.6 Επιμονή (Persistence)

Ένα δίκτυο Petri με $PN = (P, T, I, O)$, ονομάζεται **επίμονο (persistent)** αν για δύο οποιεσδήποτε ενεργοποιημένες μεταβάσεις, η πυροδότηση της μιας δεν καθιστά την άλλη απενεργοποιημένη. Όταν μια μετάβαση ενεργοποιείται, παραμένει ενεργοποιημένη μέχρι να πυροδοτήσει.

Η ιδιότητα της επιμονής, εάν αυτή ερμηνευτεί αντίστροφα, εάν δηλαδή για οποιεσδήποτε δύο ενεργοποιημένες μεταβάσεις, η πυροδότηση της μιας καθιστά την άλλη απενεργοποιημένη, μπορεί να χρησιμοποιηθεί ώστε να ελεγχθεί ότι διατηρείται ο αμοιβαίος αποκλεισμός για τους διαμοιραζόμενους πόρους ενός συστήματος.

2.6.7 Απόσταση Συγχρονισμού (Synchronic Distance)

Σε ένα δίκτυο Petri με $PN = (P, T, I, O)$, η απόσταση συγχρονισμού d_{ij} μεταξύ δύο μεταβάσεων t_i και t_j , με $t_i, t_j \in T$ για κάθε i, j , είναι :

$$d_{ij} = \max | \sigma(t_i) - \sigma(t_j) |,$$

όπου $\sigma(t_i)$, $\sigma(t_j)$ είναι ο αριθμός των φορών που πυροδοτούνται οι μεταβάσεις t_i , t_j αντίστοιχα, στην ακολουθία πυροδοτήσεων σ . Η απόσταση συγχρονισμού είναι δηλαδή, η μέγιστη διαφορά των φορών που δύο μεταβάσεις πυροδοτούν σε μια ακολουθία πυροδοτήσεων. Η απόσταση συγχρονισμού δίνει ένα μέτρο της εξάρτησης μεταξύ των μεταβάσεων.

2.6.8 Συντηρητικότητα (Conservation)

Ένα δίκτυο Petri με $PN = (P, T, I, O)$ και αρχικό μαρκάρισμα M_0 είναι **συντηρητικό (conservative)**, αν για κάθε θέση ισχύει ότι το ζυγισμένο άθροισμα των μαρκών παραμένει σταθερό, ενώ το PN είναι **μερικώς συντηρητικό (partially conservative)** αν αυτό ισχύει για ένα υποσύνολο των θέσεων.

Το PN είναι **αυστηρώς συντηρητικό (strictly conservative)**, αν το βάρος κάθε τόξου είναι ίσο με τη μονάδα, δηλαδή ισχύει $w = 1$. Η αυστηρή συντηρητικότητα

είναι μια εξειδικευμένη περίπτωση της συντηρητικότητας και υποδεικνύει ότι σε ένα δίκτυο Petri, ο αριθμός από μάρκες διατηρείται. Από δομικής πλευράς του δικτύου, η αυστηρή συντηρητικότητα σημαίνει ότι ο αριθμός των τόξων εισόδου σε μία μετάβαση πρέπει να ισούται με τον αριθμό των τόξων εξόδου. Αν αυτή η συνθήκη δεν ικανοποιείται, ο αριθμός των μαρκών πριν την πυροδότηση της μετάβασης θα διαφέρει από τον αριθμό των μαρκών μετά από αυτήν, και η αυστηρή συντηρητικότητα θα παραβιαστεί.

Στις περισσότερες εφαρμογές, η αυστηρή συντηρητικότητα σε όλο το δίκτυο, δεν είναι πολύ χρήσιμη ιδιότητα. Αντίθετα, η αυστηρή συντηρητικότητα στα υποδίκτυα είναι πολύ χρήσιμη, ιδιαίτερα για τη μοντελοποίηση πόρων των οποίων ο αριθμός παραμένει σταθερός, σε συστήματα διεργασιών και σε συστήματα μεταβλητής δομής. Το δίκτυο PN3 του Σχήματος 7 είναι αυστηρά συντηρητικό, όπως φαίνεται από τη μέτρηση του αριθμού των μαρκών στα Σχήματα 7-10.

2.6.9 Συνέπεια (Consistence)

Ένα δίκτυο Petri με $PN = (P, T, I, O)$ και αρχικό μαρκάρισμα M_0 , ονομάζεται **συνεπές (consistent)** αν υπάρχει μια ακολουθία πυροδοτήσεων σ από το M_0 πίσω στο M_0 , τέτοια ώστε κάθε μετάβαση του δικτύου να πυροδοτεί τουλάχιστον μια φορά στην σ . Εάν αυτό ισχύει μόνο για ένα υποσύνολο μεταβάσεων τότε το δίκτυο καλείται **μερικώς συνεπές (partially consistent)**.

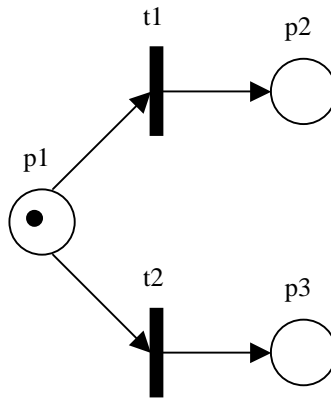
2.7 Μοντελοποίηση με τη Βοήθεια των Δικτύων Petri

Τα δίκτυα Petri χρησιμοποιούνται ευρέως για τη μοντελοποίηση συστημάτων με χαρακτηριστικά σύγχρονης και ταυτόχρονης εκτέλεσης, στα οποία παρουσιάζονται καταστάσεις σύγκρουσης και σύγχυσης.

2.7.1 Σύγκρουση (Conflict)

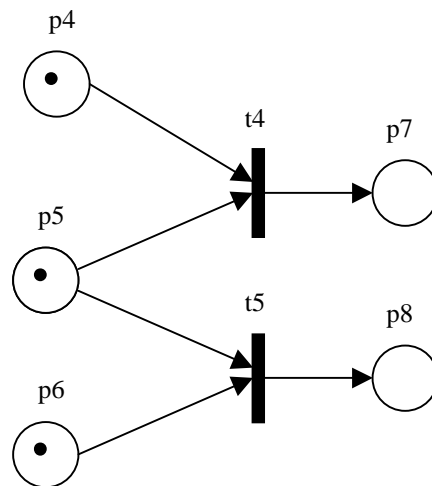
Σε ένα δίκτυο Petri, η παρουσία μιας κοινής θέσης εισόδου μπορεί να δημιουργήσει **σύγκρουση (conflict)**. Το σύστημα πρέπει να είναι σε θέση να επιλύσει τη σύγκρουση αποφασίζοντας μη ντετερμινιστικά ποιά από τις συγκρουόμενες μεταβάσεις θα λάβει χώρα. Ένα δίκτυο Petri είναι **ελεύθερο από συγκρούσεις (conflict-free)**, αν δεν υπάρχει καμία σύγκρουση σε αυτό.

Στο δίκτυο του Σχήματος 12, φαίνεται ότι η θέση p_1 είναι το προσύνολο και των δύο μεταβάσεων t_1 και t_2 . Και οι δύο μεταβάσεις ενεργοποιούνται από την παρουσία της μάρκας στη θέση p_1 . Αν η μετάβαση t_1 πυροδοτήσει, τότε η t_2 δεν είναι πλέον ενεργοποιημένη. Αντίστροφα αν η t_2 πυροδοτήσει, η t_1 δεν είναι πλέον ενεργοποιημένη.



Σχήμα 12: Παράδειγμα δικτύου Petri με σύγκρουση

Μια λίγο πιο περίπλοκη περίπτωση φαίνεται στο Σχήμα 13. Εδώ η θέση p5 ανήκει στα προσύνολα και της μετάβασης t4 και της μετάβασης t5. Σε αυτό το παράδειγμα, η σύγκρουση συμβαίνει γιατί και οι δύο θέσεις, p4 και p6, περιέχουν μάρκες.



Σχήμα 13: Ένα άλλο παράδειγμα δικτύου Petri με σύγκρουση

Γενικά, δεν είναι επιθυμητό να υπάρχει σύγκρουση σε ένα μοντέλο, γιατί η σύγκρουση υποδεικνύει ότι το μοντέλο είναι διαφορούμενο. Υπάρχουν δύο επιλογές :

- I. εισαγωγή ενός κανόνα ο οποίος θα λύνει τη σύγκρουση, υποδεικνύοντας ποια από τις ταυτόχρονα ενεργοποιημένες μεταβάσεις θα πυροδοτήσει πρώτη, ή
- II. αλλαγή της δομής του συστήματος – επανασχεδίαση – για να απαλειφθεί η σύγκρουση.

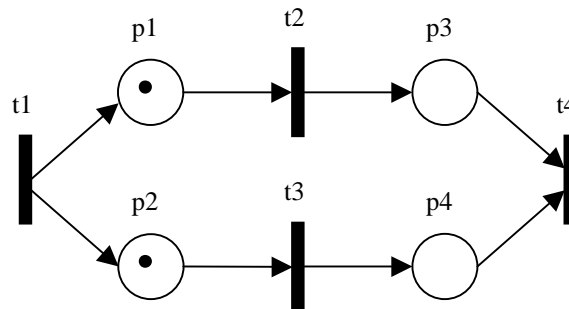
2.7.2 Ταυτοχρονισμός (Concurrency) και Συγχρονισμός (Synchronisation)

Σε ένα δίκτυο Petri, δύο ή περισσότερες μεταβάσεις ονομάζονται **ταυτόχρονες (concurrent)**, όταν δεν έχουν σχέση αιτιότητας μεταξύ τους και μπορούν να

εκτελεστούν με οποιαδήποτε σειρά, χωρίς η εκτέλεση της μιας να αποτρέπει την εκτέλεση της άλλης. Δηλαδή, οι ταυτόχρονες μεταβάσεις πυροδοτούν, είτε σειριακά η μία μετά την άλλη είτε παράλληλα, οπότε και ονομάζονται **παράλληλες μεταβάσεις (parallel transitions)**.

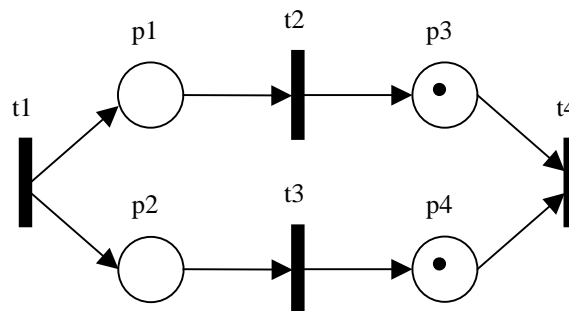
Μέσω των ταυτόχρονων μεταβάσεων, τα δίκτυα Petri μπορούν να απεικονίσουν τόσο παράλληλη εκτέλεση προγραμμάτων όσο και σειριακή εκτέλεση κι έτσι χρησιμοποιούνται ως μοντέλα τόσο παραλλήλων όσο και σειριακών συστημάτων.

Στο δίκτυο Petri του Σχήματος 14, οι μεταβάσεις t2 και t3, είναι ταυτόχρονες.



Σχήμα 14: Παράδειγμα δικτύου Petri με ταυτόχρονες μεταβάσεις

Επίσης, σε ένα δίκτυο Petri μπορεί να υπάρχει **συγχρονισμός (synchronisation)**. Για παράδειγμα το δίκτυο Petri του Σχήματος 15, που είναι το δίκτυο του Σχήματος 14 μετά την πυροδότηση των ταυτόχρονων μεταβάσεων t2 και t3. Σε αυτό το παράδειγμα η μετάβαση t4 μπορεί να πυροδοτηθεί μόνο αν ικανοποιούνται οι συνθήκες, που σχετίζονται και με τις δύο θέσεις, p3 και p4.

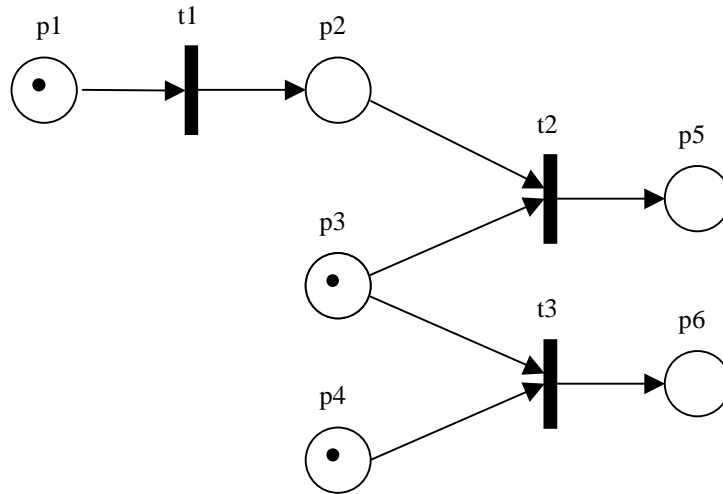


Σχήμα 15: Δίκτυο Petri με συγχρονισμό

2.7.3 Σύγχυση (Confusion)

Ενώ η σύγκρουση είναι ένα τοπικό φαινόμενο, με την έννοια ότι μόνο τα προσύνολα των μεταβάσεων με κοινές θέσεις εισόδου εμπλέκονται, η **σύγχυση (confusion)** εμπλέκει τις ακολουθίες πυροδοτήσεων. Η σύγχυση συμβαίνει όταν, παρουσία ταυτοχρονισμού, μία ακολουθία πυροδοτήσεων οδηγεί σε σύγκρουση, ενώ μια άλλη όχι. Το Σχήμα 16 απεικονίζει μια τέτοια περίπτωση. Οι μεταβάσεις t1 και t3 είναι ενεργοποιημένες ταυτόχρονα. Σε αυτό το σημείο, δεν υπάρχει σύγκρουση. Αν η

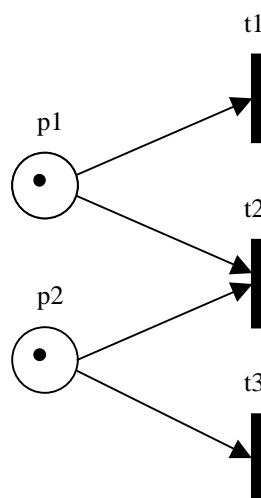
μετάβαση t_3 πυροδοτήσει πρώτη, τότε η μετάβαση t_1 πυροδοτεί και η εκτέλεση τερματίζεται, αφού καμία άλλη μετάβαση δεν ενεργοποιείται. Αλλά, αν η μετάβαση t_1 πυροδοτήσει πρώτη, τότε η παρουσία μάρκας στη θέση p_3 ενεργοποιεί τη μετάβαση t_2 κι έχει ως αποτέλεσμα τη δημιουργία σύγκρουσης, όπως φαίνεται και στο Σχήμα 13.



Σχήμα 16: Παράδειγμα δικτύου Petri με σύγκυση (ασυμμετρική)

Στο παραπάνω δίκτυο Petri, η σύγκυση ονομάζεται ασυμμετρική. **Ασυμμετρική σύγκυση (asymmetric confusion)** παρουσιάζεται όταν μια μετάβαση (t_3) είναι παράλληλη με μια άλλη μετάβαση (t_1) και θα συγκρουστεί με μια τρίτη μετάβαση (t_2), αν η παράλληλη της (t_1) πυροδοτήσει πρώτη.

Το δίκτυο Petri του Σχήματος 17 έχει **συμμετρική σύγκυση (symmetric confusion)**, η οποία παρουσιάζεται όταν δύο ή περισσότερες μεταβάσεις (t_1, t_3) είναι παράλληλες μεταξύ τους και σε σύγκρουση με μια άλλη (t_2).



Σχήμα 17: Παράδειγμα δικτύου Petri με συμμετρική σύγκυση

2.8 Επεκτάσεις των Δικτύων Petri

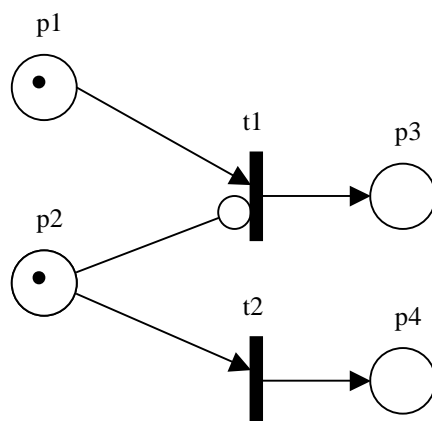
Μια σημαντική επέκταση για την αύξηση της ισχύος μοντελοποίησης των δικτύων Petri επιτυγχάνεται με την προσθήκη της **δυνατότητας μηδενικού ελέγχου (zero testing ability)**, δηλαδή της δυνατότητας ελέγχου, αν μια θέση έχει μηδενικό αριθμό από μάρκες. Ένα **τόξο αποτροπής (inhibitor arc)** χρησιμοποιείται για να υλοποιηθεί αυτή η δυνατότητα. Το τόξο αποτροπής συνδέει μια θέση εισόδου με μια μετάβαση, και σχεδιάζεται με ένα τόξο το οποίο στην αρχή του έχει έναν μικρό κύκλο, και αυτό διότι στα ψηφιακά κυκλώματα ο μικρός κύκλος υποδηλώνει το NOT. Ένα δίκτυο Petri με τόξα αποτροπής παρουσιάζει αυξημένη δυνατότητα έκφρασης (υπολογιστική δύναμη), ίση με εκείνη μιας **Μηχανής Turing (Turing Machine)**.

Η παρουσία του τόξου αποτροπής που συνδέει μια θέση εισόδου με μια μετάβαση αλλάζει τις συνθήκες ενεργοποίησης της μετάβασης. Λόγω της παρουσίας του, μια μετάβαση θεωρείται ως ενεργοποιημένη, όταν :

- i. κάθε θέση εισόδου που είναι συνδεδεμένη με την μετάβαση με ένα κανονικό τόξο περιέχει αριθμό μαρκών τουλάχιστον ίσο με το βάρος του τόξου και
- ii. καμία μάρκα δεν είναι παρόν στις θέσεις εισόδου που συνδέονται με τόξα αποτροπής .

Οι κανόνες ενεργοποίησης της μετάβασης είναι ίδιοι με αυτούς για τις συνδεδεμένες με κανονικά τόξα μεταβάσεις. Η ενεργοποίηση όμως δεν αλλάζει το μαρκάρισμα στις θέσεις που είναι συνδεδεμένες με τόξα αποτροπής. Το τόξο αποτροπής μπορεί να γενικευθεί με μια πολλαπλότητα. Έτσι, όταν ο αριθμός των μαρκών στις θέσεις είναι μικρότερος από τον αριθμό πολλαπλότητας του τόξου αποτροπής, η μετάβαση επιτρέπεται να ενεργοποιηθεί από τις κανονικές θέσεις εισόδου.

Στο δίκτυο Petri του Σχήματος 18, ένα τόξο αποτροπής συνδέει τη θέση p2 με τη μετάβαση t1. Σε πρώτη φάση η μετάβαση t1 δεν ενεργοποιείται γιατί υπάρχει μία μάρκα στη θέση p2. Οπότε πυροδοτεί πρώτα, η μετάβαση t2 που είναι ενεργοποιημένη. Μετα την πυροδότηση της μετάβασης t2, η μάρκα στη θέση p2 δεν υπάρχει πλέον κι έτσι η μετάβαση t1 τώρα μπορεί να ενεργοποιηθεί.



Σχήμα 18: Παράδειγμα δικτύου Petri με τόξο αποτροπής

Όσον αφορά τον ορισμό του δικτύου Petri, το τόξο αποτροπής προστίθεται μέσω μιας συνάρτησης αποτροπής $H : P \times T \rightarrow N$. Μία μετάβαση $t \in T$ θεωρείται ενεργοποιημένη όταν και μόνο όταν :

$$m(p) \geq I(p, t) \text{ και } m(p) < H(p, t), \quad p \in P.$$

Πρέπει ακόμα να σημειωθεί, ότι όταν ισχύει $H(p, t) = 1$, τότε $I(p, t) = 0$, αφού εάν δεν ήταν έτσι τότε η t δεν θα ενεργοποιείτο ποτέ, σύμφωνα με τον προαναφερθέν κανόνα. Στις περιπτώσεις όπου ισχύει $H(p, t) > 1$, ισχύει $I(p, t) < H(p, t)$ για οποιοδήποτε κανονικό τόξο υπάρχει. Θεωρητικά, ένα τόξο αποτροπής γίνεται απαραίτητο όταν υπάρχουν θέσεις που ενδεχομένως αποκτούν άπειρο αριθμό από μάρκες και το σύστημα πρέπει να ξέρει τον αριθμό των μαρκών σε αυτές τις θέσεις.

Ένας άλλος τρόπος παράστασης του τόξου αποτροπής είναι ο κλειστός βρόγχος, ο οποίος, όπως αναφέρθηκε και προηγουμένως, είναι ένα ζευγάρι θέσεων όπου ισχύει $I(p, t) = O(p, t)$, και η θέση p έχει τουλάχιστον $I(p, t)$ μάρκες. Τόσο ο κλειστός βρόγχος όσο και το τόξο αποτροπής αναπαριστούν ακριβώς την ίδια κατάσταση συστημάτων.

Άλλες μέθοδοι βελτίωσης και εμπλουτισμού της ισχύος των δικτύων Petri περιλαμβάνουν :

- 1) την ανάθεση **προτεραιοτήτων (priorities)** μεταξύ των μεταβάσεων
- 2) την εισαγωγή χρονικής καθυστέρησης σε θέσεις, μεταβάσεις, ή/και σε τόξα.

Η προσθήκη προτεραιοτήτων μπορεί να έχει το ίδιο αποτέλεσμα με το τόξο αποτροπής. Στο δίκτυο Petri του Σχήματος 19, η ανάθεση προτεραιότητας στη μετάβαση t_2 , έχει ως αποτέλεσμα να πυροδοτήσει πρώτη, πριν τη μετάβαση t_1 .

2.9 Μέθοδοι Ανάλυσης

Σε ό,τι αφορά τις μεθόδους ανάλυσης που συνοδεύουν τον τυπικό φορμαλισμό των δικτύων Petri, μπορεί να εφαρμοστεί η **προσομοίωση (simulation)** και η παρατήρηση της δυναμικής συμπεριφοράς του μοντέλου. Η εφαρμογή της προσομοίωσης μπορεί να συντελέσει στη μελέτη και στην ανίχνευση εσφαλμένης συμπεριφοράς. Γενικά όμως, ως μέθοδος, η προσομοίωση δεν είναι πάντοτε αποτελεσματική στο να βοηθήσει στην πλήρη επαλήθευση και απόδειξη των ιδιοτήτων ενός συστήματος, είτε πρόκειται για ιδιότητες συμπεριφοράς (δυναμικές ιδιότητες), είτε για ιδιότητες δομής (στατικές ιδιότητες). Αποτελεί όμως αποδοτική λύση όταν παρατηρείται αυξημένη πολυπλοκότητα του χώρου καταστάσεων, δηλαδή μεγάλος αριθμός πιθανών μαρκαρισμάτων του δικτύου. Επιπρόσθετα, έχουν αναπτυχθεί τυπικές τεχνικές, που έχουν ως κύριο στόχο την ανάλυση και επαλήθευση της ορθότητας των προδιαγραφών ενός συστήματος που περιγράφεται με όρους των δικτύων Petri. Οι τεχνικές αυτές είναι:

- i. η μέθοδος του **δένδρου κάλυψης - προσιτότητας (coverability - reachability tree)**,
- ii. η μέθοδος με τη βοήθεια **αμετάβλητων διανυσμάτων (invariants)**,
- iii. η κατάταξη ενός δοθέντος δικτύου σε μια συγκεκριμένη κατηγορία δικτύων,

- iv. οι μέθοδοι **απλοποίησης (reduction)** και **σύνθεσης (synthesis)**.

2.9.1 Μέθοδος Ανάλυσης μέσω Προσιτότητας

Η μέθοδος ανάλυσης μέσω προσιτότητας σχετίζεται με τη συστηματική **απαρίθμηση (enumeration)** όλων των μαρκαρισμάτων που είναι προσιτά από το αρχικό μαρκάρισμα M_0 και τα οποία αποτελούν το σύνολο προσιτότητας του δικτύου. Η μέθοδος καταγράφει κάθε μαρκάρισμα που προκύπτει από το αρχικό, μέσω μιας ακολουθίας πυροδοτήσεων μεταβάσεων, και τελικά καταλήγει σε μια κατάλληλη δενδρική αναπαράσταση όλων αυτών των προσιτών μαρκαρισμάτων. Η αναπαράσταση αυτή ονομάζεται **δένδρο κάλυψης (coverability tree)**. Κάθε κόμβος του δένδρου αναπαριστά ένα προσιτό μαρκάρισμα, ενώ η ρίζα του δένδρου αναπαριστά το αρχικό μαρκάρισμα M_0 . Κάθε κατευθυνόμενη ακμή (βέλος) στο δένδρο κάλυψης αναπαριστά την πυροδότηση της μετάβασης, που “μετασχηματίζει” ένα μαρκάρισμα σε ένα άλλο. Στην περίπτωση των πεπερασμένων (bounded) δικτύων - δηλαδή όταν, σε κάθε θέση του δικτύου, ο αριθμός των μαρκών δεν είναι μεγαλύτερος από ένα πεπερασμένο φυσικό αριθμό - το δένδρο κάλυψης καλείται **δένδρο προσιτότητας (reachability tree)**. Επίσης, όταν το παραγόμενο δένδρο κάλυψης ή προσιτότητας περιέχει βέλη προς τα πίσω, τότε αντιστοιχεί σε μια δομή γραφήματος που καλείται **γράφος κάλυψης-προσιτότητας (coverability - reachability graph)**.

Ο γράφος προσιτότητας μπορεί να δημιουργηθεί με δύο στρατηγικές : κατά βάθος και κατά πλάτος. Και οι δύο στρατηγικές οδηγούν στον ίδιο γράφο προσιτότητας. Στην πραγματικότητα, κάθε δίκτυο έχει μοναδικό γράφο προσιτότητας.

Στην κατά βάθος στρατηγική, ξεκινώντας από το αρχικό μαρκάρισμα M_0 , αναγνωρίζουμε όλες τις μεταβάσεις που μπορούν να πυροδοτήσουν, και πυροδοτούμε μία στην τύχη, προκαλώντας ένα νέο μαρκάρισμα. Εάν το μαρκάρισμα αυτό είναι νεκρό ή “παλιό”, δηλαδή έχει ήδη δημιουργηθεί, τότε το αγνοούμε, επιστρέφουμε στο μαρκάρισμα από το οποίο προέκυψε, και συνεχίζουμε με κάποια άλλη μετάβαση. Εάν το μαρκάρισμα δεν είναι “παλιό” ή νεκρό, αναγνωρίζουμε ξανά τις μεταβάσεις που μπορούν να πυροδοτήσουν, και πυροδοτούμε μία στην τύχη, προκαλώντας ξανά ένα νέο μαρκάρισμα. Αυτή η διαδικασία συνεχίζεται έως ότου όλες οι ενεργοποιημένες μεταβάσεις έχουν πυροδοτήσει και όλα τα μαρκάρια έχουν δημιουργηθεί, εφόσον ο αριθμός των μαρκαρισμάτων είναι πεπερασμένος.

Στην κατά πλάτος στρατηγική, αναγνωρίζουμε όλες τις μεταβάσεις που μπορούν να πυροδοτήσουν, και τις πυροδοτούμε όλες, προκαλώντας νέα μαρκάρια. Για κάθε μαρκάρισμα, εάν είναι “παλιό” ή νεκρό, προχωράμε στο επόμενο. Σε διαφορετική περίπτωση, αναγνωρίζουμε όλες τις μεταβάσεις που μπορούν να πυροδοτήσουν και τις πυροδοτούμε, προκαλώντας νέα μαρκάρια, και προχωράμε στο επόμενο μαρκάρισμα, μέχρις ότου στα μαρκάρια του ίδιου επιπέδου να εξαντληθούν. Στη συνέχεια, αρχίζουμε από το επόμενο επίπεδο μαρκαρισμάτων.

Όταν ένα δίκτυο Petri είναι μη-πεπερασμένο, τότε η παραπάνω διαδικασία θα συνεχίζεται για πάντα. Για να διατηρηθεί το δένδρο πεπερασμένο, εισάγουμε το σύμβολο ω , το οποίο συμβολίζει το άπειρο και ικανοποιεί τις σχέσεις :

$$\omega > k, \omega \leq \omega \text{ και } \omega \pm k = \omega,$$

για οποιονδήποτε πεπερασμένο ακέραιο k

Από το γράφο προσιτότητας μπορούν να εξαχθούν τα παρακάτω συμπεράσματα :

- i. Το δίκτυο είναι φραγμένο, αν και μόνο αν το ω δεν εμφανίζεται σε κανέναν κόμβο.
- ii. Το δίκτυο είναι ασφαλές, αν και μόνο αν κάθε κόμβος του δένδρου περιέχει μόνο μηδενικά και άσους.
- iii. Μία μετάβαση θεωρείται νεκρή εάν δεν εμφανίζεται σε κανένα τόξο στο δένδρο.
- iv. Εάν πάρουμε δύο κόμβους του δένδρου που δεν περιέχουν το σύμβολο ω , και στο μονοπάτι που τους συνδέει εμφανίζονται όλες οι μεταβάσεις του δικτύου, τότε το δίκτυο είναι ζωντανό.
- v. Εάν υπάρχει ένα μονοπάτι από έναν οποιονδήποτε κόμβο προς το αρχικό μαρκάρισμα χωρίς το σύμβολο ω , τότε το δίκτυο είναι αντιστρέψιμο.

Πρέπει όμως να σημειωθεί ότι η μέθοδος αποτελεί μια συστηματική τεχνική απαρίθμησης που οδηγεί στην καταγραφή της “ακολουθιακής” συμπεριφοράς ενός δικτύου και γενικά καλύπτει την ανάλυση των ιδιοτήτων συμπεριφοράς. Συχνά η εφαρμογή της μεθόδου περιορίζεται σε πεπερασμένα, απλά δίκτυα. Αυτό συμβαίνει διότι το μέγεθος του παραγόμενου τελικά δένδρου-γραφήματος συνήθως αυξάνει εκθετικά όταν αυξάνει το μέγεθος και η πολυπλοκότητα του αντίστοιχου δικτύου. Στην περίπτωση αυτή λαμβάνει χώρα το λεγόμενο πρόβλημα της **έκρηξης του χώρου καταστάσεων (state space explosion problem)**.

2.9.2 Μέθοδος Ανάλυσης μέσω Αμετάβλητων Διανυσμάτων

Σύμφωνα με τη δεύτερη μέθοδο ανάλυσης, κάθε μαρκάρισμα αναπαρίσταται ως ένα **διάνυσμα (vector)**, με αποτέλεσμα ένα δίκτυο να μπορεί να μετασχηματιστεί σε ένα σύνολο από γραμμικές εξισώσεις. Η μέθοδος είναι ιδιαίτερα κατάλληλη για την επαλήθευση των δομικών ιδιοτήτων ενός δικτύου. Συγκεκριμένα, πρωταρχικός στόχος είναι ο υπολογισμός των **αμετάβλητων διανυσμάτων θέσεων και μεταβάσεων (place and transition invariants)**. Ως αμετάβλητο διάνυσμα θέσεων ορίζεται εκείνο το διάνυσμα στο οποίο κάθε θέση του δικτύου αντιστοιχεί σε ένα συγκεκριμένο αριθμό μαρκών, αριθμός που παραμένει πάντοτε σταθερός κατά την εκτέλεση του δικτύου (δηλαδή κατά την πυροδότηση των μεταβάσεων του δικτύου). Ανάλογα, κάθε στοιχείο ενός αμετάβλητου διανύσματος μεταβάσεων αντιστοιχεί στον αριθμό πυροδοτήσεων μιας μετάβασης σε μια ακολουθία πυροδοτήσεων που οδηγεί ξανά στο αρχικό μαρκάρισμα. Ο υπολογισμός των αμετάβλητων διανυσμάτων συχνά αποδεικνύεται μια εξαιρετικά επίπονη διαδικασία και στην περίπτωση πολύπλοκων και μεγάλων σε έκταση δικτύων το πλήθος των αμετάβλητων διανυσμάτων μπορεί να αυξηθεί σημαντικά. Επίσης, δύσκολα μπορεί να διαπιστωθεί ποια από αυτά τα διανύσματα είναι τελικά χρήσιμα, προκειμένου να εξαχθούν κατάλληλα συμπεράσματα για τις δομικές ιδιότητες του δικτύου. Δύο κύριοι παράγοντες είναι υπεύθυνοι γι' αυτόν τον περιορισμό: η μη ντετερμινιστική φύση του μοντέλου (ασθενής κανόνας πυροδότησης) και η απαίτηση όλες οι πιθανές λύσεις του συνόλου των γραμμικών εξισώσεων να είναι μη αρνητικοί ακέραιοι αριθμοί.

2.9.3 Μέθοδος Ανάλυσης βάσει της Κατηγορίας του Δικτύου Petri

Η επαλήθευση των ιδιοτήτων ενός συστήματος μπορεί συχνά να διευκολυνθεί μέσω μιας διαδικασίας ανάλυσης πάνω στο αντίστοιχο κατευθυνόμενο γράφημα. Μια διαδικασία αυτού του είδους επικεντρώνεται στο να ανιχνεύσει σε ποια συγκεκριμένη κατηγορία δικτύων Petri μπορεί να ανήκει ένα δεδομένο δίκτυο. Ως επακόλουθο, τα χαρακτηριστικά που εμφανίζουν γενικά τα δίκτυα της κάθε κατηγορίας μπορούν να χρησιμοποιηθούν για την ανάλυση των ιδιοτήτων. Στο επόμενο κεφάλαιο θα αναλυθούν τα χαρακτηριστικά κάθε κατηγορίας δικτύων Petri.

2.9.4 Μέθοδοι Ανάλυσης μέσω Απλοποίησης και Σύνθεσης

Το πιο σημαντικό μειονέκτημα όλων των προαναφερθέντων μεθόδων είναι ότι η εφαρμογή τους συχνά αποδεικνύεται μη πρακτική και επίπονη κατά την ανάλυση πολύπλοκων, ρεαλιστικών συστημάτων. Λύση στο πρόβλημα αυτό προσπαθούν να δώσουν οι τεχνικές **απλοποίησης (reduction)** και **σύνθεσης (synthesis)**. Μια μέθοδος απλοποίησης στοχεύει στην αναγωγή ενός σύνθετου μοντέλου σε ένα απλούστερο, διατηρώντας παράλληλα τις ιδιότητες του αρχικού μοντέλου. Μια μέθοδος σύνθεσης έχει ως στόχο τη συστηματική δόμηση δικτύων, έχοντας πάντοτε ως γνώμονα την εξασφάλιση των απαιτούμενων ιδιοτήτων και αποφεύγοντας, όσο είναι δυνατόν, την ανάλυση στα τελικώς εξαγόμενα σύνθετα δίκτυα. Διακρίνουμε δύο τύπους σύνθεσης : αυτή που πραγματοποιείται με **κατεύθυνση από πάνω προς τα κάτω (top-down)** και εκείνη που πραγματοποιείται με **κατεύθυνση από κάτω προς τα πάνω (bottom-up)**. Σύμφωνα με την πρώτη μέθοδο, οι μεταβάσεις και οι θέσεις του δικτύου μπορούν να **εκλεπτυνθούν (refined)** σε πιο αναλυτικά και πλήρη δίκτυα. Σύμφωνα με τη δεύτερη μέθοδο, το τελικό δίκτυο δομείται μέσω της **σύμπτυξης (fusion)** κοινών θέσεων, κοινών μεταβάσεων ή ακόμη κοινών **μονοπατιών (paths)** ανάμεσα σε δύο ή περισσότερα τμήματα (υποδίκτυα) ενός δικτύου.

ΚΕΦΑΛΑΙΟ 3^ο

Κατηγορίες Δικτύων Petri

3.1 Δίκτυα Petri Χαμηλού Επιπέδου

Η εισαγωγή δομικών περιορισμών στο γενικό μοντέλο των δικτύων Petri που περιγράψαμε στο προηγούμενο κεφάλαιο οδηγεί στον ορισμό μιας σειράς κατηγοριών για τα **δίκτυα Petri χαμηλού επιπέδου (low-level Petri nets)**. Η διαδικασία της απάντησης στο ερώτημα εάν ένα δίκτυο ανήκει ή όχι σε μια συγκεκριμένη δομική κατηγορία, σε αρκετές περιπτώσεις, αποδεικνύεται εξαιρετικά χρήσιμη, καθώς οι ιδιότητες που παρουσιάζουν τα δίκτυα κάθε κατηγορίας χρησιμοποιούνται για να διευκολύνουν την ανάλυση. Εάν λοιπόν κύριος στόχος του σχεδιαστή είναι η εφαρμογή της ανάλυσης, η επιλογή ενός δικτύου χαμηλού επιπέδου μπορεί να αποδειχτεί εξαιρετικά χρήσιμη. Όμως, στην περίπτωση μοντελοποίησης πολύπλοκων ρεαλιστικών συστημάτων, υπάρχει πάντοτε η πιθανότητα, εάν επιλεγεί ένα μοντέλο χαμηλού επιπέδου, να προκύψουν σύνθετα δίκτυα. Για το λόγο αυτό στη βιβλιογραφία έχουν προταθεί τεχνικές σύνθεσης και αποσύνθεσης, ώστε να διευκολύνεται, σε ένα βαθμό, η συστηματική δόμηση δικτύων, με βάση συνιστώσες (τμήματα δικτύων) που ανήκουν σε μια συγκεκριμένη κάθε φορά κατηγορία, όπου μπορούν εύκολα να εφαρμοστούν μέθοδοι ανάλυσης. Οι κατηγορίες των δικτύων Petri χαμηλού επιπέδου είναι οι εξής :

1) Μηχανές Κατάστασης (State Machines)

Μια μηχανή κατάστασης είναι ένα δίκτυο Petri με την ιδιότητα ότι κάθε μετάβαση έχει ακριβώς μία θέση εισόδου και ακριβώς μία θέση εξόδου. Δηλαδή κάθε μετάβαση έχει μόνο ένα εισερχόμενο και μόνο ένα εξερχόμενο τόξο. Οι μηχανές κατάστασης επιτρέπουν την αναπαράσταση των συγκρούσεων αλλά δεν μπορούν να περιγράψουν συγχρονισμό ανάμεσα σε ταυτόχρονες ενέργειες. Σχήμα 1(α)

2) Μαρκκαρισμένοι Γράφοι (Marked Graphs)

Ένας μαρκκαρισμένος γράφος είναι ένα δίκτυο Petri με την ιδιότητα ότι κάθε θέση έχει ακριβώς μία μετάβαση εισόδου και ακριβώς μία μετάβαση εξόδου. Δηλαδή κάθε θέση έχει μόνο ένα εισερχόμενο και μόνο ένα εξερχόμενο τόξο. Όλοι οι μαρκκαρισμένοι γράφοι είναι επίμονοι κι επιτρέπουν την αναπαράσταση ταυτόχρονων ενεργειών, αλλά δεν είναι δυνατόν να περιγράψουν δομές συγκρούσεων. Επομένως, οι μαρκκαρισμένοι γράφοι κρίνονται κατάλληλοι για τη μοντελοποίηση ταυτόχρονων συστημάτων που είναι **ελεύθερα αποφάσεων (decision-free)**. Σχήμα 1(β)

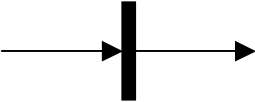
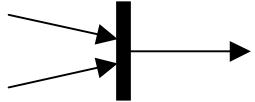
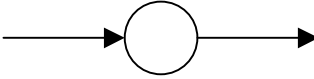
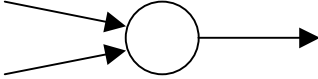
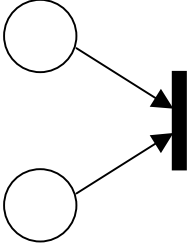
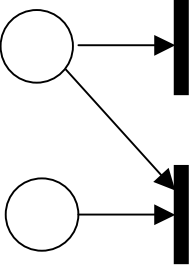
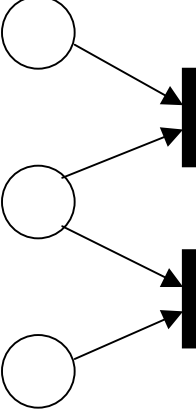
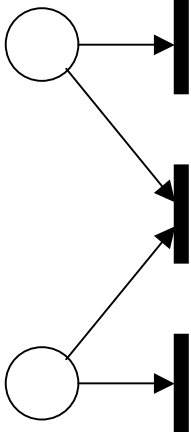
3) Δίκτυα Petri Ελεύθερης Επιλογής (Free Choice Nets)

Ένα δίκτυο Petri ελεύθερης επιλογής έχει την ιδιότητα ότι στην τελική σχέση κάθε θέση με κάθε μετάβαση συνδέονται ως εξής : είτε η μετάβαση είναι η μοναδική μετάβαση εξόδου αυτής της θέσης είτε η θέση είναι η μοναδική

θέση εισόδου αυτής της μετάβασης. Δηλαδή κάθε τόξο, είτε είναι το μοναδικό εξερχόμενο τόξο μιας θέσης, είτε είναι το μοναδικό εισερχόμενο τόξο μιας μετάβασης. Αυτά τα δίκτυα δεν αναπαριστούν καταστάσεις σύγκυσης. Σχήμα 1(γ)

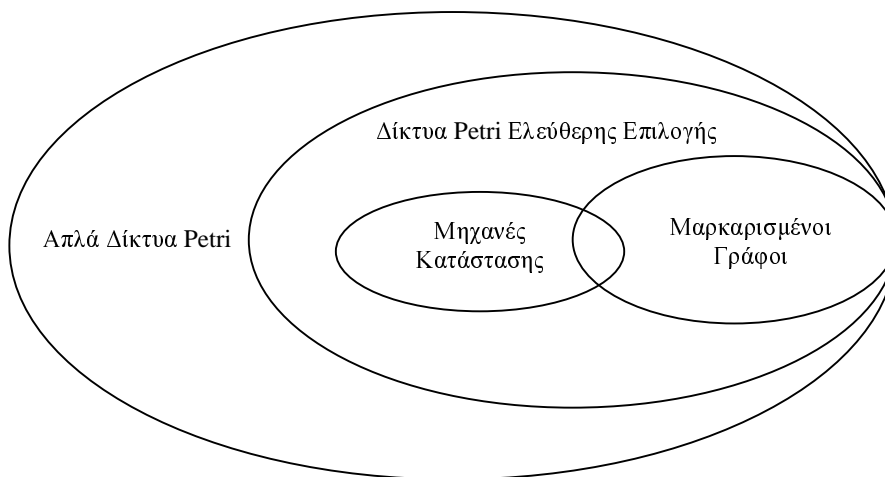
4) Απλά Δίκτυα Petri (Simple Nets)

Ένα απλό δίκτυο Petri είναι ένα δίκτυο Petri με την ιδιότητα ότι όλες οι μεταβάσεις έχουν το πολύ μία θέση εισόδου που οδηγεί σε άλλες μεταβάσεις. Δηλαδή, δεν υπάρχουν στο δίκτυο, δύο μεταβάσεις οι οποίες να έχουν τις ίδιες θέσεις εισόδου και ταυτόχρονα τις ίδιες θέσεις εξόδου. Σχήμα 1(δ)

Κατηγορίες Δικτύων Petri	Επιτρεπτή Μορφή	Μη Επιτρεπτή Μορφή
Μηχανές Κατάστασης (State Machines)		
Μαρκαρισμένοι Γράφοι (Marked Graphs)		
Δίκτυα Petri Ελεύθερης Επιλογής (Free Choice Nets)		
Απλά Δίκτυα Petri (Simple Nets)		

Σχήμα 1: Κατηγορίες Δικτύων Petri

Στο Σχήμα 2 φαίνονται οι σχέσεις μεταξύ των παραπάνω κατηγοριών δικτύων Petri. Οι μηχανές κατάστασης και οι μαρκαρισμένοι γράφοι είναι υποσύνολα των δικτύων Petri ελεύθερης επιλογής, τα οποία με τη σειρά τους είναι υποσύνολα των απλών δικτύων Petri.



Σχήμα 2: Οι σχέσεις μεταξύ των κατηγοριών των δικτύων Petri

3.2 Δίκτυα Petri Υψηλού Επιπέδου

Ο βασικός περιορισμός όλων των μοντέλων δικτύων Petri χαμηλού επιπέδου συνδέεται με το γεγονός ότι απαιτούνται συνήθως πολύπλοκα δίκτυα για την περιγραφή εφαρμογών μεσαίου βαθμού πολυπλοκότητας. Έτσι, γίνεται δύσκολη, τόσο η διαδικασία υπολογισμού των αμετάβλητων διανυσμάτων όσο και η ανάλυση του δένδρου προσιτότητας, αφού σε μη πεπερασμένα δίκτυα παρουσιάζεται το λεγόμενο **πρόβλημα της έκρηξης του χώρου καταστάσεων (state explosion problem)**. Επίσης, τα χαμηλού επιπέδου μοντέλα δεν υποστηρίζουν επαρκώς το σαφή προσδιορισμό των **διακεκριμένων οντοτήτων ενός συστήματος (individuals)**, των ιδιοτήτων και των συσχετισμών μεταξύ τους. Τέλος, ο φορμαλισμός των περισσότερων μοντέλων δεν υποστηρίζει ρητά τον προσδιορισμό συγκεκριμένων μηχανισμών δόμησης, όπως είναι οι τελεστές σύνθεσης.

Για όλους τους παραπάνω λόγους, έχουν προταθεί γενικευμένα μοντέλα που είναι γνωστά με τον όρο **υψηλού επιπέδου δίκτυα Petri (high-level Petri nets / HPNs)** ή, διαφορετικά, **μοντέλα βασισμένα στα δίκτυα Petri (PN-based models)**. Τα μοντέλα αυτά προσφέρουν τόσο περιεκτικότητα, όσο και ευχρηστία στην τελική αναπαράσταση ενός συστήματος, ενώ ταυτόχρονα μπορούν να χρησιμοποιηθούν για να περιγράψουν με σαφήνεια τη ροή δεδομένων και ελέγχου, σύνθετες συνθήκες και ενέργειες πυροδότησης, ποικίλα χαρακτηριστικά για τους διαφορετικούς πόρους και τις οντότητες ενός συστήματος κλπ.

Ορίζονται τέσσερις γενικές κατηγορίες για τα υψηλού επιπέδου δίκτυα Petri :

- (i) επεκτάσεις που εμφανίζουν **διακεκριμένες μάρκες (individual tokens)**, οι οποίες και καλούνται **“καθαρά” δίκτυα Petri υψηλού επιπέδου (pure HPNs)**,
- (ii) **υψηλού επιπέδου δίκτυα που εμφανίζουν τροποποιημένη σημασιολογία (high-level PN with modified semantics)**,
- (iii) επεκτάσεις που προτείνουν συγκεκριμένους μηχανισμούς δόμησης, γνωστές ως **ιεραρχικά δίκτυα Petri υψηλού επιπέδου (hierarchical high-level PN / HHPNs)**, και
- (iv) μοντέλα που υιοθετούν χαρακτηριστικά από άλλες τυπικές μέθοδες προσδιορισμού προδιαγραφών.

3.2.1 Κατηγορίες Δικτύων Petri Υψηλού Επιπέδου

Στην πρώτη κατηγορία, δηλαδή στα μοντέλα με διακεκριμένες μάρκες (“καθαρά” δίκτυα Petri Υψηλού Επιπέδου), ανήκουν φορμαλισμοί όπως τα **Χρωματισμένα δίκτυα Petri (Coloured PN - CPNs)**, τα **δίκτυα Κατηγορήματος-Μετάβασης (Predicate-Transition nets / PrT-nets)** και τα **δίκτυα με Διακεκριμένες Μάρκες (Individual Token Nets - ITNs)**. Τα Χρωματισμένα δίκτυα Petri επεκτείνουν τα δίκτυα χαμηλού επιπέδου συσχετίζοντας χρώματα με τις μάρκες, τις θέσεις ή και τις μεταβάσεις ενός δικτύου. Σε αντιστοιχία με τις έννοιες που συναντούμε σε μια γλώσσα προγραμματισμού, με τον όρο χρώμα χαρακτηρίζεται απλά ένας τύπος, μια μέθοδος δηλαδή με την οποία διακρίνονται κλάσεις αντικειμένων που παρουσιάζουν χαρακτηριστικά μιας κοινής δομικής κατηγορίας. Ένα Χρωματισμένο δίκτυο Petri μπορεί να μετασχηματιστεί στο αντίστοιχο, αλλά σαφώς πιο σύνθετο, δίκτυο Θέσης-Μετάβασης, όταν ο αριθμός των χρωμάτων του δικτύου είναι πεπερασμένος. Αυτό συμβαίνει διότι κάθε δίκτυο Θέσης-Μετάβασης μπορεί να θεωρηθεί ως ειδική περίπτωση (ένα στιγμιότυπο) ενός αντίστοιχου Χρωματισμένου δικτύου Petri, όπου κάθε σύνολο χρωμάτων αποτελείται από ένα και μοναδικό στοιχείο. Στην γενική περίπτωση όμως, όταν ο αριθμός των χρωμάτων γίνεται μη πεπερασμένος (σε μη πεπερασμένα δίκτυα), το μοντέλο των Χρωματισμένων δικτύων Petri αποκτά την υπολογιστική ισχύ των Μηχανών Turing, επιτρέποντας έτσι την αναπαράσταση κάθε υπολογιστικού συστήματος. Τα Χρωματισμένα δίκτυα Petri θα αναλυθούν εκτενέστερα στο επόμενο κεφάλαιο.

Τα Χρωματισμένα δίκτυα Petri παρουσιάζονται ισοδύναμα σε σχέση με τα δίκτυα Κατηγορήματος-Μετάβασης, όσον αφορά στην ισχύ υπολογισμού. Αυτό σημαίνει ότι κάθε έννοια, αλγόριθμος ανάλυσης ή θεώρημα που βρίσκει εφαρμογή στο πρώτο μοντέλο μπορεί να εφαρμοστεί ανάλογα και στο δεύτερο μοντέλο. Ωστόσο τα Χρωματισμένα δίκτυα Petri κρίνονται ως περισσότερο κατάλληλα στο να περιγράψουν ευκρινώς τη λειτουργία ενός συστήματος. Ο φορμαλισμός των δικτύων Κατηγορήματος-Μετάβασης βασίζεται σε μια **σημειογραφία αλγεβρικού τύπου (algebraic notation)** και γι'αυτό το λόγο τα δίκτυα Κατηγορήματος-Μετάβασης αποτέλεσαν τον πρόδρομο μιας σειράς δικτύων Petri υψηλού επιπέδου που συνιστούν αλγεβρικές επεκτάσεις στο βασικό μοντέλο. Έτσι, για παράδειγμα, τα βέλη ενός δικτύου Κατηγορήματος-Μετάβασης επιγράφονται με **αθροίσματα ν-άδων μεταβλητών (sums of tuples of variables)**.

Στην κατηγορία των επεκτάσεων με διακεκριμένες μάρκες ανήκουν και τα δίκτυα με Διακεκριμένες Μάρκες. Ο φορμαλισμός τους εμφανίζεται ανάλογος με εκείνον των Χρωματισμένων δικτύων. Ωστόσο στο μοντέλο οι κανόνες ενεργοποίησης και πυροδότησης κάθε μετάβασης παρουσιάζουν μικρές διαφοροποιήσεις. Το μοντέλο μπορεί να συσχετίζει επιπρόσθετες **συνθήκες (conditions)** και **λειτουργίες (operations)** με κάθε μετάβαση, επηρεάζοντας αντίστοιχα με τον τρόπο αυτό τους κανόνες ενεργοποίησης και πυροδότησης.

Στα δίκτυα Petri υψηλού επιπέδου περιλαμβάνεται ένα σύνολο μοντέλων που παρουσιάζουν τροποποιημένη σημασιολογία, σε ό,τι αφορά στον καθορισμό της δυναμική τους συμπεριφοράς. Τα μοντέλα αυτής της κατηγορίας προτείνουν τροποποιήσεις πάνω σε συγκεκριμένες παραμέτρους, παραδείγματος χάριν στη χωρητικότητα των θέσεων, ορίζουν συγκεκριμένους τύπους για τα βέλη, τις μεταβάσεις και τις θέσεις ή ακόμη τροποποιούν τη σημασιολογία της πυροδότησης των μεταβάσεων.

Ωστόσο, για την περίπτωση συστημάτων που εμφανίζουν υψηλό βαθμό πολυπλοκότητας, ακόμη και εκείνα τα μοντέλα που εμφανίζουν τροποποιημένα σημασιολογικά χαρακτηριστικά, δεν μπορούν να κριθούν ως επαρκή όσον αφορά στη δυνατότητα περιγραφής. Η κύρια αιτία για τον περιορισμό αυτό είναι ότι οι περισσότεροι φορμαλισμοί παρέχουν μια απλή, **“επίπεδη” όψη (flat view)** για το υπό μοντελοποίηση σύστημα. Για τη λύση του προβλήματος έχουν προταθεί τα **Ιεραρχικά δίκτυα Petri υψηλού επιπέδου (hierarchical high-level PNs/HHPNs)**. Αυτά ενσωματώνουν συγκεκριμένους μηχανισμούς με σκοπό τη συστηματική και ιεραρχική δόμηση ενός δικτύου. Ένα ιεραρχικό μοντέλο δίνει έμφαση (i) είτε στην αντικατάσταση (αποσύνθεση) των θέσεων ή των μεταβάσεων του δικτύου από τμήματα δικτύων (υποδίκτυα) που παρουσιάζουν υψηλότερο βαθμό λεπτομέρειας, (ii) είτε στη σύνθεση των τμημάτων ενός δικτύου μέσω της σύμπτυξης μεταβάσεων, θέσεων ή βελών. Οι διαδικασίες σύνθεσης προδιαγράφουν και τους μηχανισμούς επικοινωνίας μεταξύ των τμημάτων ενός δικτύου.

Συγκεκριμένα, η σύνθεση μέσω της **σύμπτυξης μεταβάσεων** επιτρέπει την αναπαράσταση σύγχρονης επικοινωνίας. Η προσέγγιση αυτή πρέπει να διατηρεί σταθερές τις ιδιότητες τόσο των δικτύων-συνιστωσών (υποδίκτυα) όσο και του συνολικού δικτύου, με αποτέλεσμα να διευκολύνεται η εφαρμογή της διαδικασίας της ανάλυσης. Η σύμπτυξη μεταβάσεων όμως μπορεί να οδηγήσει σε **υποδίκτυα που παρουσιάζουν ισχυρό βαθμό σύζευξης (tightly coupled net components)**, με αποτέλεσμα κάθε υποδίκτυο να μην αντιστοιχεί ευκρινώς σε ένα αυτόνομο υποσύστημα του συστήματος.

Η σύνθεση μέσω της **σύμπτυξης θέσεων** επιτρέπει την αναπαράσταση ασύγχρονης επικοινωνίας ανάμεσα σε υποδίκτυα. Στην περίπτωση αυτή η επικοινωνία λαμβάνει χώρα μέσω κοινών θέσεων που αντιστοιχούν στους **κοινά διαμοιραζόμενους πόρους (shared resources)** του υπό αναπαράσταση συστήματος. Αυτή η τεχνική είναι συμβατή με τη βασική θεώρηση της θεωρίας των δικτύων Petri, αφού οι μεταβάσεις θεωρούνται ως οι ενεργητικές οντότητες που επικοινωνούν μεταξύ τους μέσω των θέσεων, οι οποίες με τη σειρά τους παίζουν τον ρόλο των παθητικών οντοτήτων. Το πρόβλημα όμως που μπορεί να προκύψει στην περίπτωση αυτή είναι ότι απαιτείται επιπρόσθετη πολυπλοκότητα στην περιγραφή, όταν πρέπει να εξασφαλιστεί **ορθότητα κατά το συγχρονισμό (correct synchronization)** των επικοινωνούντων

υποδικτύων. Επίσης, παραμένει να είναι πιθανό το πρόβλημα του ισχυρού βαθμού σύζευξης ανάμεσα στις συνιστώσες του τελικού δικτύου.

Τέλος, ενδιαφέρον παρουσιάζει η σύνθεση μέσω της σύμπτυξης βελών, η οποία μπορεί να οδηγήσει σε χαμηλό βαθμό σύζευξης. Ακολουθώντας την τεχνική αυτή, η επικοινωνία ανάμεσα στα υποδίκτυα μπορεί να θεωρηθεί ως μια **διαδικασία ανταλλαγής μηνυμάτων (message passing)**, κατά την οποία στέλνονται και λαμβάνονται δεδομένα (μηνύματα) ανάμεσα στις συνιστώσες ενός δικτύου.

Μολονότι τα ιεραρχικά δίκτυα Petri υψηλού επιπέδου διευκολύνουν τη συστηματική προδιαγραφή των παραμέτρων ενός συστήματος, οι διαδικασίες ανάλυσης συνήθως εφαρμόζονται στο τελικά παραγόμενο, επίπεδο, πολυσύνθετο και εκτελέσιμο δίκτυο. Κατά συνέπεια, το πρόβλημα της έκρηξης του χώρου καταστάσεων παραμένει ως πιθανό. Για το λόγο αυτό, έχουν προταθεί συγκεκριμένοι μηχανισμοί δόμησης και ολοκληρωμένες τεχνικές που υποστηρίζουν διαδικασίες άμεσης, **“συνθετικής” (“compositional”)** ανάλυσης στο τελικά παραγόμενο δίκτυο. Οι τεχνικές αυτές δίνουν έμφαση στον προσδιορισμό τμημάτων δικτύου που υποκαθιστούν με ισοδύναμο τρόπο (δηλ. παρουσιάζουν τις ίδιες ιδιότητες) με συγκεκριμένα υποδίκτυα. Με τον τρόπο αυτό επιτυγχάνεται μείωση της πολυπλοκότητας του χώρου καταστάσεων ενός δικτύου.

3.3 Χρονικές Εκδόσεις Δικτύων Petri

Η κοινή συνισταμένη όλων των δικτύων Petri που αναλύθηκαν στα προηγούμενα κεφάλαια - είτε πρόκειται για χαμηλού, είτε για υψηλού επιπέδου μοντέλα - είναι ότι εμφανίζουν ασύγχρονη, μη ντετερμινιστική και χρονικά ανεξάρτητη δυναμική συμπεριφορά. Τα μοντέλα δεν λαμβάνουν υπόψη τη χρονική διάσταση της συμπεριφοράς ενός συστήματος και η εκτέλεσή τους δεν εξαρτάται από τη “ροή” του χρόνου. Πρόκειται λοιπόν για συμβατικά, μη-χρονικά μοντέλα, στα οποία οποιεσδήποτε, ανεξάρτητες μεταξύ τους, πυροδοτήσεις μεταβάσεων δεν προβάλλονται πάνω στο γραμμικό χρονικό άξονα. Αντίθετα, οποιεσδήποτε, ανεξάρτητες μεταξύ τους, πυροδοτήσεις μεταβάσεων συνδέονται με μια **μερική, μη-παρεμβαλλόμενη στο χρόνο διάταξη (noninterleaving partial-order relation)**. Η σχέση αυτή δεν προσδιορίζει τη χρονική διάταξη που ισχύει ανάμεσα σε δύο ή περισσότερα γεγονότα (πυροδοτήσεις μεταβάσεων), καθώς η διάταξη λαμβάνει χώρα με τρόπο τυχαίο.

Έτσι τα κλασσικά δίκτυα Petri, ενώ είναι χρήσιμα στη διερεύνηση των ποιοτικών ή λογικών ιδιοτήτων ταυτόχρονων συστημάτων, όπως είναι ο αμοιβαίος αποκλεισμός, η παρουσία και η απουσία αδιεξόδων, η περατότητα και η δικαιοσύνη, πρέπει να επαυξηθούν με χρονικές παραμέτρους, ώστε να μπορούν να εκτιμήσουν ποσοτικά ένα σύστημα. Η πρώτη προσέγγιση έγινε από τον Ramchandani το 1973 στο MIT και ακολούθησαν ο Sifakis το 1977 και οι Ramamoorthy και Ho το 1980. Η έννοια του χρόνου εισάγεται σε διάφορες επεκτάσεις, τις λεγόμενες **χρονικές εκδόσεις δικτύων Petri (timed versions of PNs)**, μοντέλα που χρησιμοποιήθηκαν ευρύτατα για την αναπαράσταση ταυτόχρονων συστημάτων με χρονικά εξαρτημένη συμπεριφορά και, ειδικότερα, για τη μοντελοποίηση συστημάτων με χρονικές απαιτήσεις. Σήμερα πάρα πολλά εργαλεία λογισμικού χρησιμοποιούν χρονικές εκδόσεις δικτύων Petri για τη

μοντελοποίηση και την ανάλυση απόδοσης συστημάτων. Ένας πλήρης κατάλογος με εργαλεία λογισμικού παρατίθεται στην ιστοσελίδα :

<http://www.daimi.au.dk/PetriNets/tools>

Ένα από τα πιο γνωστά εργαλεία που χρησιμοποιούν δίκτυα Petri με χρόνο είναι το TimeNet. Περισσότερες πληροφορίες υπάρχουν στην ιστοσελίδα του εργαλείου :

<http://www.pdv.cs.tu-berlin.de/~timenet>

Αντιπροσωπευτικές χρονικές εκδόσεις των δικτύων Petri είναι τα **Χρονισμένα δίκτυα (Timed PNs)**, τα **Απλά Χρονικά δίκτυα (Simple Time PNs)**, τα **Χρονικά δίκτυα (Time PNs)**, τα **Επικοινωνούντα Χρονικά δίκτυα (Communicating Time PNs)**, τα δίκτυα με **Χρονισμένες Θέσεις (Timed Place PNs)**, τα δίκτυα με **Χρονικούς Περιορισμούς (Timing Constraint PNs)**, τα (Γενικευμένα) **Στοχαστικά δίκτυα (Stochastic (Generalized or not) PNs)**, τα **Χρονικά δίκτυα με Ουρές (Queuing PNs)**, τα δίκτυα **Περιβάλλοντος-Συσχέτισης (Environment-Relationship Nets)**, τα **Χρονικά Βασικά δίκτυα (Time Basic nets)**, τα **Χρονισμένα δίκτυα με Χρώματα (Timed Coloured PNs)**, και τέλος, τα **Χρονισμένα δίκτυα με Χρώματα και Χρονικά Διαστήματα (Interval Timed Coloured PNs)**.

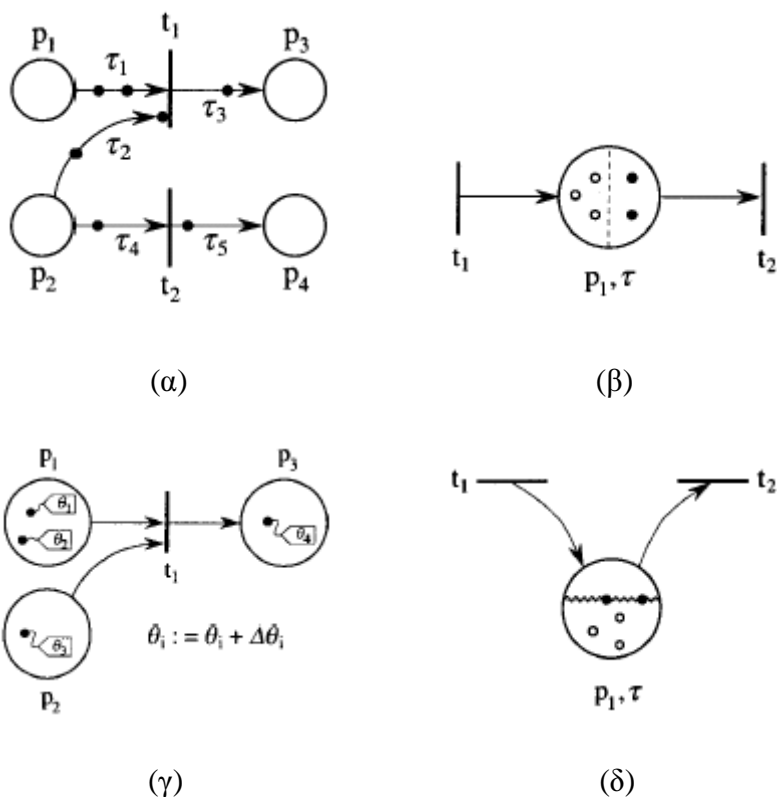
Η διαδικασία ορισμού χρονικών παραμέτρων σε ένα μη-χρονικό, συμβατικό μοντέλο εξαρτάται από να σύνολο από γενικούς βασικούς παράγοντες και επιλογές. Συγκεκριμένα, θα πρέπει να ληφθούν υπόψη τα παρακάτω βασικά ζητήματα :

➤ *Εντοπισμός των Χρονικών Στοιχείων (Location of Time Semantics)*

Σε ένα χρονικό μοντέλο χαμηλού επιπέδου οι χρονικές προδιαγραφές μπορούν να συνδεθούν με διαφορετικά στοιχεία ενός δικτύου Petri. Υπάρχουν τέσσερις διαφορετικοί τρόποι :

- (i) **Χρονισμένες μεταβάσεις (timed transitions)** : Ο χρόνος συνδέεται με τις μεταβάσεις ενός δικτύου. Η εκκίνηση της δράσης που σχετίζεται με μια μετάβαση αντιστοιχεί στην ενεργοποίηση της μετάβασης, ενώ ο τερματισμός της δράσης αντιστοιχεί στην πυροδότηση της μετάβασης. Σχήμα 3(α)
- (ii) **Χρονισμένες θέσεις (timed places)** : Ο χρόνος συνδέεται με τις θέσεις ενός δικτύου. Οι μάρκες στη θέση εισόδου μιας μετάβασης επιτρέπεται να πυροδοτήσουν τη μετάβαση, μόνο μετά το τέλος της χρονικής καθυστέρησης που σχετίζεται με τη συγκεκριμένη θέση εισόδου. Σχήμα 3(β)
- (iii) **Χρονισμένα τόξα (timed arcs)** : Ο χρόνος συνδέεται με τα τόξα ενός δικτύου. Μια καθυστέρηση ταξιδιού (traveling delay) συνδέεται με κάθε τόξο. Οι μάρκες μπορούν να πυροδοτήσουν μια μετάβαση μόνο όταν φτάσουν στη μετάβαση. Σχήμα 3(γ)
- (iv) **Χρονισμένες μάρκες (timed tokens)** : Στην περίπτωση ενός χρονικού μοντέλου υψηλού επιπέδου, οι χρονικές προδιαγραφές συνήθως σχετίζονται με τις μάρκες του δικτύου, υπό τον τύπο **χρονικών σφραγίδων (time stamps)**. Οι μάρκες φέρουν μια χρονική σφραγίδα, η οποία υποδεικνύει πότε μπορούν να πυροδοτήσουν μια μετάβαση. Αυτή η

χρονική σφραγίδα μπορεί να προσαυξηθεί σε κάθε πυροδότηση μετάβασης. Σχήμα 3(δ)



Σχήμα 3: Χρονικά στοιχεία δικτύων Petri

➤ Ο Τύπος Χρονικών Στοιχείων (Type of Time Semantics)

Οι διάφορες χρονικές εκδόσεις δικτύων Petri παρουσιάζουν, ανάλογα με την περιοχή εφαρμογής, διαφορετικούς τύπους χρονικών προδιαγραφών. Συγκεκριμένα, για την αναπαράσταση της χρονικής διάρκειας μιας δράσης ενός συστήματος, μπορούν να χρησιμοποιηθούν :

- i. σταθεροί χρονομετρητές οι οποίοι εισάγουν **σταθερές χρονικές καθυστερήσεις (fixed delays)**,
- ii. **χρονικά διαστήματα (time intervals)** που προσδιορίζονται από ζεύγη **χρονικών τιμών (time pairs)** ή, ακόμη
- iii. **στοχαστικές καθυστερήσεις (stochastic delays)**.

➤ Ο Τύπος Χρονικής Αναφοράς (Time Mode)

Η επιλογή του χρονικού σημείου, που αποτελεί το σημείο αναφοράς για την εκτέλεση ενός μοντέλου, καθορίζει τη δυναμική συμπεριφορά του μοντέλου στο χρόνο, και κατά συνέπεια, τον τρόπο με τον οποίο θα εφαρμοστεί η διαδικασία της ανάλυσης. Υπάρχουν δύο εναλλακτικές δυνατότητες. Πρώτον, όταν η χρονική αναφορά είναι

απόλυτη (absolute time mode), οι μεταβάσεις από μια κατάσταση (μαρκάρισμα) σε μια άλλη, εξαιτίας των πυροδοτήσεων των μεταβάσεων του δικτύου, λαμβάνουν χώρα σε προκαθορισμένες χρονικές στιγμές. Δεύτερον, όταν επιλεγεί η χρονική αναφορά να είναι **σχετική (relative time mode)**, κάθε αλλαγή από μια κατάσταση σε μια άλλη είναι σχετική με τη χρονική στιγμή της εμφάνισης ενός συγκεκριμένου γεγονότος, δηλαδή, πραγματοποιείται σε ένα χρονικό ορίζοντα από τη στιγμή που εναποτίθενται μάρκες στις θέσεις εισόδου μιας μετάβασης, καθιστώντας με τον τρόπο αυτό τη μετάβαση ενεργοποιημένη.

➤ *Η Σημασιολογία του Τρόπου Πυροδότησης των Μεταβάσεων (Semantics of Transition Firing Mode)*

Όλα τα συμβατικά, μη-χρονικά μοντέλα υιοθετούν τον **ασθενή κανόνα πυροδότησης (weak firing mode)** για τη εκτέλεση κάθε μετάβασης του δικτύου. Ο ασθενής κανόνας έχει άμεση σχέση με τη μη-ντετερμινιστική συμπεριφορά του μοντέλου. Συγκεκριμένα, το γεγονός ότι μια μετάβαση είναι ενεργοποιημένη δεν σημαίνει αναγκαστικά ότι αυτή θα πυροδοτήσει. Στις χρονικές εκδόσεις των **δικτύων Petri** μπορεί συμβατικά να θεωρηθεί, είτε ο ασθενής κανόνας, είτε, διαφορετικά, να υιοθετηθεί ο **ισχυρός κανόνας πυροδότησης (strong firing mode)**, σύμφωνα με τον οποίο κάθε ενεργοποιημένη μετάβαση πυροδοτείται κατά τρόπο άμεσο και υποχρεωτικό.

➤ *Ο Βαθμός Ενεργοποίησης για κάθε Μετάβαση (Degree of Transition Enabling)*

Ο βαθμός ενεργοποίησης για κάθε μετάβαση ενός χρονικού μοντέλου πρέπει κατά σύμβαση να καθοριστεί στην περίπτωση που ο συνολικός αριθμός των μαρκών σε μια θέση εισόδου επιτρέπει πολλαπλές πυροδοτήσεις της μετάβασης. Ο βαθμός ενεργοποίησης προσδιορίζει το μέγιστο αριθμό όλων των ανεξάρτητων μεταξύ τους περιπτώσεων, κατά τις οποίες κάθε μετάβαση είναι ενεργοποιημένη (maximum number of transition “enablings”) και μπορεί ταυτόχρονα να πυροδοτήσει.

➤ *Ο Τύπος Χρονικής Ροής (Time Model)*

Το κοινό χαρακτηριστικό που κατά σύμβαση υιοθετούν ως επί το πλείστον όλες οι χρονικές εκδόσεις είναι ότι είναι δυνατόν εναλλακτικά να ακολουθούν το **δυϊκό μοντέλο “ροής” του χρόνου (dual model of time)**. Ένα χρονικό μοντέλο μπορεί συγκεκριμένα να εκτελεστεί υποθέτοντας, είτε **πυκνή χρονική ροή (dense time model)**, είτε **σποραδική χρονική ροή (sparse time model)**. Στην πρώτη περίπτωση υποθέτουμε ότι ανάμεσα σε οποιοδήποτε δύο χρονικές στιγμές μεσολαβεί ένας μη πεπερασμένος αριθμός από άλλες χρονικές στιγμές, ενώ στη δεύτερη περίπτωση υποθέτουμε ότι ανάμεσα σε δύο τυχαίες χρονικές στιγμές μεσολαβεί ένας πεπερασμένος αριθμός από **διακεκριμένες (discrete)** χρονικές στιγμές, που τοποθετούνται σε συγκεκριμένη χρονική απόσταση μεταξύ τους με βάση ένα **ελάχιστο χρονικό διάστημα (minimum time granularity)**. Τα περισσότερα χρονικά μοντέλα μπορούν να εκτελεστούν δυναμικά, είτε με την πρώτη, είτε με τη δεύτερη επιλογή. Όμως το πρόβλημα της έκρηξης του χώρου καταστάσεων ενός μοντέλου (άρα και η δυσκολία εφαρμογής των μεθόδων ανάλυσης) επαυξάνει με την επιλογή

της πυκνής χρονικής ροής, καθώς τότε οι πιθανές καταστάσεις (μαρκαρίσματα) στις οποίες μπορεί να βρεθεί το δίκτυο, κατά την εκτέλεσή του στο χρόνο, είναι σαφώς περισσότερες.

Στη συνέχεια του κεφαλαίου θα περιγραφούν οι πιο σημαντικές χρονικές εκδόσεις των δικτύων Petri :

- i. τα δίκτυα Petri με χρονισμένες μεταβάσεις και
- ii. τα δίκτυα Petri με χρονισμένες θέσεις.

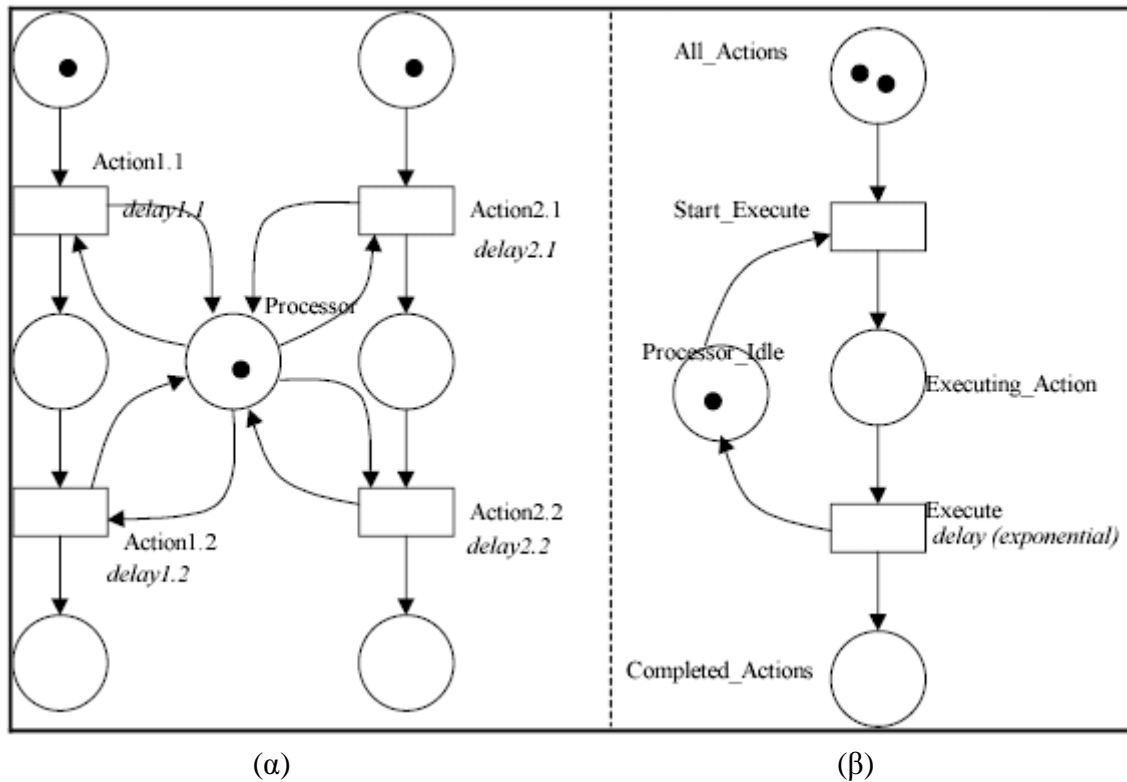
Κατά την παρουσίαση του κάθε μοντέλου θα αναλυθούν θέματα που σχετίζονται με τη αναπαράσταση των χρονικών προδιαγραφών και με την πρακτική εφαρμογή του σε προβλήματα μοντελοποίησης συστημάτων με χρονικά εξαρτημένη και ταυτόχρονη συμπεριφορά.

3.4 Δίκτυα Petri με Χρονισμένες Μεταβάσεις

Στη κατηγορία των δικτύων Petri με χρονισμένες μεταβάσεις (**Timed Transition PNs**) ανήκουν μοντέλα που συσχετίζουν χρονικές προδιαγραφές με τις μεταβάσεις ενός δικτύου. Πρόκειται για την κατηγορία που περιλαμβάνει τις περισσότερες χρονικές εκδόσεις. Τα μοντέλα της κατηγορίας ακολουθούν γενικά δύο εναλλακτικές προσεγγίσεις για την αναπαράσταση εφαρμογών με χρονικά εξαρτημένη συμπεριφορά.

Η πρώτη προσέγγιση επικεντρώνεται στην **αναπαράσταση των δράσεων (action oriented design)** που συνθέτουν μια εφαρμογή. Περιγράφεται με λεπτομέρεια ο τρόπος με τον οποίο κάθε δράση αποσυντίθεται σε επιμέρους δράσεις. Κάθε δράση αντιστοιχεί σε μια διαφορετική μετάβαση και η χρονική διάρκεια εκτέλεσης της δράσης εκφράζεται με ένα χρονικό προσδιορισμό για την αντίστοιχη μετάβαση. Η διαθεσιμότητα ενός επεξεργαστή ή οποιουδήποτε άλλου κοινά διαμοιραζόμενου πόρου είτε αναπαρίσταται από την ύπαρξη μιας μάρκας σε μια αντίστοιχη θέση, είτε, πολλές φορές, δεν αναπαρίσταται καθόλου. Στο Σχήμα 4(α) δίνεται ένα παράδειγμα δικτύου που επικεντρώνεται στην αναπαράσταση δράσεων. Η προσέγγιση γενικά μπορεί να εφαρμοστεί για να αναπαρασταθεί η δομή μιας εφαρμογής. Παρουσιάζει όμως μειονεκτήματα καθώς αδυνατεί να περιγράψει φαινόμενα όπως η καταχώρηση των πόρων του συστήματος στις δράσεις και η **διακοπή εκτέλεσης (preemption)** των δράσεων.

Η δεύτερη προσέγγιση επικεντρώνεται στην **αναπαράσταση των πόρων (resource oriented design)** που χρησιμοποιούν οι δράσεις της εφαρμογής. Οι δράσεις αναπαρίστανται με μάρκες που διακινούνται στο δίκτυο. Δίνεται έμφαση στην αναπαράσταση της δέσμευσης και της αποδέσμευσης κάθε εμπλεκόμενου πόρου (π.χ. του επεξεργαστή) αλλά δεν περιγράφεται ο τρόπος αποσύνθεσης κάθε δράσης, όσο και η ακριβής χρονική διάρκεια της εκτέλεσής της. Η προσέγγιση εφαρμόζεται συνήθως στο πεδίο της ανάλυσης της απόδοσης εφαρμογών. Στο Σχήμα 4(β) δίνεται ένα παράδειγμα δικτύου που δίνει έμφαση στην αναπαράσταση πόρων.



Σχήμα 4: Δίκτυα χρονισμένων μεταβάσεων : (α) σχεδίαση επικεντρωμένη στις δράσεις και (β) σχεδίαση επικεντρωμένη στους πόρους

Αντιπροσωπευτικά μοντέλα στην κατηγορία των δικτύων με χρονισμένες μεταβάσεις είναι τα **Χρονισμένα δίκτυα (Timed PNs)** και τα **Χρονικά δίκτυα (Time PNs)**. Στη γενική περίπτωση, και σε σύγκριση με το μοντέλο των Χρονικών δικτύων, ένα Χρονισμένο δίκτυο υπολείπεται σε εκφραστικές δυνατότητες καθώς δεν μπορεί να περιγράψει αλλαγές στη διάρκεια της πυροδότησης μιας μετάβασης, και συνεπώς, διακυμάνσεις στη χρονική διάρκεια μιας αντίστοιχης δράσης της υπό σχεδίαση εφαρμογής (εάν υιοθετηθεί ο προσανατολισμός σε δράσεις). Το μοντέλο όμως των Χρονισμένων δικτύων υπερέχει στη δυνατότητα εφαρμογής μεθόδων ανάλυσης καθώς (πάντοτε στη γενική περίπτωση) παρουσιάζει σαφώς μικρότερο χώρο καταστάσεων. Τα δύο μοντέλα θα παρουσιαστούν κριτικά στις επόμενες παραγράφους.

3.4.1 Χρονισμένα Δίκτυα Petri

Στο μοντέλο των Χρονισμένων δικτύων κάθε μετάβαση συνδέεται με ένα σταθερό χρονομετρητή που αντιστοιχεί σε μια συγκεκριμένη, κάθε φορά, σταθερή χρονική καθυστέρηση. Η πυροδότηση μιας μετάβασης καθυστερεί για χρονική διάρκεια ίση με τον αντίστοιχο χρονομετρητή. Στο χρονικό διάστημα αυτό διατηρούνται στις θέσεις εισόδου της μετάβασης όλες οι μάρκες που την κατέστησαν ενεργοποιημένη. Τελικά, η μετάβαση πυροδοτείται όταν λήξει ο χρονομετρητής, δηλαδή με την “εκπνοή” της αντίστοιχης καθυστέρησης. Κάθε χρονική καθυστέρηση είναι σαφώς καθορισμένη και προβλέψιμη (ντετερμινιστική). Για το λόγο αυτό, εάν υιοθετηθεί το παράδειγμα σχεδίασης που επικεντρώνεται στις δράσεις, το μοντέλο κρίνεται ως

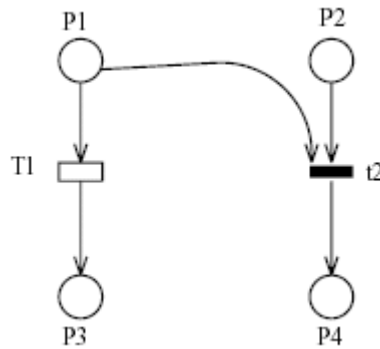
κατάλληλο για να εκφραστούν **χρόνοι εκτέλεσης χειρότερης περίπτωσης (worst-case execution times)** για τις δράσεις, που συνθέτουν μια εφαρμογή με απαιτήσεις πραγματικού χρόνου.

Το μοντέλο ακολουθεί τον ισχυρό κανόνα πυροδότησης για κάθε μετάβαση. Η τυχαία μετάβαση t συνδέεται με μια χρονική καθυστέρηση T_{del} και θα πυροδοτήσει άμεσα στη χρονική στιγμή $T_0 + T_{del}$, υπό την προϋπόθεση ότι στις θέσεις εισόδου της t υπάρχει ο απαιτούμενος αριθμός μαρκών από τη χρονική στιγμή T_0 . Ένα Χρονισμένο δίκτυο είναι δυνατόν να περιλαμβάνει και **άμεσες μεταβάσεις (immediate transitions)**, προκειμένου να εκφραστεί η εκτέλεση συγκεκριμένων δράσεων και η αλλαγή λογικών συνθηκών που λαμβάνουν χώρα χωρίς να καταναλώνουν χρόνο (δηλ. δράσεις και συνθήκες που η εκτέλεση τους και ο έλεγχος τους θεωρείται αμελητέας χρονικής διάρκειας). Κάθε άμεση μετάβαση συνδέεται αντίστοιχα με μια **μηδενική καθυστέρηση (null delay)** και εάν ενεργοποιηθεί πυροδοτείται άμεσα. Οι άμεσες μεταβάσεις πλεονεκτούν σε **προτεραιότητα (priority)**, σε σύγκριση με τις χρονισμένες μεταβάσεις. Έτσι οι πρώτες υπερέχουν από τις δεύτερες σε καταστάσεις **σύγκρουσης/απόφασης (conflict/decision)**.

Στη γραφική αναπαράσταση των Χρονισμένων δικτύων Petri, οι άμεσες μεταβάσεις παριστάνονται ως μπάρες και συνήθως συμβολίζονται με ένα όνομα τύπου t_x , όπου x κάποιος αριθμός ή κάποια μνημονική λέξη, ενώ οι χρονισμένες μεταβάσεις παριστάνονται ως ορθογώνια κουτιά (λευκά ή μαύρα) και συνήθως συμβολίζονται με ένα όνομα τύπου T_x .

Οι συγκρούσεις που περιλαμβάνουν άμεσες και χρονισμένες μεταβάσεις έχουν μια σημαντική χρήση στα Χρονισμένα δίκτυα Petri : επιτρέπουν τη διακοπή (ή την προεκχώρηση) των δράσεων που βρίσκονται σε εξέλιξη, όταν συμβαίνει μια έκτακτη κατάσταση. Έστω για παράδειγμα, το υποδίκτυο του Σχήματος 5 που έχει μία χρονισμένη μετάβαση, τη $T1$ και μία άμεση μετάβαση, τη $t2$. Μία μάρκα στη θέση $p1$ εκκινεί τη δράση που μοντελοποιείται από τη $T1$, δηλαδή ενεργοποιεί τη $T1$. Εάν μία μάρκα φτάσει στη θέση $p2$ πριν την πυροδότηση της $T1$, η άμεση μετάβαση $t2$ ενεργοποιείται και πυροδοτεί, με αποτέλεσμα να απενεργοποιείται η χρονισμένη μετάβαση $T1$. Η συμπεριφορά αυτή αποτελεί παράδειγμα της προτεραιότητας των άμεσων μεταβάσεων έναντι των χρονισμένων.

Η παρουσία άμεσων μεταβάσεων προκαλεί τη διάκριση των μαρκαρισμάτων. Τα μαρκαρίσματα στα οποία καμία άμεση μετάβαση δεν είναι ενεργοποιημένη ονομάζονται **απτά (tangible)**, ενώ τα μαρκαρίσματα στα οποία τουλάχιστον μία άμεση μετάβαση είναι ενεργοποιημένη ονομάζονται **εξαφανιζόμενα (vanishing)**. Το σύστημα του Χρονισμένου δικτύου Petri ξοδεύει θετική ποσότητα χρόνου σε απτά μαρκαρίσματα και μηδενικό χρόνο σε εξαφανιζόμενα μαρκαρίσματα.



Σχήμα 5: Διακοπή μιας δράσης λόγω άμεσης μετάβασης

Η χρονική καθυστέρηση μιας μετάβασης μπορεί να ερμηνευτεί ως ένα **τοπικό ρολόι-χρονομετρητής (local clock-timer)**. Ο χρονομετρητής αναπαριστά τη χρονική διάρκεια κατά την οποία η μετάβαση παραμένει ενεργοποιημένη, πριν να γίνει τελικά εκτελέσιμη. Όταν δημιουργείται ικανοποιητικός αριθμός από μάρκες στις θέσεις εισόδου μιας μετάβασης, τότε ο χρονομετρητής της μετάβασης τίθεται στην αρχική του τιμή (ίση με τη χρονική καθυστέρηση της μετάβασης). Στη συνέχεια μειώνεται σταθερά, και τελικά η μετάβαση πυροδοτείται δημιουργώντας μάρκες στις θέσεις εξόδου, όταν ο χρονομετρητής τεθεί στη μηδενική τιμή.

Κάθε μετάβαση ακολουθεί λοιπόν τον **ατομικό κανόνα εκτέλεσης-πυροδότησης (atomic transition firing rule)**, αφού οι ενέργειες της αφαίρεσης μαρκών, από τις θέσεις εισόδου της και της προσθήκης μαρκών, στις θέσεις εξόδου της μετάβασης, συνιστούν μια άμεση και αδιαίρετη λειτουργία. Ο ατομικός κανόνας είναι σε πλήρη σύμβαση με τη σημασιολογία των συμβατικών, μη-χρονικών μοντέλων, όπου η πυροδότηση κάθε μετάβασης γίνεται άμεσα και αδιαίρετα. Στην περίπτωση του ατομικού κανόνα πυροδότησης οι χρονικές καθυστερήσεις αποτελούν **χρόνους ενεργοποίησης (enabling times)** για τις μεταβάσεις του δικτύου.

Ωστόσο ορισμένα μοντέλα χρονισμένων μεταβάσεων υιοθετούν τον **κανόνα εκτέλεσης - πυροδότησης των τριών φάσεων (three phases firing rule)**, σύμφωνα με τον οποίο μια ενεργοποιημένη μετάβαση αρχικά αφαιρεί μάρκες από τις θέσεις εισόδου (μια πρώτη φάση που δεν καταναλώνει χρόνο), στη συνέχεια “κρύβει” τις μάρκες για χρονική περίοδο ίση με τη χρονική καθυστέρηση που προσδιορίζει τη μετάβαση (μια δεύτερη φάση που διαρκεί τόσο χρόνο όσο δηλώνει η αντίστοιχη χρονική καθυστέρηση), και τελικά, εναποθέτει μάρκες στις θέσεις εξόδου (μια τρίτη φάση που δεν καταναλώνει χρόνο). Στην περίπτωση λοιπόν του κανόνα πυροδότησης των τριών φάσεων οι χρονικές καθυστερήσεις ερμηνεύονται ως **χρόνοι πυροδότησης (firing times)** για τις μεταβάσεις του δικτύου.

Όμως ο κανόνας πυροδότησης τριών φάσεων βρίσκεται σε αντίθεση με τις υποθέσεις που διέπουν τη βασική συμπεριφορά των συμβατικών, μη-χρονικών μοντέλων. Αντίθετα, τα Χρονισμένα δίκτυα υπό τον ατομικό κανόνα ορίζουν πανομοιότυπα την έννοια της κατάστασης, δηλαδή το μαρκάρισμα ενός δικτύου, με τα αντίστοιχα μη-χρονικά μοντέλα. Επομένως, ένα Χρονισμένο δίκτυο υπό τον ατομικό κανόνα πυροδότησης πλεονεκτεί σε ό,τι αφορά στην ευχρηστία των μεθόδων ανάλυσης.

Συνεπώς, μπορούν σε αυτό να εφαρμοστούν μέθοδοι ανάλυσης όπως η ανάλυση του γραφήματος προσιτότητας.

Επιπλέον, ο ατομικός κανόνας κρίνεται περισσότερο κατάλληλος για να περιγράψει δομές σύγκρουσης, οι οποίες και επιλύονται, κατά την εκτέλεση του δικτύου, με βάση τις τιθέμενες χρονικές προδιαγραφές, δηλαδή με βάση τις χρονικές καθυστερήσεις των μεταβάσεων που βρίσκονται σε σύγκρουση. Συγκεκριμένα, ο ισχυρός κανόνας, όπως αναφέρθηκε και προηγουμένως, ορίζει ότι μια ενεργοποιημένη μετάβαση πυροδοτείται κατά τρόπο υποχρεωτικό. Έτσι η εφαρμογή του ισχυρού κανόνα είναι πιθανό να προκαλέσει την ταυτόχρονη πυροδότηση δύο συγκρουόμενων μεταβάσεων, ένα γεγονός που δεν συμβαδίζει με τον ορισμό της δομής σύγκρουσης.

Για την κατανόηση του προβλήματος μπορεί να θεωρηθεί το Χρονισμένο δίκτυο στο Σχήμα 6 που παρουσιάζει τη δομή μιας Μηχανής Καταστάσεων, αφού κάθε μετάβαση έχει ακριβώς μία θέση εισόδου και ακριβώς μία θέση εξόδου. Οι μεταβάσεις t_3 και t_4 αναπαριστούν δύο ταυτόχρονες δράσεις μιας εφαρμογής, που είναι εντελώς ανεξάρτητες μεταξύ τους, αφού η πυροδότηση της μιας δεν επηρεάζει την πυροδότηση της άλλης. Αντίθετα, οι μεταβάσεις t_1 και t_2 συνθέτουν μια δομή σύγκρουσης, αφού η πυροδότηση της μιας αποτρέπει την πυροδότηση της άλλης.

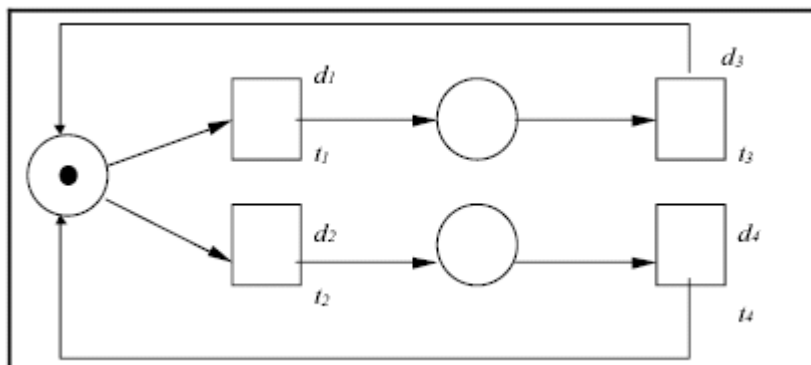
Γενικά, σε μια περίπτωση σύγκρουσης, η επιλογή για το ποια μετάβαση θα πυροδοτήσει πρώτη (και, κατά συνέπεια, η εξασθένηση της μη-ντετερμινιστικής συμπεριφοράς του δικτύου) μπορεί να γίνει υπό τη θεώρηση δύο εναλλακτικών πολιτικών: του **διαγωνισμού σε ταχύτητα (race)** και της **προεπιλογής (preselection)**.

Σύμφωνα με την πολιτική διαγωνισμού σε ταχύτητα, η σύγκρουση επιλύεται με βάση τις χρονικές προδιαγραφές των μεταβάσεων του δικτύου, δηλαδή τους χρονομετρητές των μεταβάσεων. Πρώτη απ' όλες τις συγκρουόμενες μεταβάσεις, σύμφωνα με την πολιτική αυτή, θα πυροδοτήσει εκείνη που συσχετίζεται με την ελάχιστη χρονική καθυστέρηση. Με τον τρόπο αυτό, μπορούν να αποφευχθούν πιθανές ασυνέπειες που προκύπτουν από την εφαρμογή του ισχυρού κανόνα εκτέλεσης – πυροδότησης.

Στο αρχικό μαρκάρισμα του δικτύου που παρουσιάζεται στο Σχήμα 6, οι χρονομετρητές των t_1 και t_2 αρχικοποιούνται στις τιμές d_1 και d_2 αντίστοιχα, όπου $d_1 < d_2$. Μετά από ένα χρονικό διάστημα ίσο με d_1 , η t_1 θα πυροδοτήσει και ο χρονομετρητής της t_2 σταματά να μειώνεται. Τη στιγμή αυτή ο χρονομετρητής της t_3 , ο d_3 αρχίζει να μειώνεται έως ότου μηδενιστεί. Μετά την πυροδότηση της t_3 ισχύει ξανά η σύγκρουση μεταξύ των t_1 και t_2 . Ο χρονομετρητής της t_1 τίθεται εκ νέου ίσος με d_1 , ενώ ο χρονομετρητής της t_2 συνεχίζει να μειώνεται από το χρονικό σημείο στο οποίο προηγουμένως είχε διακοπεί, δηλαδή ξεκινά από τη χρονική στιγμή $d_2 - d_1$. Ωστόσο, το αποτέλεσμα ενός τέτοιου διαγωνισμού ταχύτητας μεταξύ των μεταβάσεων θα είναι σωστό μόνο όταν οι αντίστοιχες μεταβάσεις παρουσιάζουν διαφορετικούς χρονομετρητές (στο τρέχον παράδειγμα, η τιμή d_1 θα πρέπει να είναι διαφορετική της τιμής d_2).

Διαφορετικά, πρέπει να υιοθετηθεί μια **επιλογή ανεξάρτητη των χρονικών προδιαγραφών (time-independent selection)**, η οποία και θα περιορίσει περισσότερο τη μη-ντετερμινιστική συμπεριφορά του δικτύου. Τότε ορίζουμε ότι

ακολουθείται μια πολιτική προεπιλογής, σύμφωνα με την οποία κάθε κατάσταση σύγκρουσης επιλύεται με βάση μια “εξωτερική”, χρονικά ανεξάρτητη επιλογή, καταχωρώντας δηλαδή διαφορετικές προτεραιότητες στις συγκρουόμενες μεταβάσεις. Ο χρονομετρητής της μετάβασης που παρουσιάζει τη μεγαλύτερη κάθε φορά προτεραιότητα θα είναι εκείνος που θα αρχίσει να μειώνεται. Στο Σχήμα 6, εάν υποθέσουμε ότι οι συγκρουόμενες μεταβάσεις είναι άμεσες, δηλαδή όταν οι τιμές d_1 και d_2 είναι ίσες με μηδέν, μόνο μέσω μιας πολιτικής προεπιλογής μπορεί να αποφασιστεί, ποιά μετάβαση θα αποκτήσει πλεονέκτημα έναντι της άλλης σε μια κατάσταση σύγκρουσης, ώστε να πυροδοτήσει πρώτη.



Σχήμα 6: Χροнисμένο δίκτυο για την αναπαράσταση μιας δομής σύγκρουσης

3.4.2 Χρονικά Δίκτυα Petri

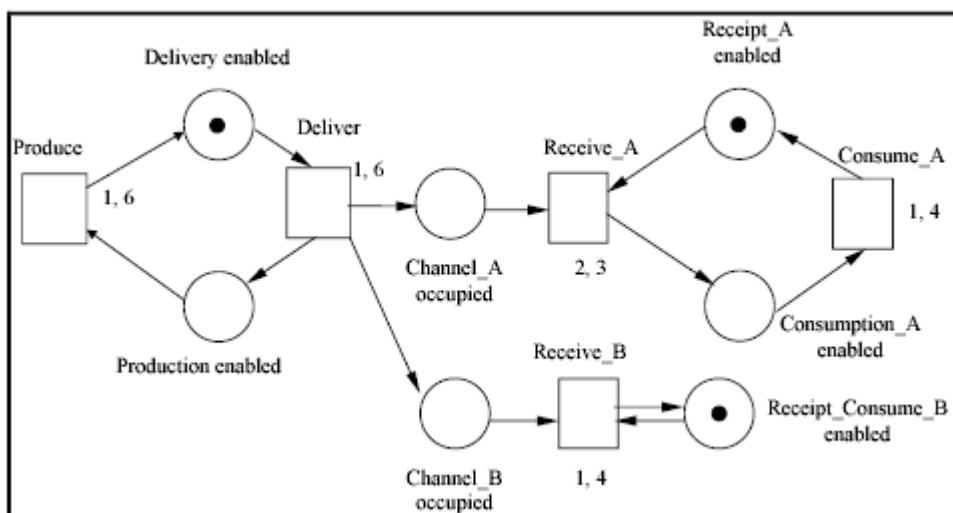
Τα Χρονικά δίκτυα χρησιμοποιούν χρονικά διαστήματα, προκειμένου να εκφράσουν τις χρονικές προδιαγραφές των μεταβάσεων. Έτσι, αντί για σταθερές χρονικές καθυστερήσεις, χρησιμοποιούνται **ελάχιστα (lower time bounds)** και **ανώτατα χρονικά όρια (upper time bounds)**. Οι χρονικές προδιαγραφές δηλαδή δίνονται ως ζεύγη χρονικών τιμών.

Τα Χρονικά δίκτυα έχουν εφαρμοστεί στην περιοχή των εφαρμογών πραγματικού χρόνου για την ανάλυση προδιαγραφών που αφορούν θέματα ασφάλειας (safety specifications) και χρονικών προθεσμιών. Το μοντέλο των Χρονικών δικτύων έχει χρησιμοποιηθεί στην ανίχνευση σφαλμάτων σε συστήματα που εμφανίζουν απαιτήσεις κρίσιμες σε ασφάλεια (**safety critical systems**) και στην επαλήθευση των απαιτήσεων ασφάλειας συστημάτων (**system safety analysis**).

Η πυροδότηση μιας μετάβασης t εξαρτάται από ένα ελάχιστο όριο, TC_{min} , και από ένα ανώτατο όριο, TC_{max} , χρονικών τιμών. Η t είναι ενεργοποιημένη τη χρονική στιγμή T_0 , εάν τότε υπάρχει επαρκής αριθμός από μάρκες στις θέσεις εισόδου και εάν το ελάχιστο χρονικό όριό της δεν είναι μεγαλύτερο από το ανώτατο όριο οποιασδήποτε άλλης ενεργοποιημένης μετάβασης. Έτσι η ενεργοποιημένη μετάβαση t δεν μπορεί να πυροδοτήσει σε χρονικές στιγμές πριν από τη χρονική στιγμή $T_0 + TC_{min}$ ή μετά από τη χρονική στιγμή $T_0 + TC_{max}$. Κατά συνέπεια, το ελάχιστο και το ανώτατο χρονικό όριο της t θεωρούνται **σχετικά (relative)** με τη χρονική στιγμή T_0 (τη στιγμή ενεργοποίησης της t). Τα ελάχιστα και τα ανώτατα χρονικά όρια τιμών δίνονται συνήθως στη μορφή ρητών αριθμών.

Στα Χρονικά δίκτυα υιοθετείται συνήθως (όπως και στα Χρονισμένα δίκτυα) ο ισχυρός κανόνας πυροδότησης. Εάν η πυροδότηση μιας μετάβασης t δεν λάβει χώρα κατά τη χρονική περίοδο που μεσολαβεί από τη χρονική στιγμή T_0+TC_{min} έως τη χρονική στιγμή T_0+TC_{max} , τότε η t πρέπει οπωσδήποτε να πυροδοτήσει τη στιγμή T_0+TC_{max} . Επιπλέον, τα ελάχιστα και ανώτατα όρια κάθε επίμονης μετάβασης, δηλαδή μιας μετάβασης που παραμένει ενεργοποιημένη, τόσο πριν, όσο και μετά, την πυροδότηση μιας άλλης μετάβασης, **μετατίθενται προς τα αριστερά (left shift)**, ώστε να ληφθεί υπόψη ο χρόνος εκτέλεσης της αντίστοιχης δράσης που αναπαρίσταται από τη μετάβαση που πυροδοτήθηκε. Ως **σύνολο διαστημάτων πυροδοτήσεων (firing interval set)** ορίζεται ένα διάστημα που αλλάζει δυναμικά κατά την εκτέλεση του δικτύου και αποτελείται από χρονικά διαστήματα που περιλαμβάνουν όλες τις πιθανές χρονικές στιγμές πυροδότησης μεταβάσεων.

Στο Σχήμα 7 δίνεται ένα Χρονικό δίκτυο όπου διακρίνουμε μία διεργασία-παραγωγό και δύο διεργασίες-καταναλωτές. Στο δίκτυο περιγράφεται αναλυτικά ο τρόπος με τον οποίο η διεργασία A καταναλώνει μάρκες (μηνύματα). Αντίθετα η διεργασία B λαμβάνει και καταναλώνει μηνύματα με τρόπο αδιαίρετο, αφού η θέση Receipt_Consume_B δεν έχει υποστεί αποσύνθεση. Στο αρχικό μαρκάρισμα υπάρχει από μια μάρκα στις Delivery enabled, Receipt_A enabled και Receipt_Consume_B enabled. Η Deliver είναι η μόνη ενεργοποιημένη μετάβαση, και συνεπώς, το σύνολο των διαστημάτων πυροδοτήσεων περιλαμβάνει μόνο το διάστημα (1, 6). Αυτό σημαίνει ότι η Deliver μπορεί να πυροδοτήσει σε οποιαδήποτε χρονική στιγμή t_1 ανάμεσα στη χρονική στιγμή 1 και τη στιγμή 6. Η πυροδότηση της Deliver οδηγεί σε ένα νέο μαρκάρισμα, όπου καθεμιά από τις θέσεις Production enabled, Channel_A occupied, Channel_B occupied, Receipt_A enabled και Receipt_Consume_B enabled διατηρεί μια μάρκα. Οι ενεργοποιημένες μεταβάσεις είναι τώρα η Produce, η Receive_A και η Receive_B και το σύνολο των διαστημάτων πυροδοτήσεων περιλαμβάνει τα διαστήματα (1, 6), (2, 3) και (1, 4). Η πυροδότηση της Produce επιτρέπεται μεταξύ της χρονικής στιγμής 1 (η ελάχιστη τιμή στο διάστημα (1, 6)) και της χρονικής στιγμής 3 (η ελάχιστη τιμή όλων των ανώτατων χρονικών ορίων). Η πυροδότηση της Produce, σε οποιαδήποτε χρονική στιγμή t_2 στο διάστημα (1, 3), οδηγεί σε ένα νέο μαρκάρισμα, όπου καθεμιά από τις θέσεις Delivery enabled, Channel_A occupied, Channel_B occupied, Receipt_A enabled και Receipt_Consume_B enabled, διατηρεί μια μάρκα. Τώρα το σύνολο των διαστημάτων πυροδοτήσεων περιλαμβάνει τα διαστήματα (1, 6), $(\max(0, 2-t_2), 3-t_2)$ και $(\max(0, 1-t_2), 4-t_2)$. Θα πρέπει να παρατηρηθεί ότι οι μεταβάσεις Receive_A και Receive_B είναι εκ νέου ενεργοποιημένες (είναι επίμονες μεταβάσεις), και, για το λόγο αυτό, τα αντίστοιχα ελάχιστα και ανώτατα χρονικά όρια μετατίθενται αριστερά.



Σχήμα 7: Χρονικό δίκτυο - Σύστημα παραγωγών-καταναλωτών

3.4.3 Μνήμη

Στα δίκτυα με χρονισμένες μεταβάσεις, ένα σημαντικό θέμα που προκύπτει σε κάθε πυροδότηση μετάβασης είναι ο τρόπος διαχείρισης των χρονομετρητών των μεταβάσεων που δεν πυροδοτούν.

Από την οπτική γωνία της μοντελοποίησης οι διαφορετικές πολιτικές που μπορούν να υιοθετηθούν, συνδέουν την περασμένη ιστορία των συστημάτων με τη μελλοντική τους εξέλιξη, μέσω διαφόρων τρόπων διατήρησης της μνήμης του χρόνου που έχει ήδη δαπανηθεί στις δράσεις. Η **πολιτική μνήμης (memory policy)** των μεταβάσεων ορίζει τον τρόπο με τον οποίο ρυθμίζονται οι χρονομετρητές των μεταβάσεων όταν συμβαίνει μια αλλαγή κατάστασης, η οποία πιθανώς τροποποιεί την ενεργοποίηση των μεταβάσεων. Υπάρχουν δύο βασικοί μηχανισμοί για τη ρύθμιση των χρονομετρητών των χρονισμένων μεταβάσεων που είναι οι εξής :

- (i) **Συνέχεια (Continue)** : Ο χρονομετρητής που συνδέεται με τη μετάβαση κρατάει την τρέχουσα τιμή του και θα συνεχίσει την αντίστροφη μέτρηση αργότερα.
- (ii) **Επανεκκίνηση (Restart)** : Ο χρονομετρητής που συνδέεται με τη μετάβαση επανεκκινείται, δηλαδή η τρέχουσα τιμή του απορρίπτεται και μια νέα τιμή θα παραχθεί όταν χρειαστεί.

Συνδέοντας με διαφορετικό τρόπο τους δύο βασικούς μηχανισμούς, της συνέχειας και της επανεκκίνησης, δημιουργούνται διαφορετικοί τρόποι παρακολούθησης του παρελθόντος που μοντελοποιούν τις διαφορετικές συμπεριφορές που προκύπτουν στα πραγματικά συστήματα. Υπάρχουν τρεις εναλλακτικοί τρόποι :

- (i) **Επαναδειγματοληψία (Resampling)** : Σε κάθε πυροδότηση μετάβασης, οι χρονομετρητές όλων των χρονισμένων μεταβάσεων απορρίπτονται. Δεν καταγράφεται μνήμη από το παρελθόν. Μετά την απόρριψη όλων των

χρονομετρητών, ορίζονται νέες τιμές για τους χρονομετρητές των μεταβάσεων που είναι ενεργοποιημένες στο νέο μαρκάρισμα.

- (ii) **Μνήμη ενεργοποίησης (Enabling memory)** : Σε κάθε πυροδότηση μετάβασης οι χρονομετρητές όλων των χρονισμένων μεταβάσεων που είναι **απενεργοποιημένες (disabled)**, επανεκκινούνται, ενώ οι χρονομετρητές όλων των μεταβάσεων που είναι ενεργοποιημένες, κρατούν τις τρέχουσες τιμές τους. Η μνήμη του παρελθόντος καταγράφεται από τη **μεταβλητή της μνήμης ενεργοποίησης (enabling memory variable)** που σχετίζεται με κάθε μετάβαση. Η μεταβλητή της μνήμης ενεργοποίησης υπολογίζει το εκτελέσιμο έργο της δράσης που σχετίζεται με τη μετάβαση, από τη στιγμή που τελευταία ρυθμίστηκε ο χρονομετρητής της. Με άλλα λόγια, η μεταβλητή της μνήμης ενεργοποίησης μετράει το χρόνο ενεργοποίησης της μετάβασης από τη στιγμή που ενεργοποιήθηκε για τελευταία φορά.
- (iii) **Μνήμη ηλικίας (Age memory)** : Σε κάθε πυροδότηση μετάβασης, οι χρονομετρητές όλων των χρονισμένων μεταβάσεων κρατούν τις τρέχουσες τιμές τους. Η μνήμη του παρελθόντος καταγράφεται από τη **μεταβλητή της μνήμης ηλικίας (age memory variable)** που σχετίζεται με κάθε χρονισμένη μετάβαση. Η μεταβλητή της μνήμης ηλικίας υπολογίζει το εκτελέσιμο έργο της δράσης που σχετίζεται με τη μετάβαση, από τη στιγμή της τελευταίας της πυροδότησης. Με άλλα λόγια, η μεταβλητή της μνήμης ηλικίας μετράει το **αθροιστικό ή συσσωρευτικό (cumulative)** χρόνο ενεργοποίησης της μετάβασης από τη στιγμή που πυροδοτήθηκε για τελευταία φορά.

Ο πίνακας του Σχήματος 8 περιέχει όλους τους πιθανούς συνδυασμούς μνήμης για τις μεταβάσεις που παραμένουν ενεργοποιημένες και για αυτές που απενεργοποιούνται σε ένα νέο μαρκάρισμα.

	για τη μετάβαση που παραμένει ενεργοποιημένη επανεκκίνηση	για τη μετάβαση που απενεργοποιείται επανεκκίνηση
Επαναδειγματοληψία (Resampling)		
Μνήμη ενεργοποίησης (Enabling memory)	συνέχεια	επανεκκίνηση
Μνήμη ηλικίας (Age memory)	συνέχεια	συνέχεια

Σχήμα 8: Μηχανισμοί μνήμης

Οι τρεις πολιτικές μνήμης μπορούν να χρησιμοποιηθούν στα μοντέλα των δικτύων Petri με χρονισμένες μεταβάσεις για διαφορετικούς σκοπούς μοντελοποίησης. Στην πρώτη περίπτωση, της επαναδειγματοληψίας, το εκτελέσιμο έργο των δράσεων που σχετίζονται με τις μεταβάσεις, οι οποίες δεν πυροδοτούν, χάνεται. Αυτό μπορεί να είναι επαρκές για τη μοντελοποίηση, παραδείγματος χάριν ανταγωνιζόμενων δράσεων. Η πρακτική χρήση αυτής της πολιτικής είναι πολύ περιορισμένη, αλλά πρέπει να ληφθεί υπ'όψη, λόγω της θεωρητικής της σημασίας στην περίπτωση των στοχαστικών και των γενικευμένων στοχαστικών δικτύων Petri που θα περιγραφούν στη συνέχεια.

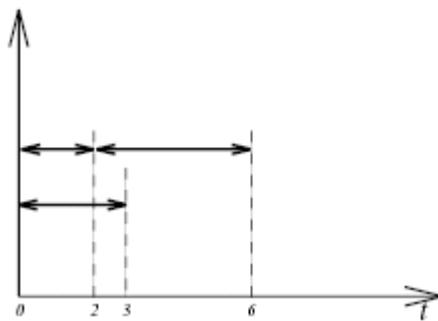
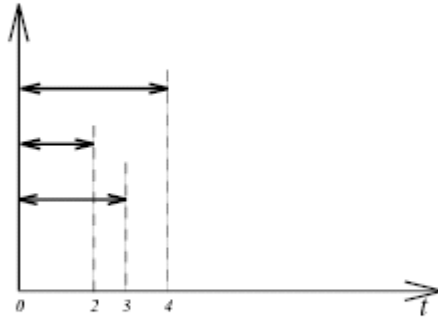
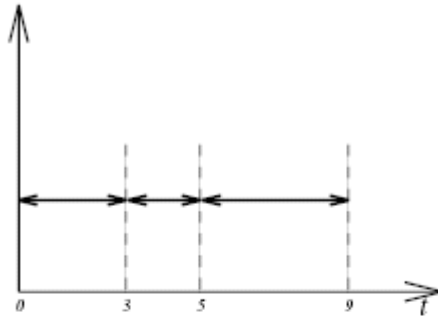
Οι άλλες δύο πολιτικές είναι μεγαλύτερης σημασίας από την οπτική γωνία των εφαρμογών. Μπορούν να συνυπάρχουν μέσα στο ίδιο χρονισμένο μοντέλο δικτύου Petri, λόγω των διαφορετικών στοιχείων που μπορούν να ανατεθούν στις μεταβάσεις του μοντέλου.

3.4.4 Πολλαπλή Ενεργοποίηση (Multiple Enabling)

Στα δίκτυα Petri με χρονισμένες μεταβάσεις δίνεται ιδιαίτερη έμφαση στον ορισμό του **βαθμού ενεργοποίησης για κάθε μετάβαση (degree of transition enabling)**, το μέγιστο δηλαδή αριθμό των ανεξάρτητων μεταξύ τους περιπτώσεων, στις οποίες μια μετάβαση είναι ενεργοποιημένη. Ο βαθμός αυτός πρέπει να προσδιοριστεί όταν επιτρέπονται περισσότερες από μία στον αριθμό μάρκες σε κάθε θέση εισόδου μιας μετάβασης, και καθορίζει τον αριθμό των φορών που μια μετάβαση είναι ενεργοποιημένη, ανεξάρτητα και ταυτόχρονα με άλλες μεταβάσεις του δικτύου. Με στόχο λοιπόν να επιλυθούν πιθανές περιπτώσεις ασάφειας στη σημασιολογία της εκτέλεσης ενός δικτύου λαμβάνονται υπόψη σχετικές έννοιες από το χώρο της θεωρίας των **Δικτύων Ουρών (Queuing Networks)** και προτείνονται τρεις εναλλακτικοί τρόποι για τον καθορισμό του βαθμού ενεργοποίησης :

- (i) η θεώρηση του **απλού εξυπηρετητή (single server)** : μια καθυστέρηση πυροδότησης ορίζεται, όταν η μετάβαση ενεργοποιείται για πρώτη φορά, και αν μετά από την πυροδότηση της μετάβασης, η μετάβαση είναι ακόμα ενεργοποιημένη στο νέο μαρκάρισμα, ορίζεται νέα καθυστέρηση.
- (ii) η θεώρηση του **άπειρου εξυπηρετητή (infinite server)** : κάθε ενεργοποιημένο σύνολο από μάρκες επεξεργάζεται, μόλις σχηματιστεί μέσα στις θέσεις εισόδου της χρονισμένης μετάβασης. Η αντίστοιχη καθυστέρηση μετάβασης του ορίζεται εκείνη τη στιγμή και οι χρονομετρητές που σχετίζονται με όλα αυτά τα ενεργοποιημένα σύνολα μετράνε αντίστροφα μέχρι το μηδέν, παράλληλα.
- (iii) η θεώρηση του **πολλαπλού εξυπηρετητή (multiple server)** : τα ενεργοποιημένα σύνολα από μάρκες επεξεργάζονται, μόλις σχηματιστούν μέσα στις θέσεις εισόδου της μετάβασης, μέχρι ένα μέγιστο βαθμό παραλληλισμού k . Για μεγαλύτερες τιμές του βαθμού ενεργοποίησης, οι χρονομετρητές που σχετίζονται με τα νέα ενεργοποιημένα σύνολα από μάρκες, ορίζονται μόνο όταν ο αριθμός των ταυτόχρονων χρονομετρητών που τρέχουν γίνεται μικρότερος από την τιμή k .

Παρατίθεται ένα απλό παράδειγμα για την καλύτερη κατανόηση των τριών εναλλακτικών τρόπων. Έστω μια χρονισμένη μετάβαση T με βαθμό ενεργοποίησης 3 κι ότι το δίκτυο αρχίζει να λειτουργεί τη χρονική στιγμή 0 με αυτό το βαθμό ενεργοποίησης. Οι τρεις ενεργοποιήσεις σχετίζονται με τρεις δράσεις, των οποίων οι χρονικές διάρκειες είναι 3, 2 και 4 μονάδες αντίστοιχα. Το Σχήμα 9 αναπαριστά τη συμπεριφορά του δικτύου σε σχέση με το χρόνο, σύμφωνα με τις τρεις παραπάνω θεωρήσεις.



Σχήμα 9: Συμπεριφορά του δικτύου σε σχέση με το χρόνο

- (i) η θεώρηση του απλού εξυπηρετητή : η σειριακή εκτέλεση των δράσεων προκαλεί την ακόλουθη σειρά γεγονότων :
- $t = 0$: η T είναι ενεργοποιημένη και η πρώτη δράση ξεκινάει.
 - $t = 3$: η πρώτη δράση, διάρκειας 3, τελειώνει, η T πυροδοτεί και η δεύτερη δράση ξεκινάει.
 - $t = 5$: η δεύτερη δράση, διάρκειας 2, τελειώνει, η T πυροδοτεί και η τρίτη δράση, διάρκειας 4, τελειώνει και η T απενεργοποιείται.
- (ii) η θεώρηση του άπειρου εξυπηρετητή : η παράλληλη εκτέλεση των δράσεων προκαλεί την ακόλουθη σειρά γεγονότων :
- $t = 0$: η T είναι ενεργοποιημένη και όλες οι δράσεις ξεκινούν.
 - $t = 2$: η T πυροδοτεί λόγω της ολοκλήρωσης της δεύτερης δράσης, διάρκειας 2.
 - $t = 3$: η T πυροδοτεί λόγω της ολοκλήρωσης της πρώτης δράσης, διάρκειας 3.
 - $t = 4$: η T πυροδοτεί λόγω της ολοκλήρωσης της τρίτης δράσης, διάρκειας 4 και απενεργοποιείται.

- (iii) η θεώρηση του πολλαπλού εξυπηρετητή : σε αυτή την περίπτωση υποθέτουμε ότι ο μέγιστος βαθμός παραλληλισμού είναι $k = 2$. Αυτό προκαλεί την ακόλουθη σειρά γεγονότων :
- $t = 0$: η T είναι ενεργοποιημένη και οι δύο πρώτες ξεκινούν.
 - $t = 2$: η T πυροδοτεί λόγω της ολοκλήρωσης της δεύτερης δράσης, διάρκειας 2, κι έτσι η τρίτη δράση μπορεί να ξεκινήσει.
 - $t = 3$: η T πυροδοτεί λόγω της ολοκλήρωσης της πρώτης δράσης, διάρκειας 3.
 - $t = 6$: η T πυροδοτεί λόγω της ολοκλήρωσης της τρίτης δράσης, διάρκειας 4 και απενεργοποιείται.

Η εισαγωγή αυτών των διαφορετικών θεωρήσεων πυροδότησης επιτρέπει τον ορισμό μοντέλων δικτύων Petri που είναι γραφικά απλά χωρίς να χάνουν κανένα από τα χαρακτηριστικά που επιτρέπουν την ανάλυση της συμπεριφοράς τους.

3.5 Δίκτυα με Χρονισμένες Θέσεις

Το μοντέλο των **δικτύων Χρονισμένων Θέσεων (Timed Place PNs)** συσχετίζει χρονικές καθυστερήσεις ή χρονικά διαστήματα με τις θέσεις του δικτύου. Το μοντέλο έχει εφαρμοστεί στην αναπαράσταση εφαρμογών με απαιτήσεις πραγματικού χρόνου. Για παράδειγμα, έχει χρησιμοποιηθεί για την περιγραφή συστημάτων των οποίων η συμπεριφορά καθορίζεται με βάση ένα **εξωτερικό χρονικό μηχανισμό-ρολόι (master timing mechanism)**. Τα συστήματα αυτά είναι γνωστά ως συστήματα που καθορίζουν τη συμπεριφορά τους με βάση το χρόνο (**time-driven systems**).

Στα δίκτυα Χρονισμένων Θέσεων οι χρονικές προδιαγραφές ορίζουν **χρόνους κατοχής (holding times)** των μαρκών από τις θέσεις. Αντιθέτως, στα δίκτυα χρονισμένων μεταβάσεων είδαμε ότι οι χρονικές προδιαγραφές ορίζουν χρόνους πυροδότησης ή χρόνους ενεργοποίησης για τις μεταβάσεις. Το γεγονός της εμφάνισης μαρκών σε μια χρονισμένη θέση δεν σημαίνει υποχρεωτικά και την πυροδότηση μιας μετάβασης εξόδου. Οι μάρκες καθίστανται **διαθέσιμες (available)** μόνο μετά το πέρας της χρονικής καθυστέρησης που προσδιορίζει τη χρονισμένη θέση. Πριν από αυτή τη χρονική στιγμή οι μάρκες είναι **μη διαθέσιμες (unavailable)**. Μετά τη χρονική στιγμή αυτή, οι μάρκες μπορούν να χρησιμοποιηθούν για την πυροδότηση κάθε μετάβασης εξόδου της θέσης, η οποία συνήθως λαμβάνει χώρα σύμφωνα με τον ισχυρό κανόνα πυροδότησης.

Η χρονική περίοδος κατά την οποία οι μάρκες σε μια θέση είναι μη διαθέσιμες μπορεί να ερμηνευτεί ως η χρονική διάρκεια εκτέλεσης μιας αντίστοιχης δράσης στην υπό σχεδίαση εφαρμογή. Έτσι, οι θέσεις του δικτύου μπορούν να χρησιμοποιηθούν για να περιγράψουν δράσεις, ενώ οι μεταβάσεις του δικτύου για να περιγράφουν **“στιγμιαία” γεγονότα (instantaneous events)**, που σηματοδοτούν την έναρξη και το πέρας μιας δράσης. Αν και η ερμηνεία αυτή έρχεται σε αντίθεση με τη θεμελιώδη υπόθεση της θεωρίας των δικτύων Petri, σύμφωνα με την οποία οι μεταβάσεις θεωρούνται ενεργητικές οντότητες και οι θέσεις θεωρούνται παθητικές οντότητες, το μοντέλο των δικτύων Χρονισμένων Θέσεων επιτυγχάνει με την ερμηνεία αυτή να διατηρήσει την ιδιότητα της αμεσότητας για την πυροδότηση κάθε μετάβασης (η χρονική διάρκεια της πυροδότησης κάθε μετάβασης θεωρείται

μηδενική). Επιπλέον, διατηρείται η βασική υπόθεση των μη-χρονικών δικτύων που αφορά στην επίλυση των συγκρούσεων. Αυτές επιλύονται όχι με βάση τις χρονικές προδιαγραφές, όπως συμβαίνει στην περίπτωση των δικτύων με χρονισμένες μεταβάσεις, αλλά με μη-ντετερμινιστικό τρόπο. Τέλος, η δυναμική συμπεριφορά του μοντέλου βασίζεται στην υπόθεση ότι η χρονική αναφορά είναι σχετική.

3.6 Στοχαστικές Εκδόσεις

Στην κατηγορία των στοχαστικών εκδόσεων (stochastic versions) των δικτύων Petri περιλαμβάνονται επεκτάσεις που βασίζονται κυρίως στα Χρονισμένα δίκτυα. Στα στοχαστικά μοντέλα προτείνεται η χρησιμοποίηση τυχαίων μεταβλητών χρονικών καθυστερήσεων, με σκοπό την περιγραφή των δράσεων μιας εφαρμογής που εμφανίζουν ασταθή διάρκεια εκτέλεσης. Κάτω από αυτή τη θεώρηση, οι στοχαστικές εκδόσεις των δικτύων Petri παρουσιάζονται ισχυρότερες σε ό,τι αφορά στη δυνατότητα περιγραφής, συγκριτικά με μοντέλα που εμφανίζουν ντετερμινιστικά χρονικά κατηγορήματα, δηλαδή τα δίκτυα Χρονισμένων Θέσεων και Μεταβάσεων. Η ιδέα της εισαγωγής τυχαίων χρονικών καθυστερήσεων στα δίκτυα Petri μελετήθηκε για πρώτη φορά από τον Natkin το 1980 και ξεχωριστά από τον Molloy το 1981.

Αντιπροσωπευτικές στοχαστικές εκδόσεις των δικτύων Petri είναι : τα **Στοχαστικά δίκτυα Petri (Stochastic Petri nets - SPNs)**, τα **Γενικευμένα Στοχαστικά δίκτυα Petri (Generalized Stochastic Petri nets - GSPNs)**, τα **Ημι-μαρκοβιανά Στοχαστικά δίκτυα Petri (Semi-Markov Stochastic Petri nets - Semi-Markov SPNs)**, τα **Στοχαστικά δίκτυα Petri τύπου Φάσης (Phase Type Stochastic Petri nets - PHSPNs)** , τα **Ντετερμινιστικά Στοχαστικά δίκτυα Petri (Deterministic Stochastic Petri nets - DSPNs)** και τα **Στοχαστικά δίκτυα Petri Αναγέννησης Markov (Markov Regenerative Stochastic Petri nets - MRSPNs)**.

Ένα στοχαστικό μοντέλο προτείνει μια συγκεκριμένη κατανομή για τις χρονικές καθυστερήσεις των μεταβάσεων. Οι χρονικές καθυστερήσεις των μεταβάσεων περιγράφονται με κατάλληλες **συναρτήσεις πυκνότητας πιθανότητας (probability density functions)**. Διακρίνουμε δύο βασικά μοντέλα: τα Στοχαστικά δίκτυα Petri και τα Γενικευμένα Στοχαστικά δίκτυα Petri, τα οποία θα αναλυθούν εκτενέστερα παρακάτω. Τα δύο μοντέλα έχουν εφαρμοστεί ευρύτατα για την ανάλυση της απόδοσης σε συστήματα με χρονικές απαιτήσεις. Συγκεκριμένα, τα μοντέλα προσφέρουν στο σχεδιαστή τη δυνατότητα να υπολογίσει ορισμένες πολύ χρήσιμες μετρικές, όπως για παράδειγμα είναι ο μέσος αριθμός μαρκών σε κάθε θέση, ο μέσος ρυθμός πυροδότησης κάθε μετάβασης και το ελάχιστο χρονικό διάστημα πριν από το οποίο ολοκληρώνεται μια ακολουθία πυροδοτήσεων που οδηγεί στο αρχικό μαρκάρισμα, αφού πυροδοτήσει κάθε μετάβαση στην ακολουθία, τουλάχιστον για μια φορά.

3.6.1 Στοχαστικά Δίκτυα Petri

Τα Στοχαστικά δίκτυα Petri είναι δίκτυα Petri με χρονισμένες μεταβάσεις και ατομικό κανόνα εκτέλεσης-πυροδότησης, στα οποία οι **καθυστερήσεις πυροδότησης των μεταβάσεων (transition firing delays)** είναι **εκθετικά κατανεμημένες τυχαίες μεταβλητές (exponentially distributed random variables)** : κάθε μετάβαση

συσχετίζεται με μια τυχαία μεταβλητή, της οποίας η συνάρτηση πυκνότητας πιθανότητας είναι αρνητικά εκθετική, και η οποία εκφράζει την χρονική καθυστέρηση της μετάβασης από τη στιγμή της ενεργοποίησης της μέχρι τη στιγμή της πυροδότησης της. Δηλαδή, κάθε μετάβαση t στο Στοχαστικό δίκτυο Petri συσχετίζεται με ένα **ρυθμό πυροδότησης (firing rate) $w(t)$** , που είναι εκθετικά καταναμημένος.

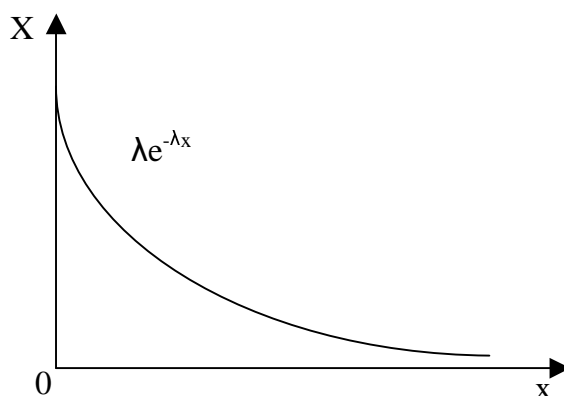
Η εκθετική κατανομή $f_X(x)$ ορίζεται ως

$$f_X(x) = \lambda e^{-\lambda x}$$

και εξαρτάται μόνο από την τιμή του ρυθμού της, λ , που είναι ο αντίστροφος της μέσης τιμής της :

$$E[X] = 1/\lambda.$$

Η γραφική αναπαράσταση της εκθετικής κατανομής φαίνεται στο Σχήμα 10.



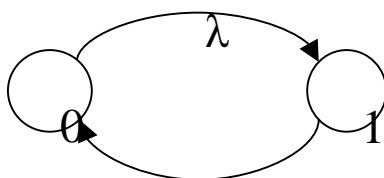
Σχήμα 10: Εκθετική κατανομή

Η εκθετική κατανομή είναι η μόνη συνάρτηση κατανομής πιθανότητας συνεχούς χρόνου, η οποία έχει τη **χωρίς μνήμη ιδιότητα (memoryless property)**. Η χρήση εκθετικών κατανομών για τον ορισμό των χρονικών προδιαγραφών είναι ιδιαίτερα ελκυστική, γιατί τα στοχαστικά δίκτυα Petri μπορούν να μετασχηματιστούν σε **αλυσίδες Markov συνεχούς χρόνου (continuous time Markov chains)**.

Μια αλυσίδα Markov είναι μια **Μαρκοβιανή διαδικασία (Markov process)** διακεκριμένων καταστάσεων. Οι Μαρκοβιανές διαδικασίες αποτελούν μια κατηγορία των στοχαστικών διαδικασιών, όπου **στοχαστικές διαδικασίες (stochastic processes)** ονομάζονται οι τυχαίες συναρτήσεις του χρόνου. Ο όρος “διακεκριμένων καταστάσεων” αναφέρεται στον αριθμό των δυνατών τιμών της διαδικασίας ο οποίος είναι πεπερασμένος. Στις Μαρκοβιανές διαδικασίες, οι μελλοντικές καταστάσεις είναι ανεξάρτητες του παρελθόντος και εξαρτώνται μόνο από το παρόν (χωρίς μνήμη ιδιότητα). Για να προβλεφθεί το μέλλον μιας Μαρκοβιανής διαδικασίας συνεχούς χρόνου χρειάζεται να γνωρίζουμε την τρέχουσα κατάσταση κι όχι πόσο έμεινε η διαδικασία στην τρέχουσα κατάσταση.

Μια αλυσίδα Markov συνεχούς χρόνου περιγράφεται από το **διάγραμμα ρυθμού μετάβασης-κατάστασης (state transition rate diagram)**. Το διάγραμμα αυτό είναι

ένας κατευθυνόμενος γράφος που αποτελείται από κόμβους και τόξα. Κάθε κόμβος έχει ως επιγραφή μία **κατάσταση** (state) της αλυσίδας Markov και κάθε τόξο έχει ως επιγραφή το ρυθμό της εκθετικής κατανομής που σχετίζεται με τη μετάβαση από μία κατάσταση σε κάποια άλλη. Ο ρυθμός αυτός ονομάζεται **ρυθμός μετάβασης-κατάστασης** (state transition rate) και είναι ο χρόνος που απαιτείται για να μεταβεί η αλυσίδα από μία κατάσταση σε κάποια άλλη. Στο Σχήμα 11 απεικονίζεται το διάγραμμα ρυθμού μετάβασης-κατάστασης μιας αλυσίδας Markov συνεχούς χρόνου με δύο καταστάσεις, τις 0 και 1 και ρυθμούς μετάβασης, λ και μ , αντίστοιχα.

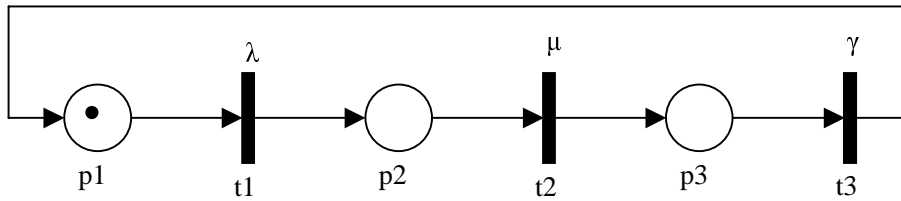


Σχήμα 11: Διάγραμμα ρυθμού μετάβασης αλυσίδας Markov

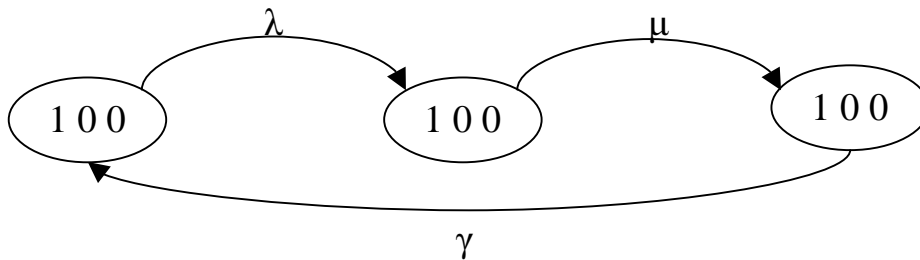
Λόγω της χωρίς μνήμη ιδιότητας της εκθετικής κατανομής, ένα Στοχαστικό δίκτυο Petri είναι ισομορφικό με μια αλυσίδα Markov συνεχούς χρόνου και ο γράφος προσιτότητας του Στοχαστικού δικτύου Petri αντιστοιχεί στο διάγραμμα ρυθμού μετάβασης της αλυσίδας Markov.

Αυτό μπορεί εύκολα να φανεί, όταν η δομή του Στοχαστικού δικτύου Petri ικανοποιεί ταυτόχρονα και τη δομή της μηχανής κατάστασης (κάθε μετάβαση έχει μόνο μία θέση εισόδου και μόνο μία θέση εξόδου) και τη δομή του μαρκκαρισμένου γράφου (κάθε θέση έχει μόνο μία μετάβαση εισόδου και μόνο μία μετάβαση εξόδου) και υπάρχει μόνο μία μάρκα στο αρχικό μαρκκάρισμα. Σε αυτήν την περίπτωση κάθε θέση του δικτύου ορίζει μονοσήμαντα μια κατάσταση του μοντέλου και η θέση της μάρκας σε μια δεδομένη χρονική στιγμή καθορίζει την κατάσταση του μοντέλου σε αυτή τη χρονική στιγμή. Κάθε μαρκκάρισμα του δικτύου αντιστοιχεί σε μια κατάσταση της αλυσίδας Markov και κάθε μετάβαση αντιστοιχεί σε ένα τόξο που έχει ως επιγραφή το ρυθμό της αντίστοιχης κατανομής χρόνου πυροδότησης. Επιπλέον, όταν οι ρυθμοί πυροδότησης των μεταβάσεων έχουν αρνητικά εκθετικές κατανομές, ο χρόνος που δαπανάται σε κάθε θέση από το δίκτυο, ορίζεται εξ'ολοκλήρου από τη μοναδική μετάβαση που μπορεί να αποσύρει τη μάρκα από αυτή τη θέση.

Στο Σχήμα 12 φαίνεται ένα Στοχαστικό δίκτυο Petri που ικανοποιεί τη δομή της μηχανής κατάστασης και του μαρκκαρισμένου γράφου και στο Σχήμα 13, φαίνεται η αντίστοιχη αλυσίδα Markov συνεχούς χρόνου.



Σχήμα 12: Στοχαστικό δίκτυο Petri που ικανοποιεί τις δομές της μηχανής κατάστασης και του μαρκαρισμένου γράφου



Σχήμα 13: Η αντίστοιχη αλυσίδα Markov συνεχούς χρόνου του Στοχαστικού δικτύου Petri του προηγούμενου Σχήματος

Γενικά, η αλυσίδα Markov που σχετίζεται με ένα Στοχαστικό δίκτυο Petri εξάγεται από αυτό με την εφαρμογή των ακόλουθων απλών κανόνων :

1. Ο χώρος καταστάσεων της αλυσίδας Markov συνεχούς χρόνου, $S = \{s_i\}$, αντιστοιχεί στο σύνολο προσιτότητας του Στοχαστικού δικτύου Petri, δηλαδή ισχύει $M_i \leftrightarrow s_i$, όπου M_i ένα μαρκάρισμα του δικτύου Petri.
2. Ο ρυθμός μετάβασης-κατάστασης από την κατάσταση s_i , που αντιστοιχεί στο μαρκάρισμα M_i , στην κατάσταση s_j , που αντιστοιχεί στο μαρκάρισμα M_j , υπολογίζεται ως το άθροισμα των ρυθμών πυροδότησης $w(t)$ όλων των μεταβάσεων t που είναι ενεργοποιημένες στο M_i , και των οποίων οι πυροδοτήσεις παράγουν το μαρκάρισμα M_j .

Σύμφωνα με την πολιτική του διαγωνισμού ταχύτητας που ισχύει στα Στοχαστικά δίκτυα Petri, η μετάβαση με τη μικρότερη χρονική καθυστέρηση θα πυροδοτήσει πρώτη. Η χρονική καθυστέρηση που υπολογίζεται για κάθε μετάβαση σε ένα νέο μαρκάρισμα, είναι είτε μια νέα τιμή από την εκθετική κατανομή που σχετίζεται με τη μετάβαση (για τις μεταβάσεις που επανεκκινούνται), είτε ο υπόλοιπος χρόνος πυροδότησης του χρονομετρητή (για τις μεταβάσεις που συνεχίζουν). Όμως, λόγω της χωρίς μνήμη ιδιότητας της αρνητικά εκθετικής κατανομής, η κατανομή του συνολικού χρόνου πυροδότησης ταυτίζεται με τη κατανομή του χρόνου πυροδότησης που απομένει μετά από μια αλλαγή κατάστασης κι έτσι, δεν χρειάζεται να οριστεί κάποια πολιτική μνήμης (επαναδειγματοληψία, μνήμη ενεργοποίησης ή μνήμη ηλικίας), αφού δεν υπάρχει διαφορά μεταξύ τους για την εκθετική κατανομή.

Μία άλλη πολύ χρήσιμη ιδιότητα της εκθετικής κατανομής, είναι η συνέχεια της. Επειδή η εκθετική κατανομή είναι μια συνεχής συνάρτηση, η πιθανότητα να επιλεγεί τυχαία μια συγκεκριμένη τιμή x , είναι ίση με το μηδέν. Άρα η πιθανότητα, δύο

χρονομετρητές να εκπνέουν την ίδια χρονική στιγμή είναι και αυτή ίση με το μηδέν. Δηλαδή, δοθέντος μιας τιμής που έχει ληφθεί τυχαία για τον πρώτο χρονομετρητή, από μια αρνητικά εκθετική κατανομή, η πιθανότητα να ληφθεί τυχαία η ίδια τιμή και για τον δεύτερο χρονομετρητή ισούται με το μηδέν.

Ένα Στοχαστικό δίκτυο Petri είναι βασικά ένα δίκτυο Petri υψηλού επιπέδου, του οποίου η δυναμική συμπεριφορά περιγράφεται μέσω μιας στοχαστικής διαδικασίας. Η εκτίμηση της απόδοσης ενός συστήματος γίνεται απλά με τη μοντελοποίηση του δοθέντος συστήματος χρησιμοποιώντας Στοχαστικά δίκτυα Petri και με την εξαγωγή της στοχαστικής διαδικασίας που διέπει τη συμπεριφορά του συστήματος. Στη συνέχεια, αυτή η στοχαστική διαδικασία αναλύεται με τη χρήση των αλυσίδων Markov συνεχούς χρόνου.

3.6.2 Γενικευμένα Στοχαστικά Δίκτυα Petri

Ο βασικός παράγοντας που περιορίζει την εφαρμοσιμότητα των Στοχαστικών δικτύων Petri είναι η πολυπλοκότητα της ανάλυσης τους. Αυτό οφείλεται σε πολλούς παράγοντες, από τους οποίους ο πιο κρίσιμος είναι, ο πάρα πολύ μεγάλος αριθμός των προσιτών μαρκαρισμάτων. Ένας άλλος παράγοντας που κάνει πιο περίπλοκη την επίλυση του μοντέλου είναι, η παρουσία, σε ένα μόνο μοντέλο, δράσεων που λαμβάνουν χώρα σε μια πολύ πιο γρήγορη (ή αργή) χρονική κλίμακα, από αυτή που σχετίζεται με τα γεγονότα που παίζουν κρίσιμο ρόλο στη συνολική απόδοση. Αυτό έχει ως αποτέλεσμα τη δημιουργία συστημάτων που είναι δύσκολο να επιλυθούν με ένα αποδεκτό βαθμό ακρίβειας, χρησιμοποιώντας τις συνήθεις αριθμητικές τεχνικές. Από την άλλη μεριά, η παραμέληση των “γρήγορων” (ή των “αργών”) δράσεων μπορεί να οδηγήσει σε μοντέλα που είναι λογικά λάθος. Επίσης, μπορεί ο αναλυτής να εισάγει, στη δομή του μοντέλου του Στοχαστικού δικτύου Petri, μεταβάσεις που παριστάνουν καθαρά λογικές πλευρές της συμπεριφοράς του συστήματος και δεν μπορούν να συσχετιστούν με χρονικές προδιαγραφές.

Για να αντιμετωπιστούν αυτά τα προβλήματα προτάθηκαν τα Γενικευμένα Στοχαστικά δίκτυα Petri, τα οποία περιέχουν δύο τύπους μεταβάσεων :

- i. τις χρονισμένες μεταβάσεις, που χαρακτηρίζονται από τυχαίες, εκθετικά κατανομημένες χρονικές καθυστερήσεις πυροδότησης (όπως και στα Στοχαστικά δίκτυα Petri) και
- ii. τις άμεσες μεταβάσεις που εμφανίζουν μηδενική χρονική καθυστέρηση πυροδότησης.

Με τις άμεσες μεταβάσεις παριστάνονται στιγμιαίες δράσεις και συγκεκριμένα χαρακτηριστικά του μοντέλου, καθώς και διαφορές χρονικής κλίμακας μεταξύ των δράσεων. Οι άμεσες μεταβάσεις έχουν προτεραιότητα έναντι των χρονισμένων μεταβάσεων και παριστάνονται γραφικά όπως και στα δίκτυα Petri με χρονισμένες μεταβάσεις. Δηλαδή οι άμεσες μεταβάσεις παριστάνονται με μπάρες και οι χρονισμένες με ορθογώνια (λευκά ή μαύρα) κουτιά.

Στα Γενικευμένα Στοχαστικά δίκτυα Petri, η παράμετρος $w(t)$ έχει δύο ρόλους, ανάλογα με τον τύπο της μετάνασης. Δηλαδή η $w(t)$ είναι :

- η παράμετρος της αρνητικά εκθετικής συνάρτησης πυκνότητας πιθανότητας της καθυστέρησης πυροδότησης της μετάβασης, αν η t είναι χρονισμένη μετάβαση,
- το βάρος που χρησιμοποιείται για τον υπολογισμό των πιθανοτήτων πυροδότησης των άμεσων μεταβάσεων, αν η t είναι άμεση μετάβαση.

Η στοχαστική ερμηνεία του μοντέλου των Γενικευμένων Στοχαστικών δικτύων Petri είναι παρόμοια με αυτή του μοντέλου των Στοχαστικών δικτύων Petri, αλλά έχουν γίνει οι απαραίτητες αλλαγές για να συμπεριλάβει και τις άμεσες μεταβάσεις. Όταν παράγεται ένα νέο μαρκαρίσμα, πρέπει να εξακριβωθεί, αν ενεργοποιεί μόνο χρονισμένες μεταβάσεις ή αν ενεργοποιεί τουλάχιστον μία άμεση μετάβαση. Όπως προαναφέρθηκε στα Χρονισμένα δίκτυα Petri, τα μαρκαρίσματα του πρώτου τύπου ονομάζονται απτά, ενώ τα μαρκαρίσματα του δεύτερου τύπου ονομάζονται εξαφανιζόμενα.

Στην περίπτωση ενός απτού μαρκαρίσματος, ισχύει ότι και στα Στοχαστικά δίκτυα Petri. Όταν μια χρονισμένη μετάβαση ενεργοποιείται για πρώτη φορά μετά την τελευταία πυροδότηση, ο χρονομετρητής της λαμβάνει τυχαία μια νέα τιμή από την αρνητικά εκθετική κατανομή που σχετίζεται με τη μετάβαση. Στη συνέχεια, οι χρονομετρητές όλων των ενεργοποιημένων μεταβάσεων μειώνονται με ίση ταχύτητα, μέχρις ότου ένας από αυτούς να μηδενιστεί. Η μετάβαση, της οποίας ο χρονομετρητής μηδενίζεται πρώτος, πυροδοτεί.

Στην περίπτωση ενός εξαφανιζόμενου μαρκαρίσματος, το Γενικευμένο Στοχαστικό δίκτυο Petri δεν καταναλώνει χρόνο γιατί όλα είναι στιγμιαία. Αυτό σημαίνει ότι, αν μόνο μία άμεση μετάβαση είναι ενεργοποιημένη, πυροδοτεί και παράγεται ένα νέο μαρκαρίσμα. Αν περισσότερες από μία άμεσες μεταβάσεις είναι ενεργοποιημένες, χρειάζεται κάποιο κριτήριο για την επιλογή της μετάβασης που θα πυροδοτήσει. Η επιλογή αυτή δεν μπορεί να βασιστεί σε κάποια χρονική προδιαγραφή, εφ'όσον όλες οι άμεσες μεταβάσεις πυροδοτούν ακριβώς σε χρόνο μηδέν. Η επιλογή σε αυτή την περίπτωση βασίζεται σε βάρη ή σε προτεραιότητες. Πρώτα σχηματίζεται το σύνολο των μεταβάσεων που ανήκουν στο υψηλότερο επίπεδο προτεραιότητας και, αν περιλαμβάνει περισσότερες από μία μεταβάσεις, η περαιτέρω επιλογή, πιθανοτικής φύσης, βασίζεται στα βάρη των μεταβάσεων σύμφωνα με την έκφραση :

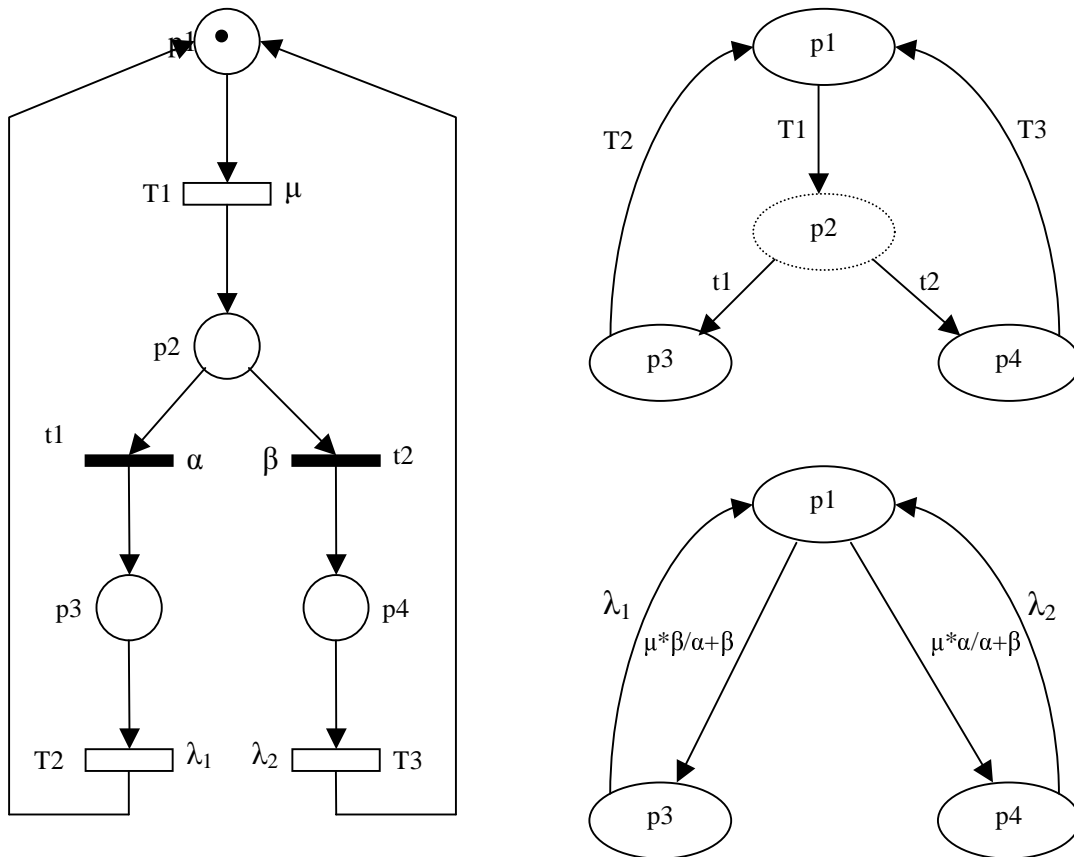
$$P\{t\} = w(t) / \sum_t w(t')$$

όπου t' είναι οι ενεργοποιημένες άμεσες μεταβάσεις που ανήκουν στο υψηλότερο επίπεδο προτεραιότητας.

Βασικά, η επιλογή της μετάβασης που θα πυροδοτήσει, είναι απαραίτητη μόνο στις περιπτώσεις που πρέπει να επιλυθεί μία σύγκρουση. Εάν οι ενεργοποιημένες μεταβάσεις είναι ταυτόχρονες, μπορούν να πυροδοτήσουν με οποιαδήποτε σειρά. Έτσι, αν δεν υπάρχει σύγκρουση στο δίκτυο, μπορούν να βρεθούν σε δομικό επίπεδο, τα σύνολα των πιθανά συγκρουόμενων μεταβάσεων, που ονομάζονται **επεκτεινόμενα σύνολα σύγκρουσης (extended conflict sets)**, και ο υπολογισμός των πιθανοτήτων να γίνεται μόνο μεταξύ των βαρών των μεταβάσεων που ανήκουν στο ίδιο επεκτεινόμενο σύνολο σύγκρουσης κι όχι μεταξύ όλων των ενεργοποιημένων μεταβάσεων.

Ένα Γενικευμένο Στοχαστικό δίκτυο Petri που περιέχει απτά και εξαφανιζόμενα μαρκαρίσματα συνεχίζει να είναι ισοδύναμο με μια αλυσίδα Markov. Σε αυτή την περίπτωση, η ισοδύναμη αλυσίδα ονομάζεται **ενσωματωμένη αλυσίδα Markov (embedded Markov chain)**. Αν απομακρύνουμε τα εξαφανιζόμενα μαρκαρίσματα, τότε θα έχουμε μια **μειωμένη ενσωματωμένη αλυσίδα Markov (reduced embedded Markov chain)**. Αυτή η αλυσίδα, που περιέχει μόνο απτά μαρκαρίσματα, χρησιμοποιείται για τον υπολογισμό πιθανοτήτων και **δεικτών απόδοσης (performance indices)**.

Ένα παράδειγμα κατασκευής ενός απτού γράφου προσιτότητας φαίνεται στο Σχήμα 14, που απεικονίζει ένα πολύ απλό Γενικευμένο Στοχαστικό δίκτυο Petri στο οποίο υπάρχει μια σύγκρουση μεταξύ των άμεσων μεταβάσεων. Ο γράφος προσιτότητας του δικτύου φαίνεται στην πάνω δεξιά γωνία. Το διακεκομμένο περίγραμμα για την κατάσταση p_2 υποδεικνύει ότι η κατάσταση είναι εξαφανιζόμενη : όταν η κατάσταση T_1 πυροδοτεί, το σύστημα μεταβαίνει σε ένα νέο μαρκαρίσμα p_2 , στο οποίο δύο άμεσες μεταβάσεις είναι ενεργοποιημένες, και η μετάβαση αλλάζει σε μηδενικό χρόνο είτε σε p_3 είτε σε p_4 , με πιθανότητες $\alpha/\alpha+\beta$ και $\beta/\alpha+\beta$, αντίστοιχα. Ο απτός γράφος προσιτότητας στην κάτω δεξιά γωνία προκύπτει από την απαλοιφή του εξαφανιζόμενου μαρκαρίσματος p_2 . Ο ρυθμός με τον οποίο το σύστημα μεταβαίνει από την p_1 στην p_3 (p_4) προκύπτει από τον πολλαπλασιασμό του ρυθμού μ της μετάβασης-κατάστασης από την p_1 στην p_2 , με τη πιθανότητα μετάβασης από την p_1 στην p_3 (p_4).



Σχήμα 14: Κατασκευή ενός απτού γράφου προσιτότητας.

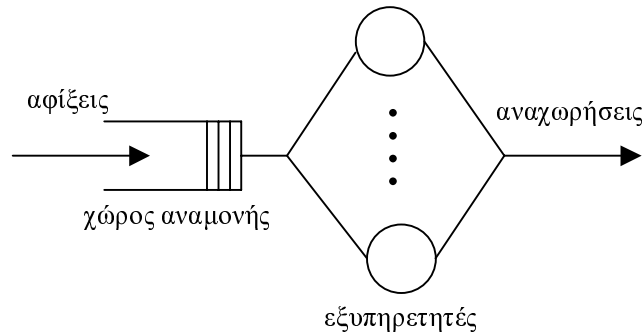
Τα μοντέλα των Στοχαστικών και των Γενικευμένων Στοχαστικών δικτύων Petri παρουσιάζονται ισχυρά σε δυνατότητες αναπαράστασης και η εφαρμογή τους σε τεχνικές μοντελοποίησης είναι πολύ παραγωγική σε αρκετές περιοχές. Ο παράγοντας που παρ' όλα αυτά, περιορίζει την αποδοχή τους ως εργαλείο μοντελοποίησης, είναι η γραφική και υπολογιστική πολυπλοκότητα των μοντέλων αναπαράστασης συστημάτων πραγματικού χρόνου. Πρέπει να επισημανθεί ότι η χρήση τους οφείλεται, κατά κύριο λόγο, στην ύπαρξη κατάλληλων εργαλείων λογισμικού, χωρίς τα οποία η κατασκευή και η επίλυση ενός μοντέλου θα ήταν δυνατές μόνο για τα πιο απλά παραδείγματα. Τα πιο γνωστά εργαλεία λογισμικού που χρησιμοποιούν Στοχαστικά και Γενικευμένα Στοχαστικά δίκτυα Petri για τη μοντελοποίηση, την αξιολόγηση και την ανάλυση της απόδοσης των συστημάτων είναι τα : GreatSPN, SPNP και PetriNetToolbox. Περισσότερες πληροφορίες παρατίθενται στις ιστολίδες των εργαλείων οι οποίες είναι οι εξής :

- GreatSPN : <http://www.di.unito.it/~greatspn>
- SPNP : <http://www.ee.duke.edu/~kst>
- PetriNetToolbox :
http://www.mathworks.com/products/connections/product_main.shtml?prod_id=556

3.7 Δίκτυα Petri με Ουρές

Στο χώρο των στοχαστικών εκδόσεων έχουν επίσης προταθεί διάφορες παραλλαγές των Γενικευμένων Στοχαστικών δικτύων που ενοποιούν πλήρως το μοντέλο των δικτύων ουρών με το μοντέλο των δικτύων Petri. Χαρακτηριστικό παράδειγμα είναι τα **δίκτυα Petri με Ουρές (Queuing Petri Nets - QPNs)**, τα οποία επεκτείνουν, καταρχήν, το αντίστοιχο μη-χρονικό μοντέλο δικτύων Petri με ουρές και, στη συνέχεια, το μοντέλο των Γενικευμένων Στοχαστικών δικτύων με το να προτείνουν τη δομή των **χρονισμένων θέσεων ουρών (time queuing places)**.

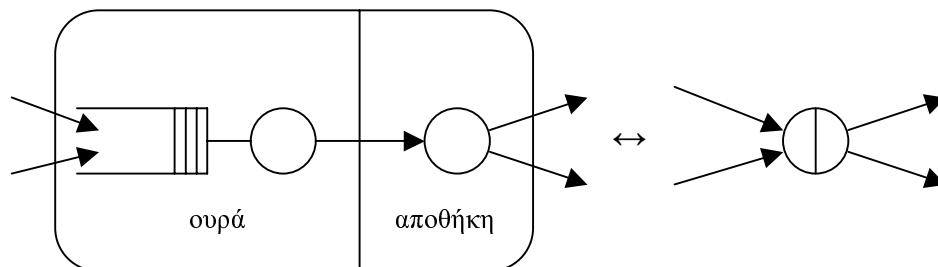
Μια απλή ουρά αποτελείται από ένα **χώρο αναμονής (waiting area)** και ένα **κέντρο εξυπηρέτησης (service center)** με έναν ή περισσότερους **εξυπηρετητές (servers)**. Μία μάρκα που φτάνει σε μια ουρά εξυπηρετείται αμέσως, αν μπορεί να διατεθεί ένας ελεύθερος εξυπηρετητής σε αυτήν ή αν μια μάρκα που εξυπηρετείται προεκχωρηθεί. Σε διαφορετική περίπτωση, η μάρκα πρέπει να περιμένει στο χώρο αναμονής, μέχρι ένας εξυπηρετητής να αφιερωθεί σε αυτήν. Η πολιτική εξυπηρέτησης της ουράς ορίζει τους κανόνες που διέπουν την εξυπηρέτηση μαρκών. Συνήθεις πολιτικές είναι οι **First Come First Served (FCFS)** και ο **διαμοιρασμός των εξυπηρετητών (Processor Sharing - PS)**. Ο **χρόνος εξυπηρέτησης (service time)** μιας μάρκας, εξαρτάται από το χρόνο που θα χρειαστεί το κέντρο εξυπηρέτησης για να την εξυπηρετήσει. Στο Σχήμα 15 φαίνεται η δομή μιας ουράς.



Σχήμα 15: Δομή ουράς

Στα δίκτυα Petri με Ουρές, κάθε χρονισμένη θέση ουράς αποτελείται από ένα **τμήμα ουράς (queue)** και ένα **τμήμα αποθήκης (depository)**, όπου αποθηκεύονται μάρκες-δράσεις που έχουν ολοκληρώσει την αναμονή τους στην ουρά. Κάθε νέα μάρκα εισέρχεται σε μια χρονισμένη θέση με ουρά, μέσω της πυροδότησης της αντίστοιχης μετάβασης εισόδου, και τοποθετείται σε αυτήν σύμφωνα με την πολιτική εξυπηρέτησης της ουράς. Έτσι, μια μάρκα σε μια χρονισμένη θέση με ουρά δεν είναι άμεσα διαθέσιμη για να πυροδοτήσει μια μετάβαση εξόδου. Μετά την “εξυπηρέτηση” της μάρκας - όταν δηλαδή λήξει ο χρόνος εξυπηρέτησης που επιπρόσθετα χαρακτηρίζει μια χρονισμένη θέση με ουρά - τότε μια μάρκα από εκείνες που βρίσκονται στην ουρά θα εισαχθεί άμεσα στο τμήμα της αποθήκης της θέσης. Μόνο οι μάρκες που βρίσκονται στην αποθήκη μιας θέσης μπορούν να θεωρηθούν ως διαθέσιμες για να πυροδοτήσουν μια μετάβαση εξόδου.

Το μοντέλο των δικτύων Petri με Ουρές περιλαμβάνει επίσης **άμεσες θέσεις ουρών (immediate queuing places)**, που ουσιαστικά είναι όμοιες με τις θέσεις ενός Γενικευμένου Στοχαστικού δικτύου. Οι μάρκες σε μια τέτοια θέση μπορούν να θεωρηθούν ότι “εξυπηρετούνται” κατά τρόπο άμεσο, και συνεπώς η εξυπηρέτησή τους λαμβάνει μεγαλύτερη προτεραιότητα από την εξυπηρέτηση των μαρκών που βρίσκονται σε μια χρονισμένη θέση με ουρά. Η πυροδότηση των ενεργοποιημένων, χρονισμένων και άμεσων μεταβάσεων ακολουθεί τους ίδιους κανόνες όπως και στα Γενικευμένα Στοχαστικά δίκτυα Petri χωρίς Ουρές. Στο Σχήμα 16 φαίνεται ένα δίκτυο Petri με Ουρές.



Σχήμα 16: Δίκτυο Petri με Ουρές

Ένα πολύ γνωστό πρόγραμμα που χρησιμοποιεί δίκτυα Petri με Ουρές για την ανάλυση συστημάτων, είναι το QPN-Tool. Περισσότερες πληροφορίες υπάρχουν στην ιστοσελίδα του προγράμματος :

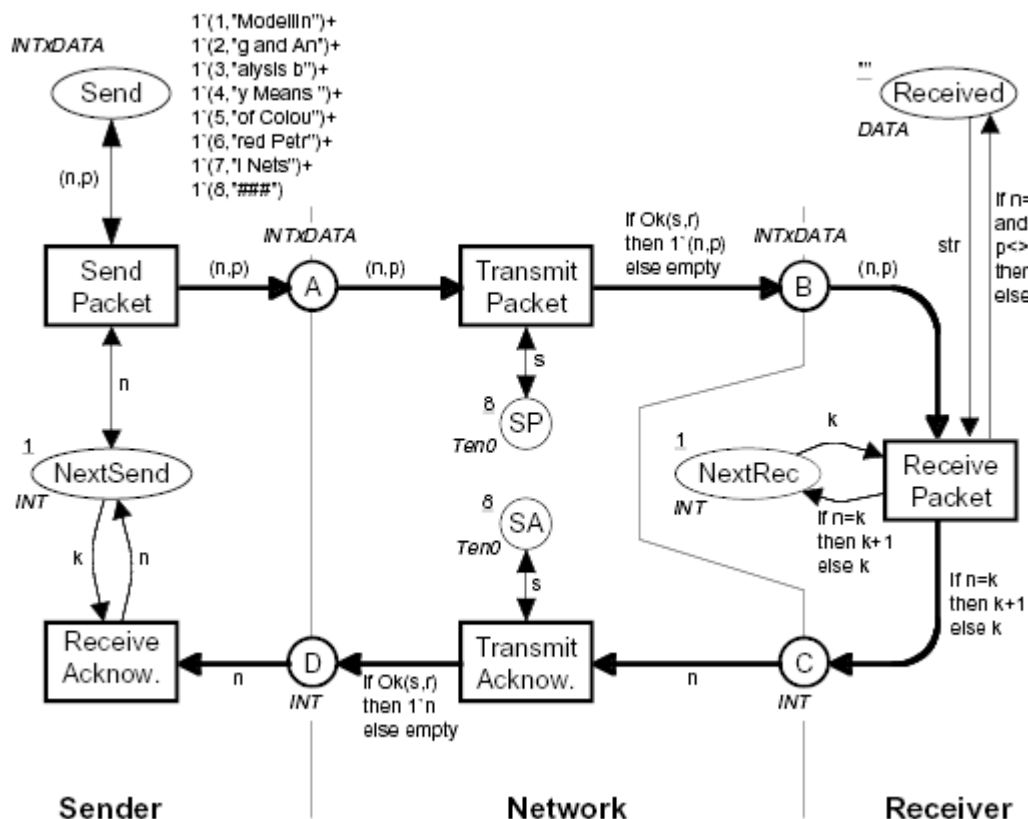
<http://ls4-www.cs.uni-dortmund.de/QPN>

ΚΕΦΑΛΑΙΟ 4^ο

Χρωματισμένα Δίκτυα Petri

4.1 Εισαγωγή στα Χρωματισμένα Δίκτυα Petri

Η παράγραφος αυτή κάνει μια ανεπίσημη εισαγωγή στα **Χρωματισμένα δίκτυα Petri (Coloured Petri nets - CPNs)**, χρησιμοποιώντας ένα παράδειγμα που μοντελοποιεί ένα απλό πρωτόκολλο επικοινωνίας. Το παράδειγμα, αν και είναι πολύ μικρό για να απεικονίσει την τυπική πρακτική χρήση των Χρωματισμένων δικτύων Petri, είναι αρκετά μεγάλο για να απεικονίσει τις βασικές έννοιες της μοντελοποίησης με Χρωματισμένα δίκτυα Petri και τις βασικές ιδέες πίσω από τις μεθόδους ανάλυσης. Στο Σχήμα 1 φαίνεται το Χρωματισμένο δίκτυο Petri που περιγράφει το απλό πρωτόκολλο επικοινωνίας, το οποίο αποτελείται από τρία μέρη : τον αποστολέα Sender, το δίκτυο επικοινωνίας Network και τον παραλήπτη Receiver.



Σχήμα 1: Το Χρωματισμένο δίκτυο Petri που περιγράφει το απλό πρωτόκολλο επικοινωνίας

Σε αντίθεση με τις περισσότερες **γλώσσες προδιαγραφών (specification languages)**, τα δίκτυα Petri είναι προσανατολισμένα και προς τις **καταστάσεις (states)** και προς τις **δράσεις (actions)**, παρέχοντας μια σαφή περιγραφή τόσο των καταστάσεων όσο και των δράσεων ενός συστήματος. Αυτό σημαίνει, ότι αυτός που μοντελοποιεί ένα σύστημα μπορεί ελεύθερα να αποφασίσει αν, σε μια δεδομένη στιγμή, θα επικεντρωθεί στις καταστάσεις ή στις δράσεις.

Οι καταστάσεις σε ένα Χρωματισμένο δίκτυο Petri, όπως και στα απλά δίκτυα Petri, παριστάνονται μέσω των **θέσεων (places)**, οι οποίες σχεδιάζονται ως ελλείψεις ή ως κύκλοι. Στο συγκεκριμένο πρωτόκολλο υπάρχουν δέκα διαφορετικές θέσεις. Τα ονόματα των θέσεων γράφονται μέσα στις ελλείψεις και δεν έχουν κάποια τυπική έννοια, αλλά έχουν τεράστια πρακτική σημασία για την αναγνωσιμότητα των έγχρωμων δικτύων Petri (όπως ακριβώς η χρήση μνημονικών ονομάτων στον παραδοσιακό προγραμματισμό). Το ίδιο ισχύει και για τη γραφική απεικόνιση των θέσεων, δηλαδή για το πάχος των γραμμών, το μέγεθος, το χρώμα, τη θέση κτλ.

Η διαφορά των Χρωματισμένων δικτύων Petri από τα υπόλοιπα, είναι ότι κάθε θέση στα Χρωματισμένα δίκτυα Petri, συσχετίζεται με ένα συγκεκριμένο **τύπο (type)**, ο οποίος ορίζει το είδος των δεδομένων που κάθε θέση μπορεί να περιέχει. Ο τύπος μιας θέσης γράφεται πλαγίως, στο κάτω, αριστερό ή δεξί μέρος της θέσης. Οι τύποι στα Χρωματισμένα δίκτυα Petri είναι παρόμοιοι με τους τύπους δεδομένων σε μια γλώσσα προγραμματισμού.

Στο πρωτόκολλο επικοινωνίας του Σχήματος 1, χρησιμοποιούνται τέσσερις διαφορετικοί τύποι. Οι θέσεις Send, A και B έχουν τον τύπο INTxDATA, που είναι το καρτεσιανό γινόμενο των τύπων INTegers και DATA. Τα στοιχεία αυτού του τύπου αναπαριστούν τα πακέτα που θα μεταδοθούν διαμέσου του Network. Κάθε πακέτο είναι ένα ζεύγος στοιχείων, του οποίου το πρώτο στοιχείο είναι ο αριθμός του πακέτου που είναι τύπου INT, δηλαδή είναι ένας ακέραιος αριθμός, ενώ το δεύτερο στοιχείο είναι τα δεδομένα που περιέχει το πακέτο, που είναι τύπου DATA, δηλαδή είναι μια **συμβολοσειρά κειμένου (text string)**.

Κάθε θέση περιέχει έναν αριθμό από μάρκες, ο οποίος κατά τη διάρκεια της εκτέλεσης ενός Χρωματισμένου δικτύου Petri, αλλάζει. Κάθε μία από αυτές τις μάρκες φέρει μια **τιμή (value)**, η οποία ανήκει στον τύπο που σχετίζεται με τη θέση, όπου βρίσκεται η μάρκα. Παραδείγματος χάριν, το Σχήμα 1 δείχνει, ότι αρχικά η θέση Send έχει τις παρακάτω οχτώ **τιμές μαρκών (token values)**, κάθε μία από τις οποίες αναπαριστούν ένα πακέτο που θα μεταδοθεί μέσω του Network :

- (1, "Modellin")
- (2, "g and An")
- (3, "alysis b")
- (4, "y Means ")
- (5, "of Colou")
- (6, "red Petr")
- (7, "i Nets")
- (8, "### ").

Στο Σχήμα 1 υπάρχει ένα 1` μπροστά από κάθε τιμή μάρκας, που υποδεικνύει ότι υπάρχει ακριβώς μία μάρκα που φέρει τη συγκεκριμένη τιμή. Γενικά, περισσότερες

από μία μάρκες μπορεί να έχουν την ίδια τιμή. Οι μάρκες αυτές σχηματίζουν ένα **πολλαπλό σύνολο (multi-set)** από τιμές μαρκών, όπως το :

$$1 \text{ (2, "g and An")} + 2 \text{ (3, "alysis b")} + 1 \text{ (5, "of Colou")}$$

Το παραπάνω πολλαπλό σύνολο αποτελείται από μία μάρκα με τιμή (2, "g and An"), δύο μάρκες με τιμή (3, "alysis b") και μία μάρκα με τιμή (5, "of Colou"). Ένα πολλαπλό σύνολο είναι παρόμοιο με ένα σύνολο, στο οποίο μπορεί να υπάρχουν πολλαπλές εμφανίσεις του ίδιου στοιχείου. Αν προσθέσουμε το στοιχείο (3, "alysis b") στο σύνολο :

$$\{(2, \text{"g and An"}), (3, \text{"alysis b"}), (5, \text{"of Colou"})\}$$

δεν συμβαίνει τίποτα, γιατί το στοιχείο (3, "alysis b") υπάρχει ήδη στο σύνολο. Αν όμως, προσθέσουμε το στοιχείο (3, "alysis b") στο πολλαπλό σύνολο :

$$1 \text{ (2, "g and An")} + 1 \text{ (3, "alysis b")} + 1 \text{ (5, "of Colou")}$$

θα πάρουμε ένα πολλαπλό σύνολο με τέσσερα στοιχεία αντί για για τρία :

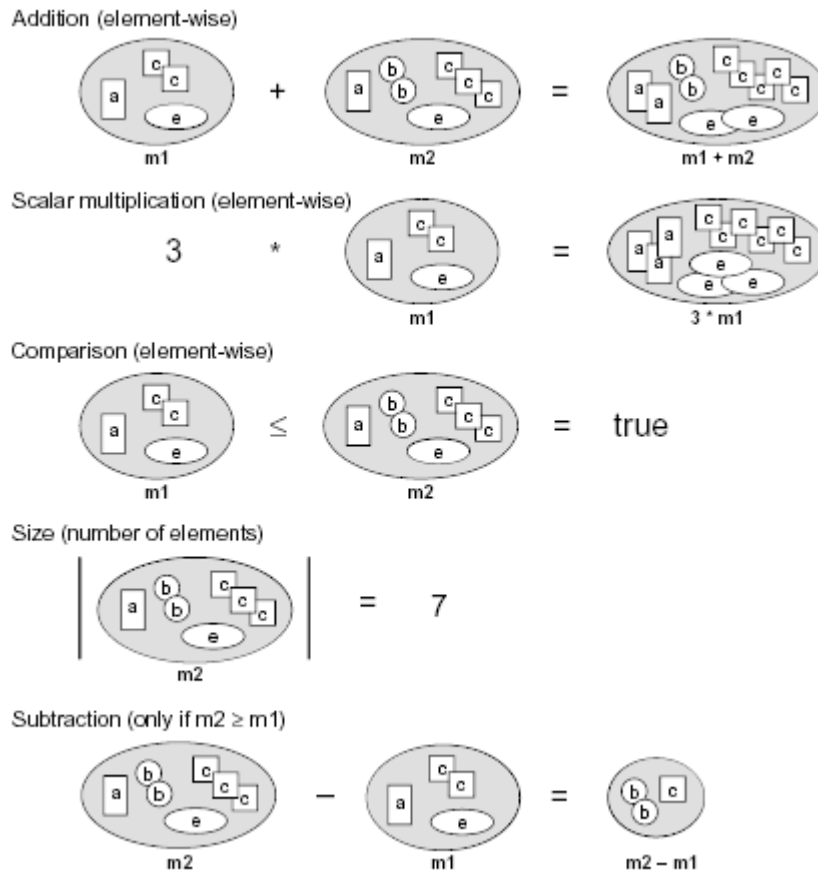
$$1 \text{ (2, "g and An")} + 2 \text{ (3, "alysis b")} + 1 \text{ (5, "of Colou").}$$

Οι ακέραιοι αριθμοί μπροστά από τον τελεστή `ονομάζονται **συντελεστές (coefficients)**. Στο παράδειγμα μας το (2, "g and An") και το (5, "of Colou") έχουν ως συντελεστή τη μονάδα, ενώ το (3, "alysis b") έχει ως συντελεστή το δύο. Όλες οι άλλες τιμές αυτού του τύπου (του τύπου INTxDATA), έχουν ως συντελεστή το μηδέν κι γι'αυτό το λόγο παραλείπονται.

Για τα πολλαπλά σύνολα, ορίζονται πράξεις όπως η **πρόσθεση (addition)**, ο **κλιμακωτός πολλαπλασιασμός (scalar multiplication)**, η **σύγκριση (comparison)**, ο **υπολογισμός του μεγέθους ενός πολλαπλού συνόλου (size)** και η **αφαίρεση (subtraction)**. Η αφαίρεση δύο πολλαπλών συνόλων $m_2 - m_1$ ορίζεται, μόνο όταν ισχύει $m_2 \geq m_1$. Στο Σχήμα 2 απεικονίζονται οι πράξεις μεταξύ πολλαπλών συνόλων που έχουν στοιχεία από το σύνολο $\{a, b, c, d, e\}$.

Ας εξετάσουμε τώρα τις υπόλοιπες εννέα θέσεις του Σχήματος 1. Η θέση A αναπαριστά πακέτα που έχουν αποσταλεί στο δίκτυο Network από τον αποστολέα Sender, αλλά δεν έχουν ακόμα μεταδοθεί από το δίκτυο Network. Ανάλογα, η θέση B παριστάνει πακέτα που έχουν μεταδοθεί από το δίκτυο Network, αλλά δεν έχουν ακόμα παραληφθεί από τον αποστολέα Receiver. Οι δύο αυτές θέσεις έχουν τον ίδιο τύπο με τη θέση Send, και αρχικά δεν περιέχουν καμία μάρκα. Μία μόνο μάρκα στη θέση Received αναπαριστά τα δεδομένα των πακέτων που έχουν παραληφθεί, αγνοώντας τα περιεχόμενα πακέτων που έχουν ήδη παραληφθεί ή έχουν παραληφθεί με λάθος σειρά. Αρχικά, δεν έχουν παραληφθεί δεδομένα κι έτσι η μάρκα της θέσης Received, που είναι τύπου DATA, είναι μια κενή συμβολοσειρά κειμένου "". Στο τέλος της μετάδοσης, η θέση Received θα πρέπει να περιέχει τη συμβολοσειρά κειμένου :

“Modelling and Analysis by Means of Coloured Petri Nets”



Σχήμα 2: Μερικές πράξεις πάνω στα πολλαπλά σύνολα

Οι θέσεις C και D είναι παρόμοιες με τις θέσεις A και B, με τη διαφορά ότι αναπαριστούν **βεβαιώσεις λήψης πακέτων (acknowledgements)**, που αποστέλλονται από τον Receiver στο Sender. Κάθε βεβαίωση λήψης πακέτου φέρει έναν αριθμό και κανένα άλλο δεδομένο. Έτσι οι θέσεις C και D είναι τύπου INT. Οι θέσεις NextSend και NextRec παριστάνουν μετρητές, που κρατάνε τον αριθμό του επόμενου πακέτου που είναι να αποσταλεί ή να παραληφθεί. Έχουν τύπο INT και κάθε μία από αυτές έχουν αρχικά μία μάρκα με τιμή 1. Οι δύο τελευταίες θέσεις SP και SA έχουν τον τύπο Ten0, ο οποίος περιέχει όλους τους ακέραιους αριθμούς από το μηδέν ως το δέκα. Ο ρόλος των θέσεων αυτών θα εξηγηθεί αργότερα.

Η κατάσταση ενός Χρωματισμένου δικτύου Petri ονομάζεται **μαρκάρισμα (marking)** και αποτελείται από έναν αριθμό από μάρκες που είναι τοποθετημένες - κατανεμημένες - στις θέσεις του δικτύου. Οι μάρκες που είναι παρούσες σε μια συγκεκριμένη θέση αποτελούν το μαρκάρισμα αυτής της θέσης. Το αρχικό μαρκάρισμα κάθε θέσης αναγράφεται υπογεγραμμένο, δίπλα από τη θέση. Όταν η επιγραφή του αρχικού μαρκάρισματος έχει μεγάλο μήκος, μπορούμε να παραλείψουμε την υπογράμμιση, όπως έγινε για τη θέση Send στο Σχήμα 1.

Για ιστορικούς λόγους, κάποιες φορές αναφερόμαστε στις τιμές που έχουν οι μάρκες ως **χρώματα (colours)** και στους τύπους δεδομένων ως **σύνολα χρωμάτων (colour sets)**. Αυτή είναι μια μεταφορική εικόνα, όπου θεωρούμε ότι οι μάρκες σε ένα

Χρωματισμένο δίκτυο Petri είναι ευδιάκριτες και ως εκ τούτου “χρωματισμένες” – σε αντίθεση με τα χαμηλού επιπέδου δίκτυα Petri που έχουν “μαύρες” μάρκες και δεν μπορούν να διακριθούν μεταξύ τους. Οι τύποι ενός Χρωματισμένου δικτύου Petri μπορεί να είναι σύνθετοι σε αυθαίρετο βαθμό, παραδείγματος χάριν μια εγγραφή (**record**), όπου το ένα **πεδίο (field)** είναι ένας πραγματικός αριθμός, το άλλο πεδίο μια συμβολοσειρά κειμένου και το τρίτο πεδίο μια λίστα ακεραίων αριθμών. Έτσι, είναι πιο σωστό να φανταζόμαστε μια συνέχεια χρωμάτων, όπως το χρωματικό φάσμα στη φυσική, αντί για μερικές διακριτές τιμές χρωμάτων, όπως το κόκκινο, το πράσινο και το μπλέ.

Οι δράσεις σε ένα Χρωματισμένο δίκτυο Petri παριστάνονται, όπως και στα χαμηλού επιπέδου δίκτυα Petri μέσω των μεταβάσεων, που σχεδιάζονται ως ορθογώνια. Στο πρωτόκολλο επικοινωνίας υπάρχουν πέντε διαφορετικές μεταβάσεις. Σε μια μετάβαση, ένα εισερχόμενο τόξο υποδεικνύει ότι η μετάβαση μπορεί να αφαιρέσει μάρκες από την αντίστοιχη θέση, ενώ ένα εξερχόμενο τόξο υποδεικνύει ότι μπορεί να προσθέσει μάρκες στην αντίστοιχη θέση. Ο ακριβής αριθμός των μαρκών, καθώς και οι τιμές τους, ορίζονται από τις **επιγραφές τόξου (arc expressions)**, οι οποίες τοποθετούνται δίπλα από τα τόξα. Η μετάβαση SendPacket έχει τρία τόξα που την περιβάλλουν, με δύο διαφορετικές επιγραφές τόξου : (n, p) και n. Δύο από αυτά τα τόξα είναι **διπλά (double)**. Κάθε διπλό τόξο είναι μια συντομογραφία δύο τόξων με αντίθετη κατεύθυνση και με ίδια επιγραφή τόξου. Δηλαδή, υπάρχουν στην πραγματικότητα πέντε διαφορετικά τόξα, δύο εισερχόμενα και τρία εξερχόμενα. Οι συγκεκριμένες επιγραφές τόξου περιέχουν δύο μεταβλητές χωρίς τιμή : τη μεταβλητή n που είναι τύπου INT και τη μεταβλητή p που είναι τύπου DATA. Για να μιλήσουμε για την **εκτέλεση (occurrence)** της μετάβασης SendPacket, πρέπει να **καταχωρήσουμε (bind)** στη μεταβλητή n μία τιμή τύπου INT (δηλαδή μία τιμή από το σύνολο των ακεραίων αριθμών) και στη μεταβλητή p μία τιμή τύπου DATA. Αλλιώς δεν μπορούμε να υπολογίσουμε τις επιγραφές τόξου (n, p) και n.

Έστω ότι καταχωρούμε στη μεταβλητή n της μετάβασης SendPacket την τιμή 1, ενώ στη μεταβλητή p της ίδιας μετάβασης την τιμή “Modellin”. Αυτό μας δίνει την **καταχώρηση (binding)** :

$$\langle n = 1, p = \text{“Modellin”} \rangle$$

για την οποία οι επιγραφές τόξου υπολογίζονται ως :

$$\begin{array}{lcl} (n, p) & \rightarrow & (1, \text{“Modellin”}) \\ n & \rightarrow & 1 \end{array}$$

Οι παραπάνω επιγραφές τόξου υποδεικνύουν ότι η εκτέλεση της μετάβασης SendPacket θα απομακρύνει μία μάρκα με τιμή (1, “Modellin”) από τη θέση Send και μία μάρκα με τιμή 1 από τη θέση NextSend. Και οι δύο μάρκες είναι διαθέσιμες, δηλαδή παρούσες στις δύο θέσεις κι έτσι, με την παραπάνω καταχώρηση, η μετάβαση SendPacket ενεργοποιείται. Αυτό σημαίνει ότι η μετάβαση μπορεί να **εκτελεστεί (occur)**. Όταν η μετάβαση SendPacket εκτελείται, οι δύο συγκεκριμένες μάρκες θα αφαιρεθούν από τις θέσεις εισόδου Send και NextSend. Ταυτόχρονα, τρεις μάρκες θα προστεθούν στις θέσεις εξόδου : οι θέσεις Send και A θα πάρουν μία μάρκα με τιμή (1, “Modellin”), ενώ η θέση NextSend θα πάρει μία μάρκα με τιμή 1. Έτσι το συνολικό αποτέλεσμα της εκτελούμενης μετάβασης είναι ότι πρόσθεσε στη θέση A, μία μάρκα που αναπαριστά το πακέτο νούμερο ένα. Αυτό σημαίνει ότι ο αποστολέας

Sender μεταφέρει ένα αντίγραφο του πακέτου νούμερο ένα στο buffer εισόδου του δικτύου Network. Το πακέτο δεν απομακρύνεται από τη θέση Send, γιατί μπορεί να χρειαστεί να αναμεταδοθεί. Επίσης, ούτε ο μετρητής NextSend αυξάνεται, γιατί το πρωτόκολλο είναι **απαισιόδοξο (pessimistic)**, με την έννοια ότι θα συνεχίζει να αναμεταδίδει ένα πακέτο, μέχρι να λάβει μία θετική βεβαίωση που να επιβεβαιώνει, ότι το πακέτο έχει παραληφθεί από τον αποστολέα Receiver.

Υπάρχουν βέβαια, πολλές άλλες καταχωρήσεις που μπορούμε να δοκιμάσουμε για τη μετάβαση SendPacket. Παρ'όλα αυτά, καμία από αυτές τις καταχωρήσεις δεν είναι επιτρεπτές στο αρχικό μαρκάρισμα του δικτύου Petri, δηλαδή δεν μπορούν να καταχωρηθούν στις μάρκες οι τιμές αυτών των καταχωρήσεων. Αυτό μπορεί να αποδειχθεί ως εξής : η θέση NextSend έχει μόνο μία μάρκα, η οποία φέρει την τιμή 1. Επομένως, πρέπει να καταχωρήσουμε στη μεταβλητή n την τιμή 1. Αυτό σημαίνει ότι η επιγραφή του εισερχόμενου τόξου από τη θέση Send θα έχει μία τιμή της μορφής (1, ...). Όμως η θέση Send έχει μόνο μάρκα αυτής της μορφής : τη (1, "Modellin"), και άρα στη μεταβλητή p πρέπει να καταχωρηθεί η τιμή "Modellin".

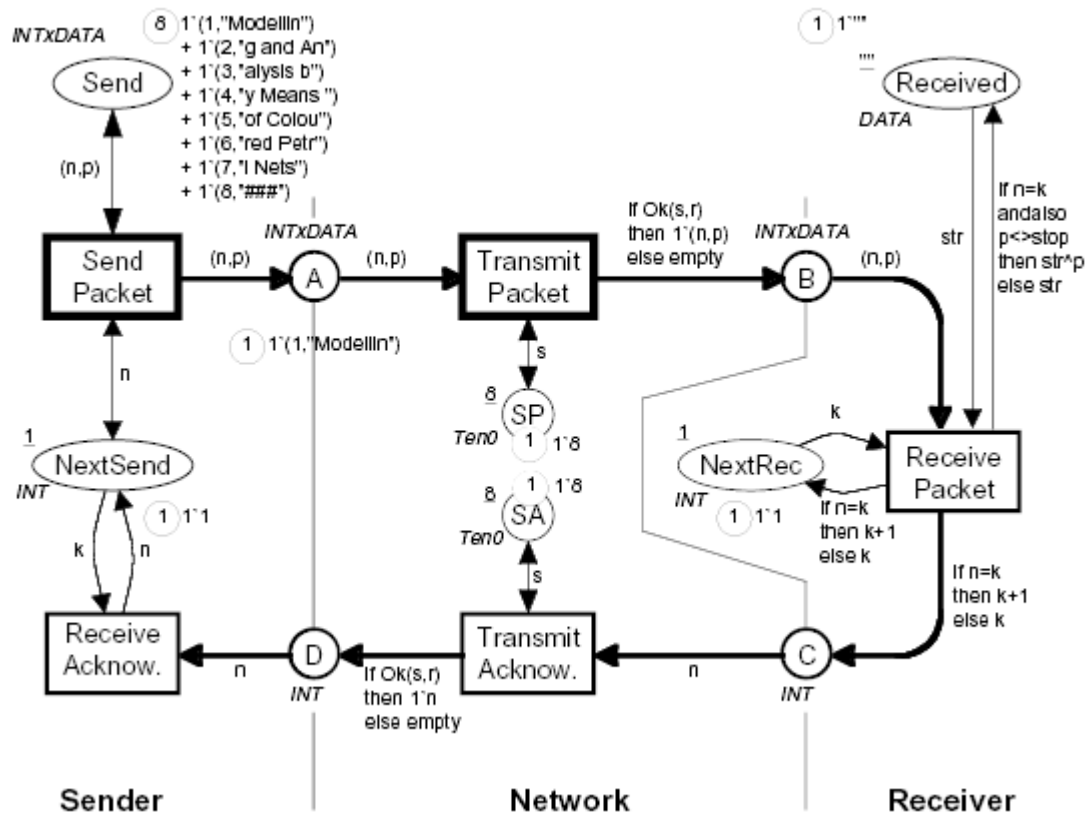
Στα Χρωματισμένα δίκτυα Petri εισάγεται η έννοια των στοιχείων καταχώρησης. Ένα ζεύγος, το οποίο αποτελείται από μία μετάβαση και από μία καταχώρηση για τις μεταβλητές που εμφανίζονται στα τόξα γύρω από τη μετάβαση (τόσο στα εισερχόμενα τόξα όσο και στα εξερχόμενα), ονομάζεται **στοιχείο καταχώρησης (binding element)**. Ένα στοιχείο καταχώρησης είναι **ενεργοποιημένο (enabled)**, όταν οι τιμές που καταχωρούνται στις μεταβλητές των τόξων γύρω από μια μετάβαση, μπορούν να ενεργοποιήσουν αυτή τη μετάβαση. Παραπάνω είδαμε ότι το στοιχείο καταχώρησης :

(SendPacket, < n = 1, p = "Modellin" >)

είναι ενεργοποιημένο στο αρχικό μαρκάρισμα και οδηγεί σε ένα νέο μαρκάρισμα που είναι ίδιο με το αρχικό, με τη διαφορά ότι μια νέα μάρκα με τιμή (1, "Modellin") έχει προστεθεί στη θέση A. Το νέο μαρκάρισμα φαίνεται στο Σχήμα 3. Ο αριθμός των μαρκών σε κάθε θέση αναγράφεται μέσα σε ένα μικρό κύκλο δίπλα από τη θέση, ενώ οι λεπτομερείς τιμές που έχουν οι μάρκες, αναγράφονται σε μια συμβολοσειρά κειμένου δίπλα στο μικρό κύκλο. Οι τιμές των μαρκών μπορεί να είναι φανερές ή κρυφές. Αυτό είναι πολύ βολικό, παραδείγματος χάριν, όταν οι τιμές είναι σύνθετες. Για παράδειγμα, υπάρχουν μοντέλα Χρωματισμένων δικτύων Petri, στα οποία μία τυπική τιμή μάρκας είναι μια λίστα αποτελούμενη, έως και από 50.000 εγγραφές τράπεζας. Σε τέτοιες περιπτώσεις, δεν είναι επιθυμητό να εκθέτονται άμεσα οι τιμές των μαρκών στο διάγραμμα του Χρωματισμένου δικτύου Petri.

Στο Σχήμα 3 οι μεταβάσεις SendPacket και TransmitPacket έχουν μια πιο χοντρή γραμμή περιγράμματος. Με αυτόν τον τρόπο, υποδεικνύεται ότι αυτές οι δύο μεταβάσεις είναι ενεργοποιημένες δηλαδή έχουν ενεργοποιημένες καταχωρήσεις, ενώ οι υπόλοιπες δεν έχουν. Για τη μετάβαση SendPacket η μοναδική καταχώρηση που είναι ενεργοποιημένη, είναι αυτή που χρησιμοποιήσαμε παραπάνω και αντιστοιχεί στην επαναμετάδοση του πακέτου νούμερο ένα. Για τη μετάβαση TransmitPacket η κατάσταση είναι ελαφρώς πιο περίπλοκη, αφού τώρα υπάρχουν τέσσερις διαφορετικές μεταβλητές : η μεταβλητή n τύπου INT, η μεταβλητή p τύπου DATA, η μεταβλητή s τύπου Ten0 και η μεταβλητή r τύπου Ten1. Ο τύπος Ten0 περιέχει

όλους τους ακέραιους αριθμούς από το μηδέν ως το δέκα, όπως προαναφέρθηκε, ενώ ο τύπος Ten1 περιέχει όλους τους ακέραιους αριθμούς από το ένα ως το δέκα.



Σχήμα 3: Το μαρκάρισμα του δικτύου Petri του Σχήματος 1 μετά την εκτέλεση της μετάβασης SendPacket

Στο μαρκάρισμα του Σχήματος 3, η θέση A έχει μία μάρκα με τιμή (1, “Modellin”). Από αυτό προκύπτει ότι στη μεταβλητή n πρέπει να καταχωρηθεί η τιμή 1, ενώ στη μεταβλητή p η τιμή “Modellin”. Η θέση SP έχει μία μάρκα με τιμή 8. Από εδώ προκύπτει ότι στη μεταβλητή s πρέπει να καταχωρηθεί η τιμή 8. Η μεταβλητή r εμφανίζεται μόνο σε ένα εξερχόμενο τόξο, κι επομένως μπορεί να καταχωρηθεί στη μεταβλητή r οποιαδήποτε τιμή που ανήκει στον τύπο Ten1, χωρίς να επηρεαστεί η ενεργοποίηση της μετάβασης TransmitPacket. Αυτό σημαίνει ότι έχουμε δέκα διαφορετικές ενεργοποιημένες καταχωρήσεις :

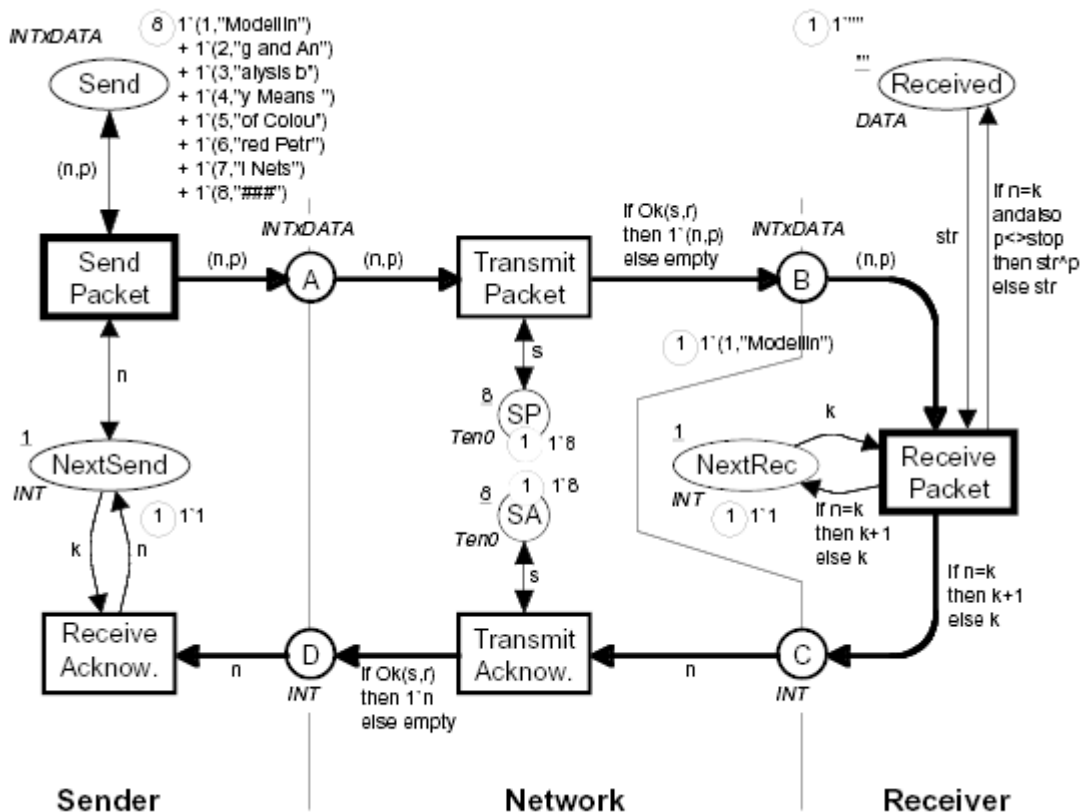
- < n = 1, p = “Modellin”, s = 8, r = 1 >
- < n = 1, p = “Modellin”, s = 8, r = 2 >
- < n = 1, p = “Modellin”, s = 8, r = 3 >
-
-
- < n = 1, p = “Modellin”, s = 8, r = 9 >
- < n = 1, p = “Modellin”, s = 8, r = 10 >

Η επιγραφή του εξερχόμενου τόξου από τη θέση TransmitPacket :

if Ok(s, r) then 1` (n, p) else empty

είναι μια πρόταση if, η οποία καλεί τη συνάρτηση Ok(s, r). Η συνάρτηση Ok(s, r) συγκρίνει τις τιμές των μεταβλητών s και r, και επιστρέφει την τιμή true, αν και μόνο αν $r \leq s$. Δηλαδή, στην προκειμένη περίπτωση, η συνάρτηση Ok(s, r) θα επιστρέψει την τιμή true για τις πρώτες οχτώ καταχωρήσεις, ενώ για τις τελευταίες δύο καταχωρήσεις θα επιστρέψει την τιμή false. Όταν η συνάρτηση Ok(s, r) είναι αληθής, η μετάβαση προσθέτει μία μάρκα με τιμή (1, "ModellIn") στη θέση B. Διαφορετικά δεν προστίθεται καμία μάρκα, όπως δηλώνει και η θέση empty η οποία συμβολίζει το κενό πολλαπλό σύνολο. Η επιλογή μεταξύ των δέκα ενεργοποιημένων στοιχείων καταχώρησης είναι δίκαια κι έτσι, η πιθανότητα για επιτυχή μετάδοση είναι 80%, ενώ η πιθανότητα να χαθεί ένα πακέτο είναι 20%. Μεταβάλλοντας την τιμή της μάρκας στη θέση SP, μπορούμε να αλλάξουμε τις πιθανότητες. Αν η τιμή της μάρκας είναι 10, δεν χάνονται ποτέ πακέτα, ενώ αν είναι 0, χάνονται όλα τα πακέτα. Δηλαδή η θέση SP παριστάνει το ρυθμό επιτυχίας μετάδοσης πακέτων (Success rate for Packets).

Έστω τώρα, ότι η μετάβαση TransmitPacket εκτελείται με μία από τις οχτώ πρώτες καταχωρήσεις. Αυτό θα οδηγήσει σε ένα νέο μαρκάρισμα του δικτύου Petri που φαίνεται στο Σχήμα 4.



Σχήμα 4: Το μαρκάρισμα του δικτύου Petri του Σχήματος 1 μετά την εκτέλεση των μεταβάσεων SendPacket και TransmitPacket

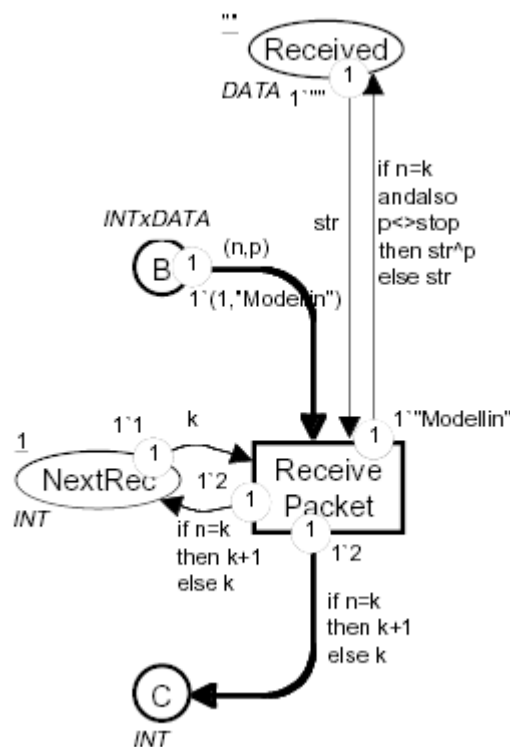
Υπάρχουν πάλι δύο ενεργοποιημένες μεταβάσεις, η SendPacket και η ReceivePacket. Μπορούμε είτε να αναμεταδώσουμε το πακέτο νούμερο ένα μέσω του στοιχείου καταχώρησης :

(SendPacket, < n = 1, p = "Modellin" >),

είτε να παραλάβουμε το πακέτο νούμερο ένα μέσω του στοιχείου καταχώρησης :

(ReceivePacket, < n = 1, p = "Modellin", k = 1, str = "" >).

Όταν εκτελείται το δεύτερο στοιχείο καταχώρησης, προστίθενται ή αφαιρούνται οι μάρκες που φαίνονται στο Σχήμα 5, το οποίο είναι ένα τμήμα του Χρωματισμένου δικτύου Petri το Σχήματος 4 κατά τη διάρκεια της εκτέλεσης της μετάβασης ReceivePacket.



Σχήμα 5: Οι μάρκες που εμπλέκονται σε μια εκτέλεση της μετάβασης ReceivePacket

Οι επιγραφές των τριών εξερχόμενων τόξων :

if n=k then k+1 else k,
if n=k and also p<>stop then str^p else str

συγκρίνουν τον αριθμό n του εισερχόμενου πακέτου με τον αριθμό k του αναμενόμενου πακέτου. Αν οι τιμές είναι ίδιες, όπως στην περίπτωση μας, το εισερχόμενο πακέτο είναι αυτό που περιμένει ο παραλήπτης Receiver. Στη συνέχεια, ο παραλήπτης Receiver προσθέτει τα δεδομένα p του νέου πακέτου στα δεδομένα str που έχουν ήδη παραληφθεί και βρίσκονται στη θέση Received, εκτός αν η μεταβλητή p είναι ίση με stop, το οποίο συμβολίζεται από τη σταθερά "###". Η τιμή της μάρκας στη θέση NextRec αυξάνεται κατά ένα και μια βεβαίωση λήψης πακέτου στέλνεται

μέσω της θέσης C. Η βεβαίωση λήψης πακέτου περιέχει τον αριθμό του επόμενου πακέτου που θέλει ο Receiver να παραλάβει, δηλαδή την τιμή της μάρκας της θέσης NextRec.

Αν οι τιμές των μεταβλητών n και k διαφέρουν η μία από την άλλη, το πακέτο που παραλαμβάνεται, δεν είναι αυτό που περιμένει ο παραλήπτης Receiver κι έτσι αγνοείται. Οι θέσεις Received και NextRec παραμένουν αμετάβλητες και μια βεβαίωση λήψης πακέτου στέλνεται μέσω της θέσης C. Όπως και προηγουμένως, η βεβαίωση λήψης πακέτου περιέχει τον αριθμό του επόμενου πακέτου που θέλει ο Receiver να παραλάβει, δηλαδή την τιμή της μάρκας της θέσης NextRec.

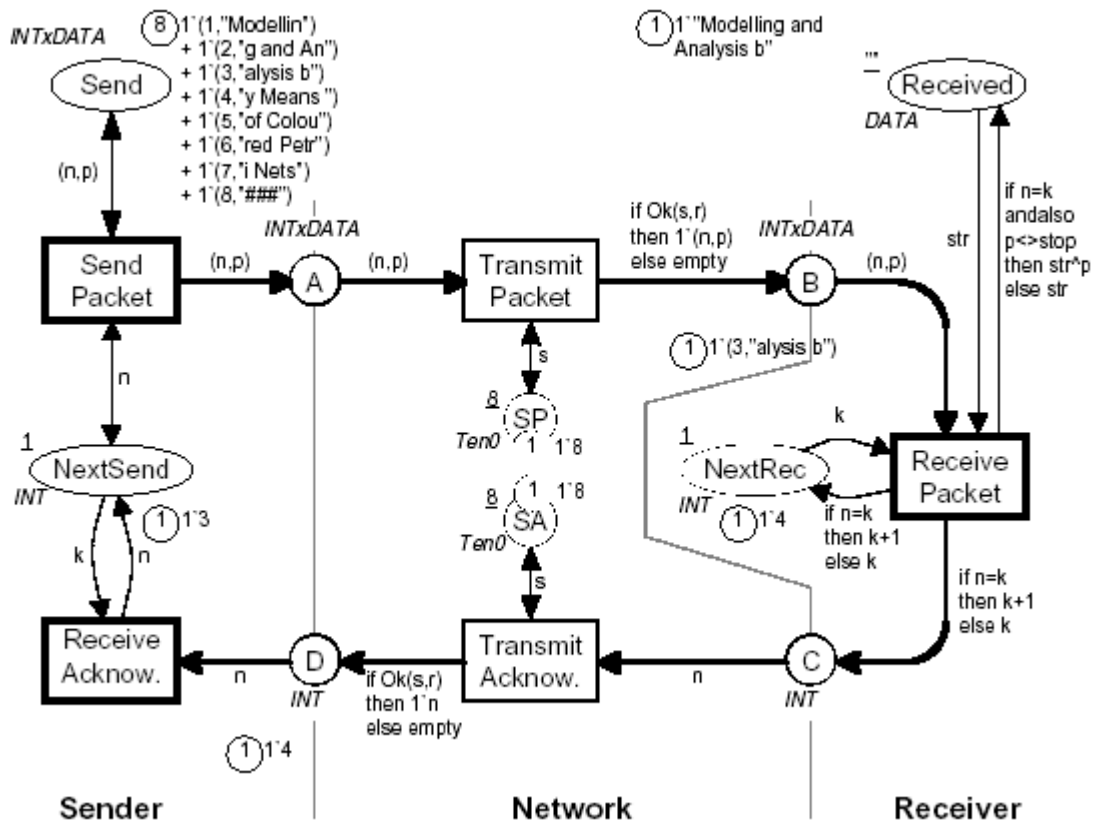
Μετά την εκτέλεση της μετάβασης ReceivePacket που φαίνεται στο Σχήμα 5, έχουμε ένα μαρκάρισμα, όπου η θέση C έχει μια μαρκά με τιμή 2. Η μάρκα αναπαριστά μια βεβαίωση λήψης πακέτου, η οποία μπορεί να μεταδοθεί (ή να χαθεί) μέσω της θέσης TransmitAcknowledgement. Η μετάβαση αυτή λειτουργεί με παρόμοιο τρόπο με τη θέση TransmitPacket. Αυτό σημαίνει ότι η βεβαίωση λήψης πακέτου μπορεί να χαθεί με πιθανότητα που καθορίζεται από τη μάρκα στη θέση SA.

Αν η βεβαίωση λήψης πακέτου φτάσει στη θέση D, η μετάβαση ReceiveAcknowledgement ενεργοποιείται. Η εκτέλεση αυτής της μετάβασης έχει ως αποτέλεσμα να ανανεωθεί ο αριθμός στη θέση NextSend, αντικαθιστώντας την παλιά τιμή k με τον αριθμό n που περιέχεται στη βεβαίωση λήψης πακέτου και στην περίπτωση μας είναι ο αριθμός 2. Αυτό σημαίνει ότι ο αποστολέας Sender αρχίζει τώρα να στέλνει το πακέτο νούμερο δύο, δηλαδή το πακέτο (2, "g and An").

Μετά την εκτέλεση πενήντα περίπου στοιχείων καταχώρησης, το Χρωματισμένο δίκτυο Petri μπορεί να φτάσει στο ενδιάμεσο μαρκάρισμα που φαίνεται στο Σχήμα 6.

Στα αριστερά του δικτύου, βλέπουμε ότι ο αποστολέας Sender στέλνει το πακέτο νούμερο τρία. Βλέπουμε επίσης, ότι ένα αντίγραφο του πακέτου είναι παρόν στη θέση B. Στα δεξιά του δικτύου, βλέπουμε ότι η συμβολοσειρά "Modelling and Analysis b", που είναι το περιεχόμενο των τριών πρώτων πακέτων, έχει παραληφθεί και ο παραλήπτης Receiver περιμένει τώρα το πακέτο νούμερο τέσσερα. Έτσι το πακέτο στη θέση B θα αγνοηθεί από τον Receiver. Ακόμα, στη θέση D υπάρχει μια βεβαίωση λήψης πακέτου. Όταν εκτελείται η μετάβαση ReceiveAcknowledgement, η μάρκα στη θέση NextSend θα πάρει την τιμή 4 και αυτό σημαίνει ότι ο αποστολέας Sender θα αρχίσει να στέλνει το πακέτο νούμερο τέσσερα.

Δεν υπάρχει εγγύηση ότι οι μάρκες θα απομακρυνθούν από μια θέση με τη σειρά με την οποία προστέθηκαν. Κατά τη διάρκεια μιας προσομοίωσης του Χρωματισμένου δικτύου Petri, η θέση A μπορεί να περιέχει αρκετές μάρκες και κάθε μία από αυτές μπορεί να επιλεγεί ως η επόμενη για να μεταδοθεί στη θέση B. Το ίδιο ισχύει και για τις θέσεις B, C και D. Έτσι τα πακέτα μπορούν να προσπεράσουν το ένα το άλλο στις θέσεις A και B, ενώ οι βεβαιώσεις λήψης πακέτων μπορούν να προσπεράσουν η μία την άλλη στις θέσεις C και D. Εάν είναι επιθυμητό, μπορεί εύκολα να οριστεί μια **πειθαρχία ουράς (queuing discipline)**. Για να γίνει αυτό, εφοδιάζουμε τις θέσεις A, B, C και D με ένα τύπο λίστας που παίρνει τιμές του προηγούμενου τύπου της θέσης. Κάθε μία από αυτές τις θέσεις έχει πάντα μία μάρκα, της οποίας η αρχική τιμή είναι η άδεια λίστα. Στο ένα άκρο της λίστας εισάγουμε πακέτα, ενώ από το άλλο άκρο αφαιρούμε πακέτα. Μια τέτοια θέση ορίζεται ως **θέση ουράς (queuing place)**.



Σχήμα 6: Ενδιάμεσο μαρκάρισμα του Χρωματισμένου δικτύου Petri του Σχήματος 1

Όταν το τελευταίο πακέτο (8, “###”) ληφθεί επιτυχώς από τον παραλήπτη Receiver, η μάρκα στη θέση NextRec θα πάρει την τιμή 9 (ένα περισσότερο από τον αριθμό των πακέτων). Η τιμή αυτή θα μεταδοθεί, μέσω μιας βεβαίωσης λήψης πακέτου στον αποστολέα Sender. Στη συνέχεια, η μάρκα στη θέση NextSend θα πάρει την τιμή 9 και η αποστολή πακέτων θα σταματήσει, αφού δεν υπάρχει κανένα πακέτο με αυτόν τον αριθμό. Μετά από μερικά ακόμα βήματα, οι θέσεις A, B, C και D έχουν καθαρίσει από πακέτα ή από βεβαιώσεις λήψης πακέτων και το Χρωματισμένο δίκτυο Petri θα φτάσει στο τελικό μαρκάρισμα που φαίνεται στο Σχήμα 7. Αυτό το μαρκάρισμα είναι νεκρό, δηλαδή δεν έχει καμία ενεργοποιημένη μετάβαση.

Στο Σχήμα 8 φαίνονται οι δηλώσεις των τύπων, των μεταβλητών και των συναρτήσεων που ορίζονται στο Χρωματισμένο δίκτυο Petri του Σχήματος 1.

Όπως είδαμε παραπάνω, σε ένα μαρκάρισμα μπορεί να υπάρχουν αρκετά ενεργοποιημένα στοιχεία καταχώρησης. Για παράδειγμα, είδαμε ότι το μαρκάρισμα του Σχήματος 4 έχει δύο ενεργοποιημένα στοιχεία καταχώρησης, το ένα που αφορά τη μετάβαση SendPacket και το άλλο που αφορά τη μετάβαση ReceivePacket. Για την ακρίβεια, αυτά τα δύο στοιχεία καταχώρησης είναι ταυτόχρονα ενεργοποιημένα, που σημαίνει ότι οι μεταβάσεις ReceivePacket και SendPacket μπορούν να **εκτελεστούν ταυτόχρονα (occur concurrently)**. Ο κανόνας για τον ταυτοχρονισμό είναι πολύ απλός. Ένα σύνολο από στοιχεία καταχώρησης είναι ταυτόχρονα ενεργοποιημένο, αν υπάρχουν τόσες πολλές μάρκες, έτσι ώστε κάθε στοιχείο καταχώρησης να μπορεί να πάρει τόσες μάρκες, όσες χρειάζεται, δηλαδή όσες προσδιορίζονται από τις επιγραφές των εισερχόμενων τόξων, χωρίς να μοιράζεται τις μάρκες με άλλα στοιχεία καταχώρησης. Έτσι, ένα **βήμα (step)** από ένα μαρκάρισμα στο επόμενο, μπορεί να εμπλέκει ένα πολλαπλό σύνολο από στοιχεία καταχώρησης. Το πολλαπλό σύνολο πρέπει να είναι πεπερασμένο και μη-κενό.

Γενικά, μπορεί μία μετάβαση να είναι ταυτόχρονα ενεργοποιημένη με τον εαυτό της, χρησιμοποιώντας δύο διαφορετικές καταχωρήσεις ή την ίδια καταχώρηση δύο φορές. Στο παράδειγμα του πρωτοκόλλου, κάθε μετάβαση έχει τουλάχιστον μία θέση εισόδου με μόνο μία μάρκα. Δύο στοιχεία καταχώρησης που αφορούν την ίδια μετάβαση, θα χρειαστούν και οι δύο αυτή τη μοναδική μάρκα. Άρα τα στοιχεία καταχώρησης βρίσκονται σε σύγκρουση μεταξύ τους και καμία μετάβαση δεν μπορεί να εκτελεστεί ταυτόχρονα με τον εαυτό της.

Η εκτέλεση ενός Χρωματισμένου δικτύου Petri περιγράφεται με τη βοήθεια μιας **ακολουθίας εκτέλεσης (occurrence sequence)**, η οποία κάνει μια λίστα των προσιτών μαρκαρισμάτων και των βημάτων που εκτελούνται. Στο παραπάνω παράδειγμα, θεωρήσαμε μια ακολουθία εκτέλεσης με πέντε βήματα. Αρχίσαμε με το αρχικό μαρκάρισμα και τελειώσαμε σε ένα μαρκάρισμα, όπου οι τιμές των θέσεων NextSend και NextRec έχουν αυξηθεί σε 2. Για να φτάσουμε σε αυτό το μαρκάρισμα, χρησιμοποιήσαμε πέντε βήματα, καθένα από τα οποία περιείχε μία μοναδική καταχώρηση. Πρώτα, χρησιμοποιήσαμε μία καταχώρηση για τη μετάβαση SendPacket, μετά μία καταχώρηση για τη μετάβαση TransmitPacket, μία για τη μετάβαση ReceivePacket, μία για τη μετάβαση TransmitAcknowledgement και τέλος μία καταχώρηση για τη μετάβαση ReceiveAcknowledgement.

Υπάρχουν πολλές άλλες ακολουθίες εκτέλεσης. Όταν ένα μαρκάρισμα έχει αρκετά ενεργοποιημένα στοιχεία καταχώρησης, οποιοδήποτε μη-κενό και χωρίς συγκρούσεις υποσύνολο τους, μπορεί να επιλεγεί για το επόμενο βήμα. Αυτό σημαίνει ότι ένα Χρωματισμένο δίκτυο Petri παρουσιάζει μη-ντετερμινιστική συμπεριφορά. Για παράδειγμα, το Χρωματισμένο δίκτυο Petri που μοντελοποιεί το απλό πρωτόκολλο επικοινωνίας, ορίζει ότι μπορούν να γίνονται αναμεταδόσεις πακέτων, αλλά δεν παρέχει λεπτομέρειες σχετικά με το πότε και το πόσο συχνά μπορεί να γίνει κάποια αναμετάδοση. Η μετάβαση SendPacket είναι ενεργοποιημένη σε όλα τα προσιτά μαρκαρίσματα, εκτός από εκείνα στα οποία η τιμή της μάρκας στη θέση NextSend είναι μεγαλύτερη του αριθμού του τελευταίου πακέτου. Άρα οι αναμεταδόσεις πακέτων μπορούν να λάβουν χώρα οποιαδήποτε στιγμή και με οποιαδήποτε συχνότητα κι έτσι έχουμε ακολουθίες εκτέλεσης χωρίς αναμεταδόσεις (όπως η ακολουθία εκτέλεσης που θεωρήσαμε παραπάνω) και ακολουθίες εκτέλεσης με πολλές αναμεταδόσεις.

Με μια πρώτη ματιά, ίσως φανεί περίεργο που δεν καθορίζουμε τις συνθήκες κάτω από τις οποίες θα συμβεί μια αναμετάδοση. Για πολλούς λόγους όμως, αυτό δεν είναι απαραίτητο. Τα περισσότερα Χρωματισμένα δίκτυα Petri χρησιμοποιούνται για να εξεταστεί η λογική και λειτουργική ορθότητα ενός συστήματος. Για αυτό το λόγο, είναι συνήθως αρκετό να ορίζουμε ότι οι αναμεταδόσεις πακέτων είναι επιτρεπτές, γιατί, παραδείγματος χάριν, το δίκτυο είναι αργό. Δεν είναι απαραίτητο ούτε και ωφέλιμο να καθορίζουμε πόσο συχνά συμβαίνουν αναμεταδόσεις, γιατί το πρωτόκολλο πρέπει να ικανοποιεί όλα τα είδη των δικτύων, και αυτά που λειτουργούν τόσο καλά, ώστε να μην υπάρχουν αναμεταδόσεις και αυτά, στα οποία οι αναμεταδόσεις είναι συχνές. Στη συνέχεια, θα δούμε ότι τα Χρωματισμένα δίκτυα Petri μπορούν να επεκταθούν, εισάγοντας την έννοια του χρόνου, η οποία μας επιτρέπει να περιγράψουμε τη διάρκεια των δράσεων και των καταστάσεων και να εξετάσουμε την απόδοση του μοντελοποιημένου συστήματος, δηλαδή πόσο γρήγορα και αποτελεσματικά λειτουργεί. Τότε θα περιγράψουμε με μεγαλύτερη λεπτομέρεια τις αναμεταδόσεις πακέτων.

4.1.1 Χρωματισμένα Δίκτυα Petri και Δίκτυα Θέσης-Μετάβασης

Μπορεί να αποδειχθεί ότι κάθε Χρωματισμένο δίκτυο Petri μπορεί να μετασχηματιστεί σε ένα ισοδύναμο **δίκτυο Θέσης-Μετάβασης (Place-Transition net)** και αντιστρόφως. Εφ'όσον η εκφραστική δύναμη των δύο φορμαλισμών είναι ίδια, δεν υπάρχει κάποιο θεωρητικό όφελος από τη χρήση των Χρωματισμένων δικτύων Petri. Παρ'όλα αυτά, στην πράξη, τα Χρωματισμένα δίκτυα Petri αποτελούν μια πολύ πιο βολική γλώσσα μοντελοποίησης από τα δίκτυα Θέσης-Μετάβασης, όπως οι γλώσσες προγραμματισμού υψηλού επιπέδου είναι πολύ πιο κατάλληλες για προγραμματισμό από τις γλώσσες μηχανής και τις μηχανές Turing.

Επισυνάπτοντας μία τιμή σε κάθε μάρκα του Χρωματισμένου δικτύου Petri, χρησιμοποιούμε πολύ λιγότερες θέσεις από αυτές που χρειάζονται σε ένα δίκτυο Θέσης-Μετάβασης. Ενώ σε ένα Χρωματισμένο δίκτυο Petri μπορούμε να επισυνάψουμε διαφορετική τιμή σε κάθε μάρκα, στα δίκτυα Θέσης-Μετάβασης, ο μόνος τρόπος για να ξεχωρίσουμε τις μάρκες είναι να τις τοποθετήσουμε σε διαφορετικές θέσεις. Επίσης, όταν ένα Χρωματισμένο δίκτυο Petri χρησιμοποιεί σύνθετους τύπους, όπως ακέραιους ή πραγματικούς αριθμούς, εγγραφές και λίστες, το ισοδύναμο δίκτυο Θέσης-Μετάβασης, συχνά, έχει έναν άπειρο ή αστρονομικά μεγάλο αριθμό θέσεων.

Η χρήση μεταβλητών στις επιγραφές τόξου σημαίνει, ότι κάθε μετάβαση στα Χρωματισμένα δίκτυα Petri, μπορεί να εκτελεστεί με διαφορετικές καταχωρήσεις, όπως οι **διαδικασίες (procedures)** μπορούν να εκτελεστούν με διαφορετικές **παράμετρους (parameters)**. Έτσι, μπορούμε να χρησιμοποιήσουμε μία μόνο μετάβαση για να περιγράψουμε μία κλάση από σχετικές δράσεις, ενώ στα δίκτυα Θέσης-Μετάβασης χρειαζόμαστε μία μετάβαση για κάθε δράση. Παραδείγματος χάριν, το Χρωματισμένο δίκτυο Petri που μοντελοποιεί το πρωτόκολλο επικοινωνίας, έχει μόνο μία μετάβαση SendPacket. Η μετάβαση αυτή μπορεί να χειριστεί όλα τα πακέτα, παρ'όλο που έχουν διαφορετικούς αριθμούς πακέτων και διαφορετικό περιεχόμενο.

4.2 Προσομοίωση των Χρωματισμένων Δικτύων Petri

Καθώς δομούνται τα τμήματα ενός Χρωματισμένου δικτύου Petri, ελέγχονται και διορθώνονται με τη βοήθεια ενός **προσομοιωτή (simulator)**, όπως ακριβώς ένας προγραμματιστής ελέγχει και διορθώνει τα νέα τμήματα ενός προγράμματος. Όπως φαίνεται και από τα παραδείγματα στην Παράγραφο 1, αυτός που μοντελοποιεί ένα σύστημα, μπορεί να επιθεωρεί όλες τις λεπτομέρειες των προσιτών μαρκαρισμάτων. Μπορεί, για παράδειγμα, να δει το σύνολο των ενεργοποιημένων μεταβάσεων και να επιλέξει ποιο στοιχείο καταχώρησης θα εκτελεστεί στη συνέχεια.

Αυτός ο τρόπος είναι ο πιο χειρωνακτικός και **διαδραστικός (interactive)** τρόπος προσομοίωσης και είναι, εκ φύσεως πολύ αργός, αφού κανένας άνθρωπος δεν μπορεί να εξετάσει περισσότερα από δυο-τρία μαρκαρίσματα το λεπτό. Αυτός ο τρόπος εργασίας είναι παρόμοιος με την **αποσφαλμάτωση βήμα-προς-βήμα (single step debugging)** σε μια κοινή γλώσσα προγραμματισμού και χρησιμοποιείται συχνά για την πρώτη εξέταση ενός νέου Χρωματισμένου δικτύου Petri ή των νέων τμημάτων ενός μεγάλου μοντέλου, για να εξακριβωθεί αν κάθε συστατικό στοιχείο του δικτύου λειτουργεί σωστά.

Είναι τυπικό, να χρησιμοποιούνται στη συνέχεια άλλα είδη προσομοίωσης. Παραδείγματος χάριν, ο προσομοιωτής μπορεί να επιλέγει μεταξύ των ενεργοποιημένων στοιχείων καταχώρησης, με τη βοήθεια μιας γεννήτριας τυχαίων αριθμών κι έτσι να έχουμε πολύ πιο γρήγορες προσομοιώσεις. Μία εντελώς αυτόματη προσομοίωση εκτελείται με ταχύτητα αρκετών χιλιάδων βημάτων το δευτερόλεπτο, η οποία εξαρτάται από το μέγεθος του Χρωματισμένου δικτύου Petri και από την ισχύ του υπολογιστή στον οποίο τρέχει ο προσομοιωτής. Είναι προφανές ότι κανένας άνθρωπος δεν είναι ικανός, παρακολουθώντας το Χρωματισμένο δίκτυο Petri και τα μαρκαρίσματα του, να παρατηρήσει τις λεπτομέρειες μιας τέτοιας προσομοίωσης. Γι'αυτό το λόγο τα αποτελέσματα της προσομοίωσης πρέπει να παρουσιάζονται με κάποιο άλλο τρόπο. Ένας άμεσος τρόπος παρουσίασης των αποτελεσμάτων είναι η **αναφορά προσομοίωσης (simulation report)**, η οποία είναι ένα αρχείο κειμένου που περιέχει λεπτομερείς πληροφορίες για όλα τα εκτελούμενα στοιχεία καταχώρησης.

Στο Σχήμα 9 φαίνεται μία αναφορά προσομοίωσης των πρώτων 15 βημάτων του μοντέλου του πρωτοκόλλου επικοινωνίας. Το "A" που ακολουθεί τον αριθμό του βήματος υποδεικνύει ότι το στοιχείο καταχώρησης εκτελέστηκε με αυτόματο τρόπο. Οι πληροφορίες μετά το σύμβολο @ προσδιορίζουν τη **σελίδα (page)**, δηλαδή το τμήμα του Χρωματισμένου δικτύου Petri, που ανήκει η εκτελούμενη μετάβαση. Για το πρωτόκολλο επικοινωνίας, η πληροφορία αυτή δεν είναι πολύ ενδιαφέρουσα, γιατί το μοντέλο είναι πολύ μικρό και έχει μόνο μία σελίδα. Αντίθετα, για ένα μεγάλο, σύνθετο μοντέλο, η πληροφορία αυτή είναι πολύ χρήσιμη.

```

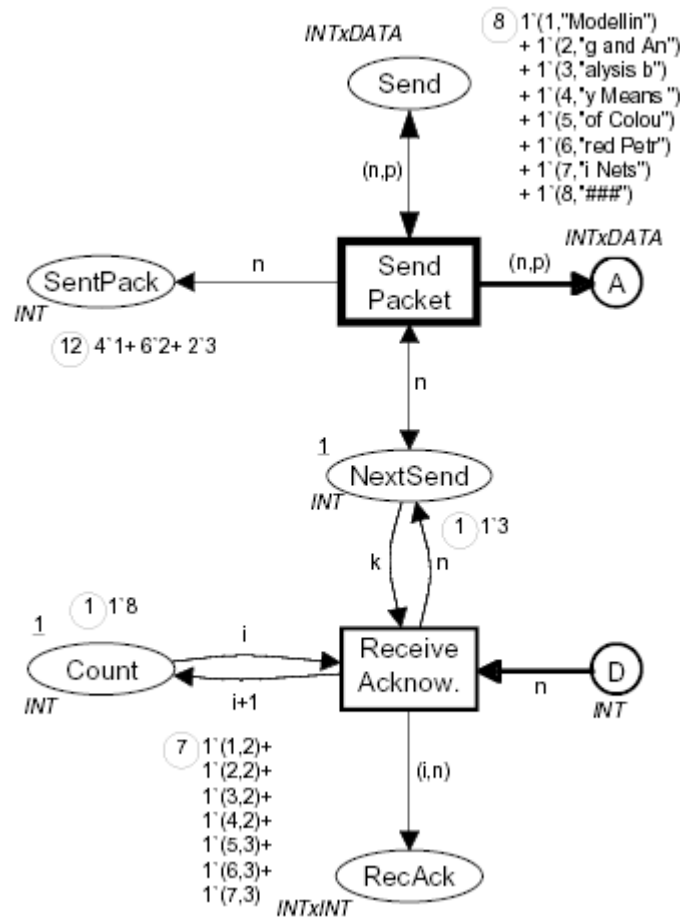
1 A SendPack@(1:Top#1)
  { n = 1, p = "Modellin"}
2 A SendPack@(1:Top#1)
  { n = 1, p = "Modellin"}
3 A TranPack@(1:Top#1)
  { n = 1, p = "Modellin", r = 2, s = 8}
4 A TranPack@(1:Top#1)
  { n = 1, p = "Modellin", r = 6, s = 8}
5 A RecPack@(1:Top#1)
  { k = 1, n = 1, p = "Modellin", str = ""}
6 A SendPack@(1:Top#1)
  { n = 1, p = "Modellin"}
7 A TranPack@(1:Top#1)
  { n = 1, p = "Modellin", r = 5, s = 8}
8 A RecPack@(1:Top#1)
  { k = 2, n = 1, p = "Modellin", str = "Modellin"}
9 A SendPack@(1:Top#1)
  { n = 1, p = "Modellin"}
10 A TranAck@(1:Top#1)
  { n = 2, r = 3, s = 8}
11 A TranAck@(1:Top#1)
  { n = 2, r = 4, s = 8}
12 A RecAck@(1:Top#1)
  { k = 1, n = 2}
13 A TranPack@(1:Top#1)
  { n = 1, p = "Modellin", r = 3, s = 8}
14 A RecPack@(1:Top#1)
  { k = 2, n = 1, p = "Modellin", str = "Modellin"}
15 A SendPack@(1:Top#1)
  { n = 2, p = "g and An"}

```

Σχήμα 9: Αναφορά προσομοίωσης

Ένας άλλος τρόπος για να καταγράψουμε τα αποτελέσματα μιας προσομοίωσης είναι, να προσθέσουμε στο Χρωματισμένο δίκτυο Petri, **θέσεις αναφοράς (report places)**. Οι θέσεις αυτές, συλλέγουν πληροφορίες σχετικά με την εκτέλεση της προσομοίωσης, χωρίς να την επηρεάζουν.

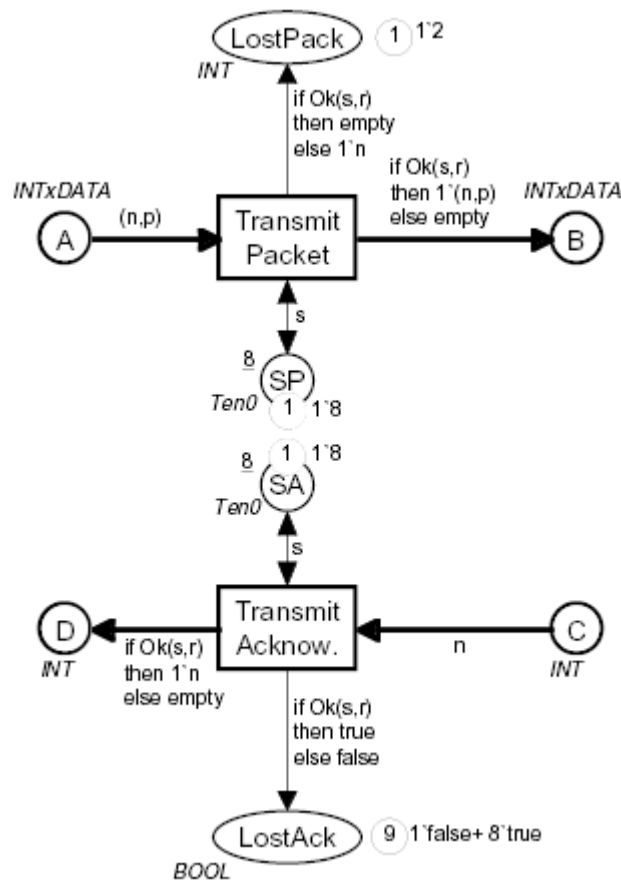
Στο Σχήμα 10, φαίνεται ο αποστολέας Sender του πρωτοκόλλου επικοινωνίας, στον οποίο προσθέσαμε τρεις θέσεις αναφοράς. Η θέση SendPack καταγράφει πόσες φορές έχει αποσταλεί κάθε πακέτο. Στο παράδειγμα μας, το πακέτο νούμερο ένα στάλθηκε τέσσερις φορές, το πακέτο νούμερο δύο, έξι φορές και το πακέτο νούμερο τρία, δύο φορές. Η θέση RecAck μας πληροφορεί για το ποιες βεβαιώσεις λήψης πακέτων έχουν παραληφθεί από τον αποστολέα Sender. Κάθε βεβαίωση λήψης πακέτου καταγράφεται ως ένα ζεύγος στοιχείων, όπου το πρώτο στοιχείο είναι ένας **αριθμός διαδοχής (sequence number)**, ενώ το δεύτερο στοιχείο είναι το περιεχόμενο της βεβαίωσης. Ο αριθμός διαδοχής προσκομείται από τη θέση Count. Στο παράδειγμα μας, αρχικά παραλήφθηκαν τέσσερες βεβαιώσεις λήψης πακέτων με τιμή 2 και στη συνέχεια, τρεις βεβαιώσεις λήψης πακέτων με τιμή 3.



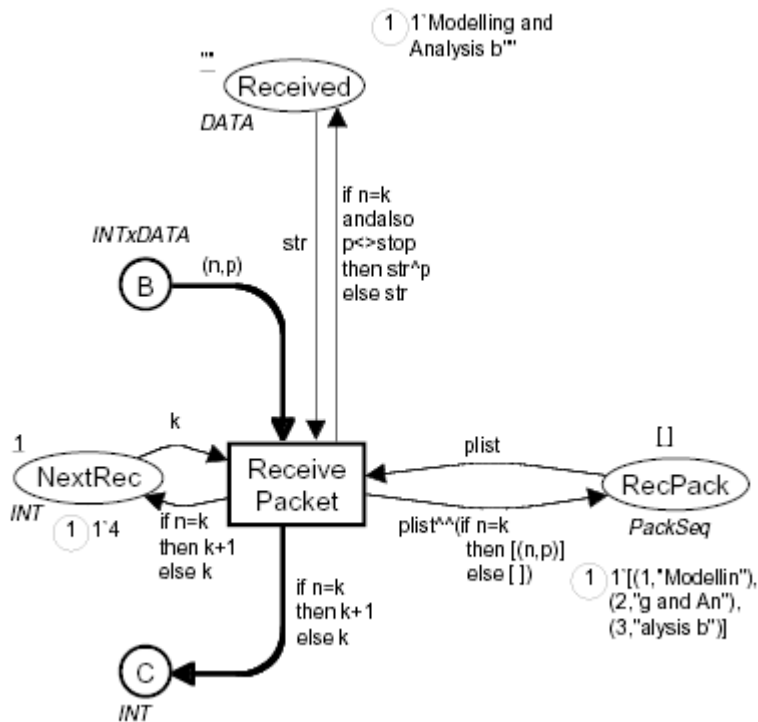
Σχήμα 10: Οι θέσεις αναφοράς για τον αποστολέα Sender του πρωτοκόλλου επικοινωνίας

Στο Σχήμα 11, απεικονίζεται το δίκτυο Network του πρωτοκόλλου επικοινωνίας, στο οποίο προσθέσαμε δύο θέσεις αναφοράς. Η θέση *LostPack* καταγράφει τον αριθμό των χαμένων πακέτων, ενώ η θέση *LostAck* μας πληροφορεί για το πόσες βεβαιώσεις λήψης πακέτων μεταδόθηκαν και πόσες χάθηκαν. Στο παράδειγμα μας, χάνεται μόνο ένα αντίγραφο του πακέτου νούμερο δύο, ενώ μεταδίδονται επιτυχώς οχτώ βεβαιώσεις λήψης πακέτων και χάνεται μία.

Στο Σχήμα 12, προσθέσαμε στον αποστολέα Receiver του πρωτοκόλλου επικοινωνίας, τη θέση αναφοράς *RecPack*, η οποία καταγράφει τα πακέτα που έχουν παραληφθεί με επιτυχία. Ο τύπος *PackSeq* της *RecPack*, είναι μια λίστα όλων των τιμών *INTxDATA*. Ο τελεστής \wedge στην επιγραφή του τόξου που εξέρχεται από τη θέση *ReceivePacket* και καταλήγει στη θέση *RecPack*, ενώνει δύο λίστες. Στο παράδειγμα μας, έχουν παραληφθεί τρία πακέτα : πρώτα, το πακέτο (1, "Modellin"), μετά το πακέτο (2, "g and An") και τέλος το πακέτο (3, "alysis b").



Σχήμα 11: Οι θέσεις αναφοράς για το δίκτυο Network του πρωτοκόλλου επικοινωνίας

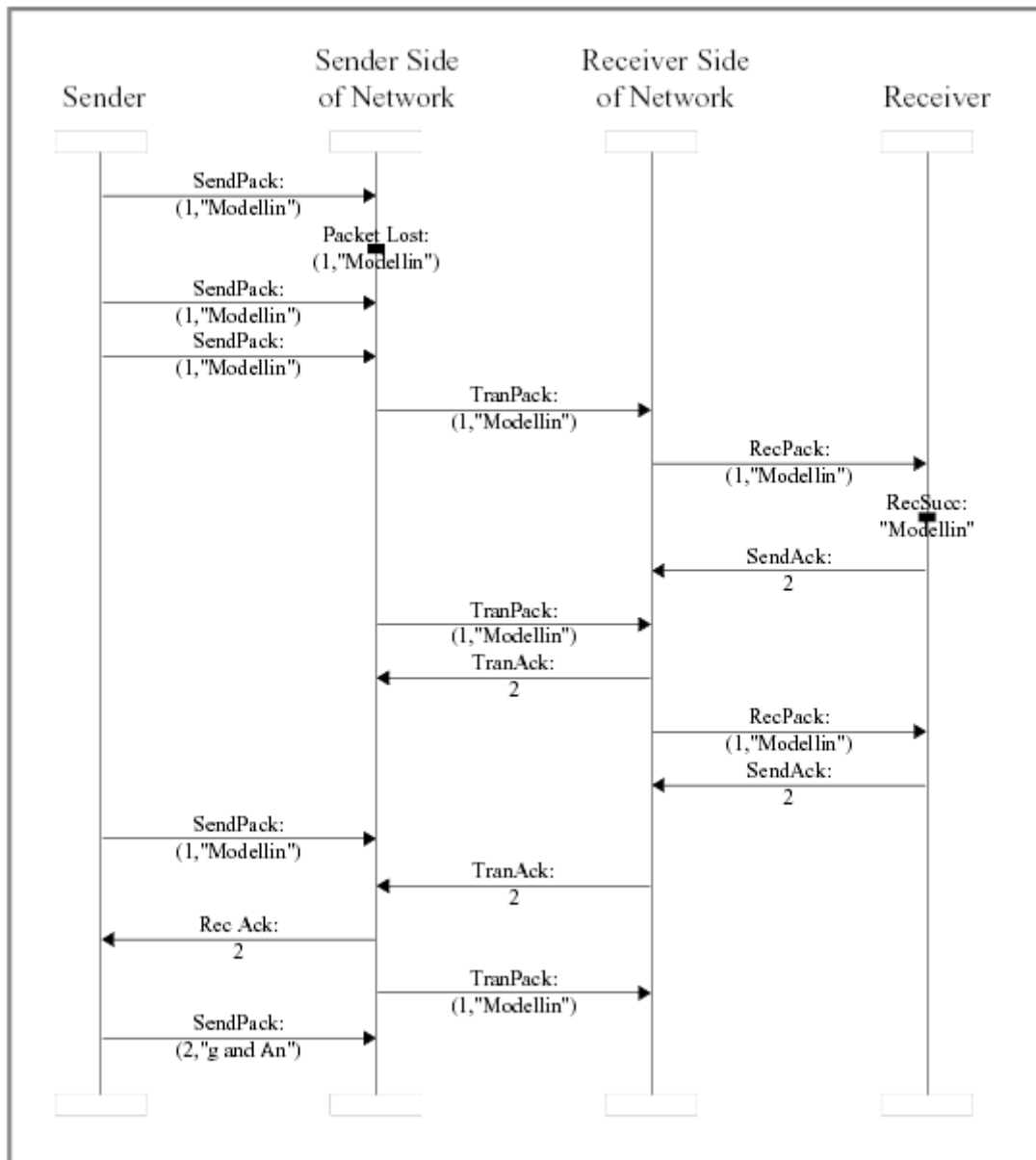


Σχήμα 12: Οι θέσεις αναφοράς για τον παραλήπτη Receiver του πρωτοκόλλου επικοινωνίας

Ένας τρίτος τρόπος καταγραφής των αποτελεσμάτων μιας προσομοίωσης είναι η χρήση ενός **διαγράμματος διαδοχής μηνυμάτων (message sequence chart)**, το οποίο παρέχει γραφικά, μια συνολική εικόνα των δράσεων σε ένα Χρωματισμένο δίκτυο Petri. Στο Σχήμα 13 φαίνεται ένα παράδειγμα ενός διαγράμματος διαδοχής μηνυμάτων για το πρωτόκολλο επικοινωνίας.

Μια εκτέλεση της μετάβασης `SendPacket` υποδεικνύεται από ένα οριζόντιο βέλος μεταξύ της πρώτης και της δεύτερης κατακόρυφης γραμμής. Το βέλος έχει ως επιγραφή το πακέτο που μεταδίδεται στο δίκτυο `Network`. Ανάλογα, μια επιτυχής εκτέλεση της μετάβασης `TransmitPacket` υποδεικνύεται από ένα οριζόντιο βέλος μεταξύ της δεύτερης και της τρίτης κατακόρυφης γραμμής. Αν όμως το πακέτο χαθεί, υπάρχει στη δεύτερη γραμμή μόνο ένα μικρό ορθογώνιο. Μια εκτέλεση της μετάβασης `ReceivePacket` υποδεικνύεται από δύο βέλη : ένα βέλος για το πακέτο που παραλήφθηκε και ένα βέλος για τη βεβαίωση λήψης πακέτου που αποστέλλεται. Αν το πακέτο που παραλήφθηκε είναι το σωστό, τότε υπάρχει επίσης, ένα μικρό ορθογώνιο μεταξύ των δύο βελών. Οι εκτελέσεις της μετάβασης `TransmitAcknowledgement` υποδεικνύονται παρόμοια με τις εκτελέσεις της μετάβασης `TransmitPacket`, με τη διαφορά ότι τα βέλη σχεδιάζονται με διεύθυνση από δεξιά προς τα αριστερά και τα ορθογώνια είναι τοποθετημένα στη τρίτη κατακόρυφη γραμμή. Τέλος οι εκτελέσεις της μετάβασης `ReceiveAcknowledgement` υποδεικνύονται παρόμοια με τις εκτελέσεις της μετάβασης `SendPacket`, με τα βέλη όμως να δείχνουν από δεξιά προς τα αριστερά.

Στο Σχήμα 13 υπάρχει ένα τόξο ή ένα ορθογώνιο για κάθε βήμα. Αυτό σημαίνει ότι ένα διάγραμμα διαδοχής μηνυμάτων περιέχει όλες τις πληροφορίες που υπάρχουν σε μια αναφορά προσομοίωσης. Συνήθως όμως, καταγράφονται μόνο οι δράσεις που παίζουν κρίσιμο ρόλο στην εκτέλεση του δικτύου, όπως είναι η μετάδοση των πακέτων και των βεβαιώσεων λήψης πακέτων. Με αυτό τον τρόπο παίρνουμε τη συνοπτική εικόνα μιας εκτενούς προσομοίωσης. Με το διαχωρισμό των σημαντικών δράσεων και την αναπαράστασή τους με γραφικό τρόπο, μπορούμε πολύ πιο γρήγορα να ερμηνεύσουμε τα αποτελέσματα μιας προσομοίωσης και να εξακριβώσουμε αν ένα Χρωματισμένο δίκτυο Petri συμπεριφέρεται όπως ήταν αναμενόμενο ή όχι. Στην τελευταία περίπτωση, μπορούμε να χρησιμοποιήσουμε είτε διαδραστικές προσομοιώσεις, είτε τις αναφορές προσομοίωσης, ώστε να ανακαλύψουμε τους λόγους που το δίκτυο παρουσιάζει διαφορετική συμπεριφορά από την αναμενόμενη.



Σχήμα 13: Διάγραμμα διαδοχής μηνυμάτων

4.2.1 Τμήματα Κώδικα (Code Segments)

Τα διαγράμματα διαδοχής μηνυμάτων δημιουργούνται με τη χρήση **τμημάτων κώδικα (code segments)**, τα οποία επισυνάπτονται στις μεταβάσεις. Αυτά τα τμήματα κώδικα καλούν συναρτήσεις μιας βιβλιοθήκης, η οποία παρέχεται μαζί με τον προσομοιωτή των Χρωματισμένων δικτύων Petri. Όταν εκτελείται μια μετάβαση, εκτελείται και το αντίστοιχο τμήμα κώδικα. Μπορεί για παράδειγμα, να διαβάζει και να γράφει αρχεία κειμένου, να δημιουργεί τα γραφικά ή ακόμη, να υπολογίζει τις τιμές που πρέπει να τεθούν ως όρια σε κάποιες μεταβλητές της μετάβασης. Με αυτόν τον τρόπο τα τμήματα κώδικα παρέχουν μια πολύ εύχρηστη και βολική **διαπροσωπεία (interface)**, μεταξύ του μοντέλου του Χρωματισμένου δικτύου Petri και του περιβάλλοντος του, όπως είναι, παραδείγματος χάριν, το σύστημα αρχείων.

Για την **ενημέρωση (update)** του διαγράμματος διαδοχής μηνυμάτων του Σχήματος 13, επισυνάπτουμε στις μεταβάσεις SendPacket και TransmitPacket, τα τμήματα κώδικα που φαίνονται στα Σχήματα 14 και 15, αντίστοιχα. Το MSCdiagram είναι ένας δείκτης προς το διάγραμμα διαδοχής μηνυμάτων, ενώ το mkst_col'INTxDATA είναι μια συνάρτηση που παριστάνει την τιμή (n, p) ως συμβολοσειρά. Τα τμήματα κώδικα των υπόλοιπων τριών μεταβάσεων είναι παρόμοια.

```
input (n,p);
action
  MSC.Message (!MSCdiagram)
    {sender = "sender",
     receiver = "sendernet",
     label = "SendPack:"^NEWLINE^
           (mkst_col'INTxDATA(n,p))};
```

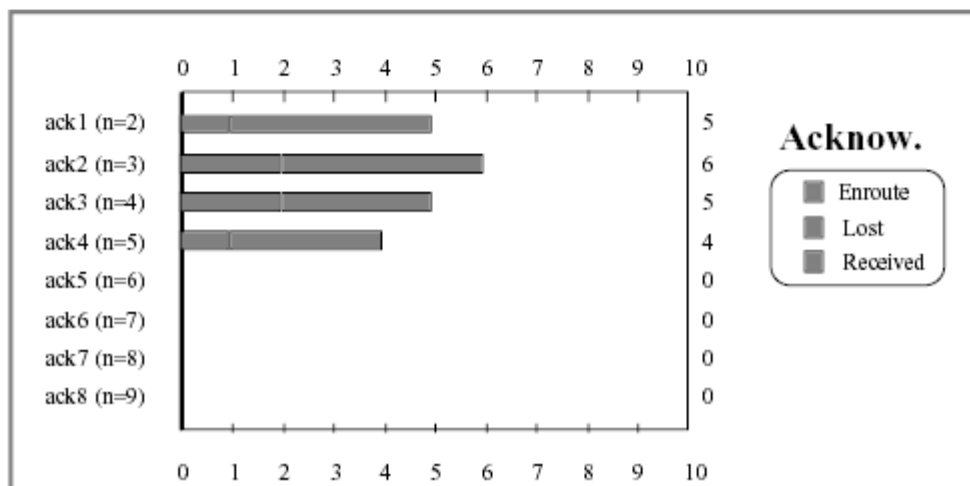
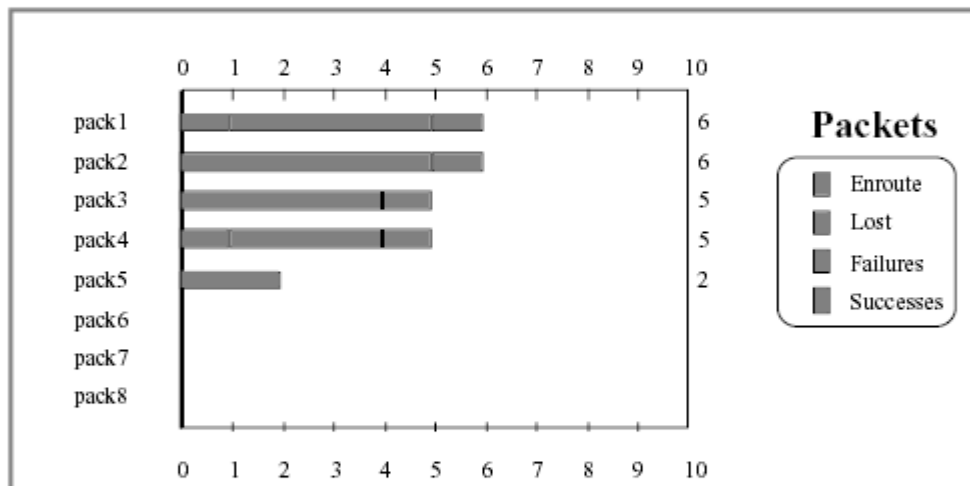
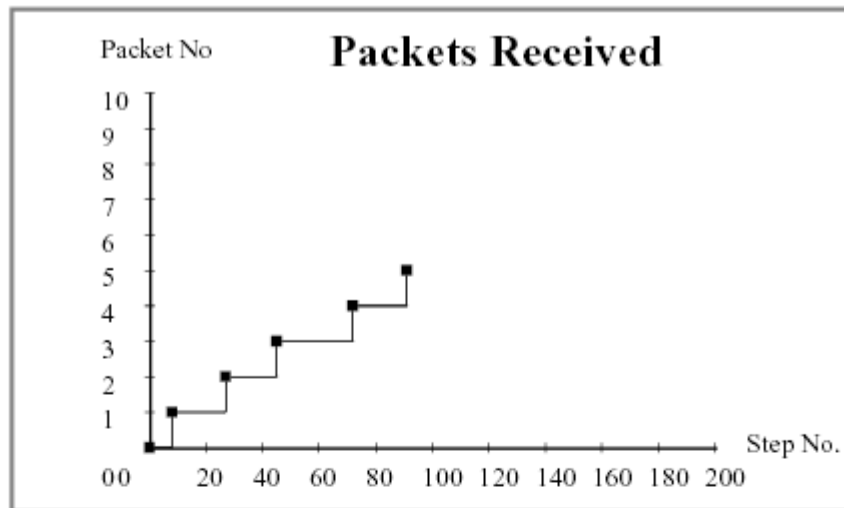
Σχήμα 14: Το τμήμα κώδικα που επισυνάπτεται στη μετάβαση SendPacket

```
input (n,p,s,r);
action
  if Ok(s,r) then
    MSC.Message (!MSCdiagram)
      {sender = "sendernet",
       receiver = "receivernet",
       label = "TranPack:"^NEWLINE^
             (mkst_col'INTxDATA(n,p))}
  else
    MSC.Processmark (!MSCdiagram)
      {process = "sendernet",
       label = "Packet Lost:"^NEWLINE^
             (mkst_col'INTxDATA(n,p))};
```

Σχήμα 15: Το τμήμα κώδικα που επισυνάπτεται στη μετάβαση TransmitPacket

Τα τμήματα κώδικα μπορούν επίσης να χρησιμοποιηθούν για την ενημέρωση και άλλων ειδών διαγραμμάτων. Για το απλό πρωτόκολλο επικοινωνίας, μπορούμε να χρησιμοποιήσουμε τα τρία διαγράμματα που φαίνονται στο Σχήμα 16.

Το πρώτο διάγραμμα είναι ένα διάγραμμα γραμμής (line chart), το οποίο δείχνει, πόσο γρήγορα παραλαμβάνεται επιτυχώς, κάθε πακέτο, ως συνάρτηση του αριθμού βήματος. Από το διάγραμμα γραμμής του Σχήματος 16, μπορούμε να δούμε ότι το πακέτο νούμερο ένα παραλήφθηκε πριν από τα δέκα βήματα, το πακέτο νούμερο δύο μετά από περίπου εικοσιπέντε βήματα, το πακέτο νούμερο τρία μετά από περίπου σαρανταπέντε βήματα και ούτω καθεξής. Το διάγραμμα γραμμής ενημερώνεται μέσω ενός τμήματος κώδικα στη θέση Receive, κάθε φορά που ένα πακέτο παραλαμβάνεται επιτυχώς.



Σχήμα 16: Ένα διάγραμμα γραμμής και δύο ραβδογράμματα

Το δεύτερο διάγραμμα είναι ένα ραβδόγραμμα, το οποίο μας πληροφορεί για το πόσες φορές έχει αποσταλεί κάθε πακέτο και με τι αποτέλεσμα. Από το δεύτερο

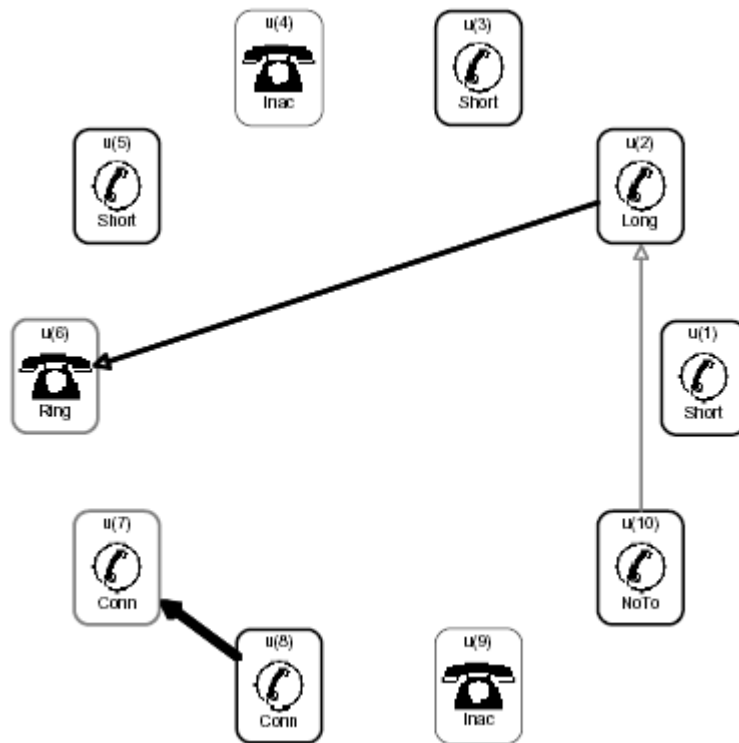
ραβδόγραμμα του Σχήματος 16, φαίνεται ότι το πακέτο νούμερο ένα έχει αποσταλεί έξι φορές. Μία φορά χάθηκε, τέσσερις φορές παραλήφθηκε ως **αποτυχία (failure)**, δηλαδή σε λάθος σειρά, και την τελευταία παραλήφθηκε με επιτυχία. Ομοίως, το πακέτο νούμερο δύο φαίνεται ότι έχει αποσταλεί έξι φορές, ενώ τα πακέτα νούμερο τρία και τέσσερα έχουν αποσταλεί από πέντε φορές το καθένα. Τέλος, το πακέτο νούμερο πέντε έχει αποσταλεί δύο φορές, και αμφότερες είναι en route, δηλαδή το πακέτο βρίσκεται στη θέση A ή στη θέση B του δικτύου Network. Το τρίτο ραβδόγραμμα είναι παρόμοιο με το δεύτερο, αλλά δείχνει την πρόοδο των βεβαιώσεων λήψης πακέτων. Τα δύο ραβδογράμματα ενημερώνονται περιοδικά, σε τακτά χρονικά διαστήματα, παραδείγματος χάριν σε κάθε πενήντα βήματα, τα οποία ορίζονται από αυτόν που μοντελοποιεί το σύστημα.

Τα τρία διαγράμματα του Σχήματος 16 μας δίνουν πολλές χρήσιμες πληροφορίες σχετικά με τη συμπεριφορά του πρωτοκόλλου. Για παράδειγμα, γίνεται προφανές ότι οι αποτυχίες, δηλαδή η λάθος σειρά των πακέτων, συχνά προκαλούν περισσότερες αναμεταδόσεις από τα χαμένα πακέτα. Είναι εύκολο να δούμε επίσης, ότι χρειάζονται περισσότερα από ενενήντα βήματα για να μεταδοθούν επιτυχώς τα πρώτα πέντε πακέτα, ενώ σε ένα δίκτυο που λειτουργεί τέλεια, αυτό θα ήταν εφικτό μέσα σε εικοσιπέντε βήματα.

Ακόμη, τα τμήματα κώδικα μπορούν να χρησιμοποιηθούν για τη δημιουργία και την ενημέρωση πιο εξειδικευμένων αναπαραστάσεων ενός μαρκαρίσματος, δηλαδή της κατάστασης ενός μοντελοποιημένου συστήματος. Το Σχήμα 17 παριστάνει την κατάσταση ενός απλού πρωτοκόλλου τηλεφώνου, το οποίο περιέχει δέκα διαφορετικά τηλέφωνα. Για κάθε τηλέφωνο απεικονίζεται η κατάσταση του, παραδείγματος χάριν, το τηλέφωνο u(9) βρίσκεται στη κατάσταση Inactive, δηλαδή είναι απενεργοποιημένο, το u(6) στη κατάσταση Ringing, δηλαδή κουδουνίζει και τα τηλέφωνα u(7) και u(8) βρίσκονται σε κατάσταση Connected, δηλαδή είναι συνδεδεμένα.

Επιπλέον απεικονίζονται οι σχέσεις μεταξύ των τηλεφώνων. Ένα λεπτό, διακεκομμένο βέλος υποδεικνύει ότι έχει γίνει αίτηση για σύνδεση, δηλαδή ότι έχει πληκτρολογηθεί ένας αριθμός. Το τηλέφωνο u(10) που καλεί, δεν έχει τόνο από την άλλη άκρη της τηλεφωνικής γραμμής (No Tone), ενώ το τηλέφωνο u(2) που καλείται, μπορεί να βρίσκεται σε οποιαδήποτε κατάσταση. Ένα πιο παχύ μη-διακεκομμένο βέλος υποδεικνύει ότι η αίτηση έχει γίνει δεκτή και ότι γίνεται προσπάθεια σύνδεσης. Το τηλέφωνο u(2) που καλεί, έχει τόνο από την άλλη άκρη της τηλεφωνικής γραμμής, με μεγάλα ενδιάμεσα διαστήματα μεταξύ των κουδουνισμάτων (Long), ενώ το καλούμενο τηλέφωνο u(6), κουδουνίζει (Ringing). Τέλος, ένα παχύ βέλος όπως αυτό που συνδέει τα τηλέφωνα u(8) και u(7), υποδεικνύει ότι έχει γίνει σύνδεση μεταξύ των τηλεφώνων, τα οποία είναι αμφότερα, συνδεδεμένα (Connected).

Όπως προαναφέρθηκε, τα τμήματα κώδικα μπορούν να χρησιμοποιηθούν για την εγγραφή αρχείων κειμένου, τα οποία με τη βοήθεια κατάλληλων προγραμμάτων, μπορούν να μας δώσουν υψηλής ποιότητας αναπαραστάσεις των σύνθετων αποτελεσμάτων της προσομοίωσης. Επίσης, τα τμήματα κώδικα μπορούν να χρησιμοποιηθούν για την ανάγνωση αρχείων κειμένου. Η ικανότητα τους αυτή, χρησιμοποιείται παραδείγματος χάριν, για την αρχικοποίηση ενός μοντέλου. Με αυτόν τον τρόπο, καθίσταται δυνατή η αλλαγή του αρχικού μαρκαρίσματος, χωρίς να τροποποιηθεί το ίδιο το μοντέλο του Χρωματισμένου δικτύου Petri.



Σχήμα 17: Μια πιο εξειδικευμένη αναπαράσταση της κατάστασης ενός συστήματος

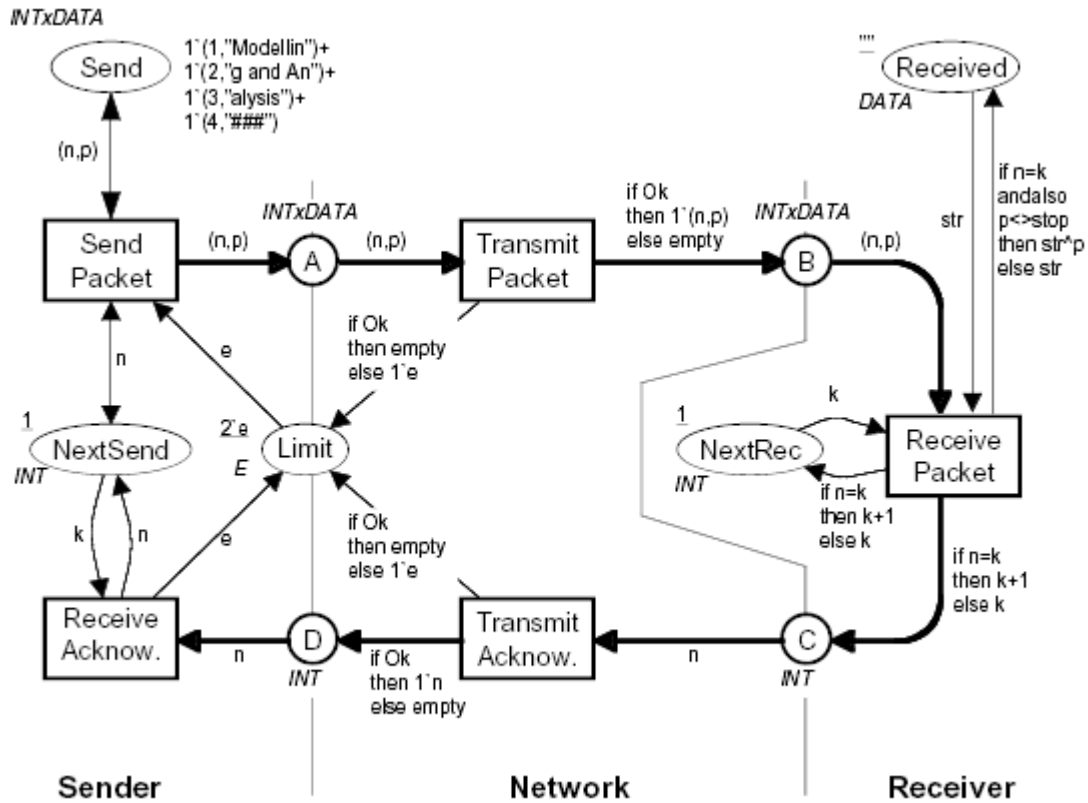
4.3 Ανάλυση του Χώρου Καταστάσεων των Χρωματισμένων Δικτύων Petri

Όπως αναφέρθηκε στην Παράγραφο 4.2, είναι σύνηθες, να εξετάζουμε και να διορθώνουμε ένα Χρωματισμένο δίκτυο Petri με τη βοήθεια της προσομοίωσης. Η προσομοίωση όμως, δεν μπορεί να ποτέ να αποδείξει την ορθότητα ενός μοντέλου, εκτός κι αν το μοντέλο είναι πολύ απλό. Έτσι, συνδυάζουμε συχνά την προσομοίωση με την κατασκευή ενός ή περισσότερων χώρων καταστάσεων.

Η βασική ιδέα πίσω από το **χώρο καταστάσεων (state space)** είναι η κατασκευή ενός γράφου, ο οποίος έχει ένα κόμβο για κάθε προσιτό μαρκάρισμα και ένα τόξο για κάθε στοιχείο καταχώρησης που εκτελείται. Οι χώροι καταστάσεων ονομάζονται επίσης **γράφοι εκτέλεσης (occurrence graphs)** ή **γράφοι/δένδρα προσιτότητας (reachability graphs/trees)**. Το πρώτο από αυτά τα ονόματα αντικατοπτρίζει το γεγονός, ότι ένας χώρος καταστάσεων περιέχει όλες τις πιθανές ακολουθίες εκτέλεσης, ενώ τα δύο επόμενα ονόματα αντικατοπτρίζουν το γεγονός, ότι ένας χώρος καταστάσεων περιέχει όλα τα προσιτά μαρκάρια. Για τους γράφους και τα δένδρα προσιτότητας έχει γίνει εκτενής αναφορά στα προηγούμενα κεφάλαια.

Για να μπορέσουμε να κατασκευάσουμε ένα χώρο καταστάσεων που να έχει λογικό μέγεθος, συχνά πρέπει να τροποποιήσουμε το Χρωματισμένο δίκτυο Petri. Στο απλό πρωτόκολλο επικοινωνίας κάνουμε τρεις τροποποιήσεις. Πρώτον, μειώνουμε τον αριθμό των πακέτων από οχτώ σε τέσσερα. Δεύτερον, εισάγουμε μία νέα θέση με το όνομα Limit, για να περιορίσουμε τον αριθμό των πακέτων και των βεβιασέων λήψης πακέτων που μπορούν να υπάρχουν ταυτόχρονα στο δίκτυο Network, δηλαδή στις θέσεις A, B, C και D. Η νέα θέση είναι τύπου E με μία μόνο πιθανή τιμή, την e.

Αυτό σημαίνει ότι οι μάρκες αυτού του τύπου δεν φέρουν δεδομένα. Τέλος, απλοποιούμε το **μηχανισμό απόφασης (decision mechanism)** για τη μετάδοση των πακέτων και των βεβαιώσεων λήψης. Για τους χώρους καταστάσεων δεν έχει νόημα, τα πακέτα να μεταδίδονται ή να χάνονται με συγκεκριμένη πιθανότητα. Έτσι, αντικαθιστούμε τη συνάρτηση Ok με τη boolean μεταβλητή Ok. Το τροποποιημένο Χρωματισμένο δίκτυο Petri φαίνεται στο Σχήμα 18, ενώ οι δηλώσεις του φαίνονται στο Σχήμα 19.



Σχήμα 18: Το τροποποιημένο Χρωματισμένο δίκτυο Petri κατάλληλο για την ανάλυση του χώρου καταστάσεων

```

color INT = int;
color DATA = string;
color INTxDATA = product INT * DATA;
var n, k : INT;
var p, str : DATA;
val stop = "###";

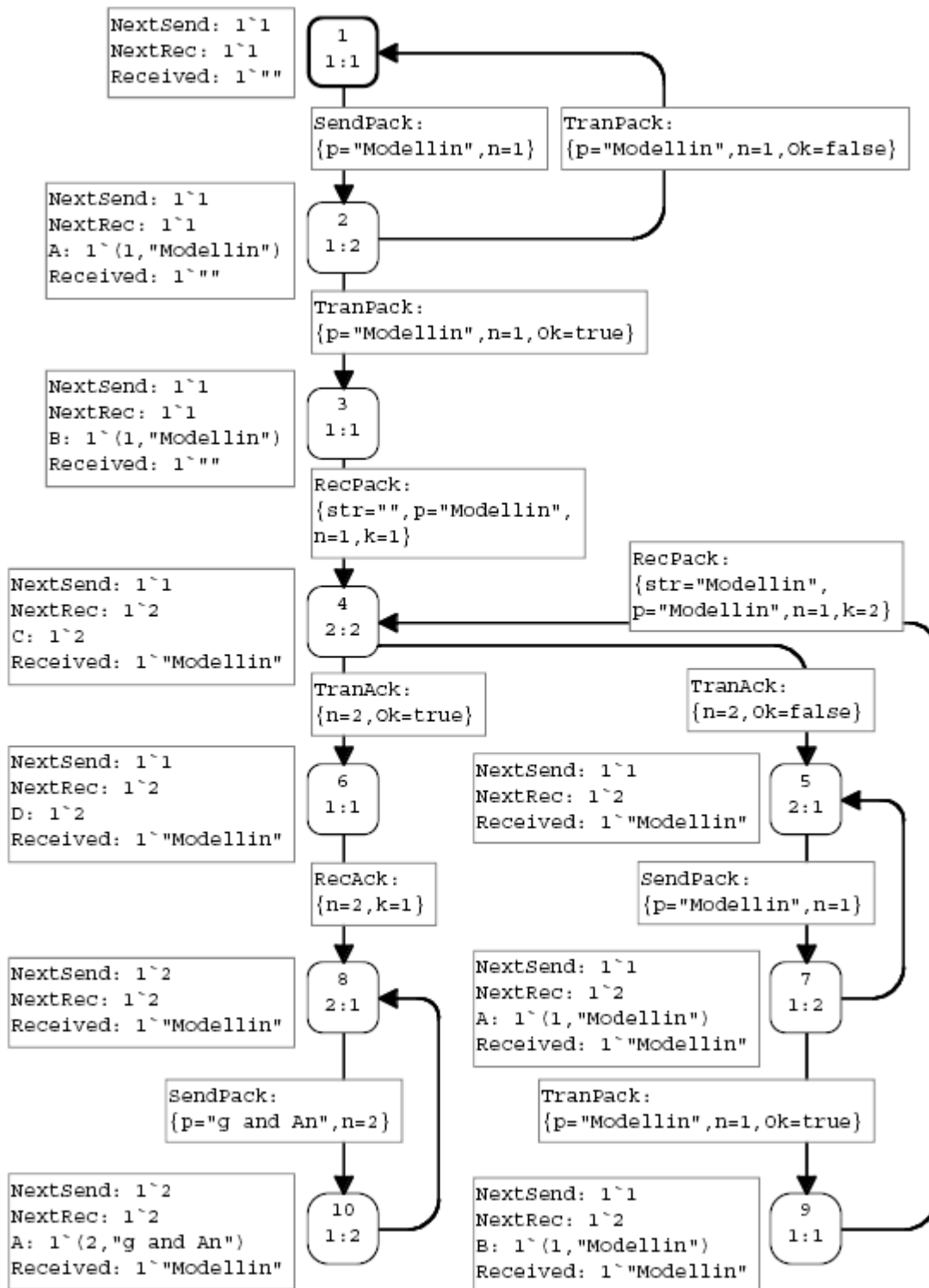
color E = with e;
color BOOL = bool;
var Ok : BOOL;

```

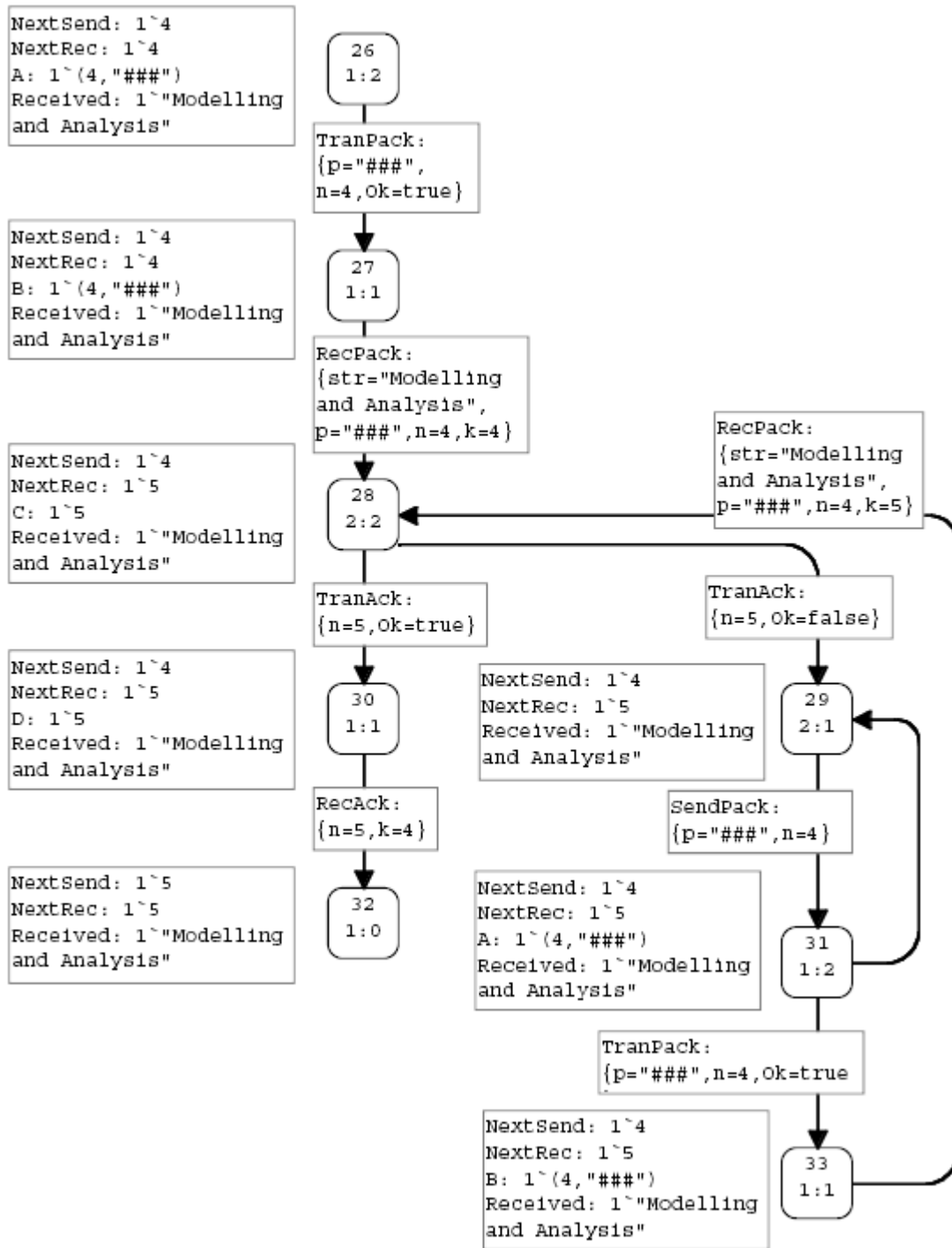
Σχήμα 19: Οι δηλώσεις Χρωματισμένου δικτύου Petri του Σχήματος 18

Αφού κάνουμε τις τροποποιήσεις που περιγράψαμε παραπάνω, είμαστε έτοιμοι να κατασκευάσουμε τους χώρους καταστάσεων. Ας ξεκινήσουμε με την περίπτωση όπου το αρχικό μαρκάρισμα της θέσης Limit είναι 1'e. Αυτό σημαίνει ότι το δίκτυο Network (δηλαδή οι θέσεις A, B, C και D) περιέχει το πολύ ένα πακέτο ή μια

βεβαίωση λήψης πακέτου κάθε στιγμή. Άρα δεν είναι δυνατό ένα πακέτο να προσπεράσει κάποιο άλλο. Ο χώρος καταστάσεων είναι μικρός. Έχει τριαντατρείς κόμβους και σαραντατέσσερα τόξα. Το αρχικό και το τελικό τμήμα του χώρου καταστάσεων φαίνονται στα Σχήματα 20 και 21, αντίστοιχα.



Σχήμα 20: Το αρχικό τμήμα του χώρου καταστάσεων του Χρωματισμένου δικτύου Petri του Σχήματος 18



Σχήμα 21: Το τελικό τμήμα του χώρου καταστάσεων του Χρωματισμένου δικτύου Petri του Σχήματος 18

Τα στρογγυλεμένα ορθογώνια είναι οι κόμβοι του χώρου καταστάσεων. Κάθε ένας κόμβος παριστάνει ένα προσιτό μαρκάρισμα και το περιεχόμενο κάθε μαρκαρίσματος περιγράφεται στο διακεκομμένο ορθογώνιο δίπλα από τον κόμβο. Οι θέσεις με κενό μαρκάρισμα παραλείπονται, όπως και τα μαρκαρίσματα των θέσεων Send και Limit, γιατί το μαρκάρισμα της θέσης Send δεν αλλάζει ποτέ, ενώ η θέση Limit προστέθηκε, μόνο για να περιορίσει το μέγεθος του χώρου καταστάσεων. Στο πάνω μέρος του

Σχήματος 20, υπάρχει ένας κόμβος με πιο παχύ περίγραμμα. Ο κόμβος αυτός παριστάνει το αρχικό μαρκάρισμα και το κείμενο μέσα στον κόμβο προσδιορίζει ότι αυτός είναι ο κόμβος νούμερο ένα και ότι έχει ένα πρόγονο και ένα απόγονο. Όμοια, ο κόμβος νούμερο δύο έχει ένα πρόγονο και δύο απογόνους. Ο αριθμός των απογόνων μπορεί να μας φανεί χρήσιμος, όταν έχει σχεδιαστεί μόνο ένα τμήμα του χώρου καταστάσεων.

Κάθε τόξο παριστάνει την εκτέλεση του στοιχείου καταχώρησης που αναγράφεται στο διακεκομμένο ορθογώνιο που βρίσκεται στην κορυφή του τόξου. Με M_n συμβολίζουμε το μαρκάρισμα του κόμβου νούμερο n . Στο μαρκάρισμα M_1 το μοναδικό ενεργοποιημένο στοιχείο καταχώρησης είναι η μετάβαση SendPacket με καταχώρηση $< n = 1, p = \text{"Modellin"} >$. Μετά την εκτέλεση αυτού του στοιχείου καταχώρησης, προκύπτει το μαρκάρισμα M_2 , στο οποίο υπάρχουν δύο ενεργοποιημένα στοιχεία καταχώρησης. Η εκτέλεση της μετάβασης TransmitPacket με τη μεταβλητή Ok να έχει τιμή true, θα οδηγήσει σε ένα νέο μαρκάρισμα M_3 , ενώ η εκτέλεση της μετάβασης TransmitPacket με τη μεταβλητή Ok να έχει την τιμή false, θα οδηγήσει πίσω στο αρχικό μαρκάρισμα M_1 . Δεν συμπεριλαμβάνουμε τόξα που αντιστοιχούν σε βήματα με περισσότερα από ένα στοιχεία καταχώρησης, γιατί τέτοια τόξα θα μας έδιναν πληροφορίες για τον ταυτοχρονισμό μεταξύ των στοιχείων καταχώρησης, αλλά δεν είναι απαραίτητα για την επαλήθευση των ιδιοτήτων συμπεριφοράς.

Από τα Σχήματα 20 και 21, φαίνεται ότι ο χώρος καταστάσεων έχει μια επαναληπτική δομή, δηλαδή κάποια τμήματα του επαναλαμβάνονται. Ο υπογράφος των κόμβων {4, 5, 7, 9} έχει την ίδια μορφή με τον υπογράφο των κόμβων {28, 29, 31, 33}. Η μόνη διαφορά είναι ότι ο δεύτερος υπογράφος είναι “τρία πακέτα μπροστά” από τον πρώτο. Αν σχεδιάσουμε και το μεσαίο τμήμα του χώρου καταστάσεων, θα βρούμε ακόμα δύο υπογράφους με την ίδια μορφή.

Ας αναλύσουμε τώρα την πιο σύνθετη περίπτωση, όπου ένα πακέτο μπορεί να προσπεράσει κάποιο άλλο. Για να το κάνουμε αυτό, κατασκευάζουμε ένα χώρο καταστάσεων για την περίπτωση που το αρχικό μαρκάρισμα της θέσης Limit είναι 2'e. Ο νέος χώρος καταστάσεων είναι αρκετά μεγαλύτερος από τον πρώτο κι έτσι δεν γίνεται καμία προσπάθεια να τον σχεδιάσουμε και χρησιμοποιούμε μια **αναφορά του χώρου καταστάσεων (state space report)**, η οποία παρέχει κρίσιμες πληροφορίες σχετικά με τη συμπεριφορά του Χρωματισμένου δικτύου Petri. Η αναφορά του χώρου καταστάσεων αποτελείται από τέσσερα μέρη.

Το πρώτο μέρος φαίνεται στο Σχήμα 22. Περιέχει **στατιστικές (statistical)** πληροφορίες σχετικά με το μέγεθος του χώρου καταστάσεων. Βλέπουμε ότι ο συγκεκριμένος χώρος καταστάσεων έχει 428 κόμβους και 1130 τόξα και ότι υπολογίστηκε ο πλήρης χώρος καταστάσεων (Status: Full) μέσα σε ένα δευτερόλεπτο (Secs: 1). Το στατιστικό μέρος περιέχει επίσης πληροφορίες σχετικά με το **SCC-γράφο (SCC-graph)**, δηλαδή τον αριθμό των **ισχυρά συνδεδεμένων τμημάτων (strongly connected components)** που περιέχει ο χώρος καταστάσεων, καθώς και τον αριθμό των τόξων που ξεκινούν από ένα ισχυρά συνδεδεμένο τμήμα και καταλήγουν σε ένα άλλο. Ένα ισχυρά συνδεδεμένο τμήμα είναι ένας μέγιστος υπογράφος, στον οποίο είναι δυνατό να βρεθεί ένα μονοπάτι από ένα κόμβο σε έναν οποιονδήποτε άλλο κόμβο. Τα ισχυρά συνδεδεμένα τμήματα είναι πολύ χρήσιμα για τον προσδιορισμό συγκεκριμένων ιδιοτήτων συμπεριφοράς και μπορούν να

υπολογιστούν από τυπικούς αλγορίθμους που είναι γραμμικοί στο χρόνο και στο χώρο. Στο Σχήμα 22 βλέπουμε ότι υπάρχουν 182 ισχυρά συνδεδεμένα τμήματα και 673 τόξα που ξεκινούν από ένα ισχυρά συνδεδεμένο τμήμα και καταλήγουν σε ένα άλλο ισχυρά συνδεδεμένο τμήμα. Άρα τα ισχυρά συνδεδεμένα τμήματα είναι λιγότερα από τους κόμβους. Αυτό σημαίνει ότι το σύστημα έχει τουλάχιστον ένα ισχυρά συνδεδεμένο τμήμα που περιέχει περισσότερους από έναν, κόμβους και άρα υπάρχουν στο σύστημα μη-πεπερασμένες ακολουθίες εκτέλεσης. Με άλλα λόγια, δεν μπορούμε να είμαστε σίγουροι ότι το πρωτόκολλο τερματίζει. Για να πετύχουμε τερματισμό του πρωτοκόλλου, συνήθως περιορίζουμε τον αριθμό των επαναμεταδόσεων.

Statistics	
State Space	
Nodes:	428
Arcs:	1130
Secs:	1
Status:	Full
SCC Graph	
Nodes:	182
Arcs:	673
Secs:	1

Σχήμα 22: Το μέγεθος του χώρου καταστάσεων και ο SCC-γράφος

Στο Σχήμα 23 φαίνεται το δεύτερο μέρος της αναφοράς του χώρου καταστάσεων, το οποίο περιέχει πληροφορίες για τα **όρια ακεραίων (integer bounds)** και τα **όρια πολλαπλών συνόλων (multi-set bounds)**. Στο πάνω μέρος του Σχήματος 23 φαίνονται τα άνω και τα κάτω όρια ακεραίων, δηλαδή ο μέγιστος και ο ελάχιστος αριθμός των μαρκών που μπορεί να έχει κάθε θέση. Όπως φαίνεται στο Σχήμα 23, κάθε μία από τις θέσεις A, B, C, D και Limit έχουν πάντα από μηδέν έως δύο μάρκες, ενώ οι θέσεις NextSend, NextRec και Received έχουν πάντα μία μάρκα η καθεμιά. Τέλος, η θέση Send έχει πάντα τέσσερις μάρκες. Καμία από αυτές τις πληροφορίες δεν προκαλεί έκπληξη, αλλά επιβεβαιώνουν ότι το σύστημα λειτουργεί σωστά.

Στα κατώτερα μέρη του Σχήματος 23 φαίνονται τα όρια πολλαπλών συνόλων. Εξ ορισμού, το άνω όριο των πολλαπλών συνόλων μιας θέσης, είναι το μικρότερο πολλαπλό σύνολο που είναι μεγαλύτερο από όλα τα προσιτά μαρκαρίσματα αυτής της θέσης. Αντίστοιχα, το κάτω όριο των πολλαπλών συνόλων μιας θέσης, είναι το μεγαλύτερο πολλαπλό σύνολο που είναι μικρότερο από όλα τα προσιτά μαρκαρίσματα αυτής της θέσης. Τα όρια ακεραίων μας δίνουν πληροφορίες για τον αριθμό των μαρκών, ενώ τα όρια πολλαπλών συνόλων μας δίνουν πληροφορίες για τις τιμές που μπορούν να έχουν οι μάρκες. Από τα όρια πολλαπλών συνόλων του Σχήματος 23, φαίνεται ότι οι θέσεις A και B μπορούν να περιέχουν και τα τέσσερα διαφορετικά πακέτα, ενώ οι θέσεις C και D μπορούν να περιέχουν και τις τέσσερις πιθανές βεβαιώσεις λήψης πακέτων. Μια βεβαίωση λήψης πακέτου καθορίζει πάντα τον αριθμό του επόμενου πακέτου για αποστολή κι έτσι δεν υπάρχει ποτέ βεβαίωση

λήψης πακέτου με τιμή 1. Για αυτές τις τέσσερις θέσεις δεν είναι δυνατές άλλες τιμές μαρκών.

Integer Bounds		
	Upper	Lower
A	2	0
B	2	0
C	2	0
D	2	0
Limit	2	0
NextRec	1	1
NextSend	1	1
Received	1	1
Send	4	4
Upper Multi-set Bounds		
A	$2^{(1, "Modellin")} + 2^{(2, "g \text{ and } An")} + 2^{(3, "alysis")} + 2^{(4, "###")}$	
B	$2^{(1, "Modellin")} + 2^{(2, "g \text{ and } An")} + 2^{(3, "alysis")} + 2^{(4, "###")}$	
C	$2^2 + 2^3 + 2^4 + 2^5$	
D	$2^2 + 2^3 + 2^4 + 2^5$	
Limit	2^e	
NextRec	$1^1 + 1^2 + 1^3 + 1^4 + 1^5$	
NextSend	$1^1 + 1^2 + 1^3 + 1^4 + 1^5$	
Received	$1^{""} + 1^{\text{"Modellin"}} + 1^{\text{"Modelling and An"}} + 1^{\text{"Modelling and Analysis"}}$	
Send	$1^{(1, "Modellin")} + 1^{(2, "g \text{ and } An")} + 1^{(3, "alysis")} + 1^{(4, "###")}$	
Lower Multi-set Bounds		
A	empty	
B	empty	
C	empty	
D	empty	
Limit	empty	
NextRec	empty	
NextSend	empty	
Received	empty	
Send	$1^{(1, "Modellin")} + 1^{(2, "g \text{ and } An")} + 1^{(3, "alysis")} + 1^{(4, "###")}$	

Σχήμα 23: Τα όρια ακεραίων και τα όρια πολλαπλών συνόλων

Το άνω όριο πολλαπλών συνόλων για τη θέση A είναι ένα πολλαπλό σύνολο με οχτώ στοιχεία, παρ'όλο που σύμφωνα με το άνω όριο ακεραίων, ποτέ δεν μπορούν να υπάρχουν στη θέση A πάνω από δύο μάρκες ταυτόχρονα. Οι δύο αυτές μάρκες μπορούν να πάρουν τις εξής τέσσερις τιμές μαρκών : (1, "Modellin"), (2, "g and An"), (3, "alysis") και (4, "###"). Άρα μπορούμε να έχουμε, για παράδειγμα, τα μαρκαρίσματα :

1` (1, "Modellin")
 1` (1, "Modellin") + 1` (3, "alysis")
 2` (3, "alysis").

Από τα παραπάνω βγαίνει εύκολα το συμπέρασμα ότι το μικρότερο πολλαπλό σύνολο που είναι μεγαλύτερο από όλα τα πιθανά μαρκαρίσματα της θέσης A είναι το πολλαπλό σύνολο :

2` (1, "Modellin") + 2` (2, "g and An") + 2` (3, "alysis") + 2` (4, "###").

Με παρόμοιο τρόπο προκύπτουν και τα όρια πολλαπλών συνόλων των θέσεων B, C, D, NextRec, NextSend και Received.

Οι δύο θέσεις μετρητών NextSend και NextRec μπορούν να πάρουν όλες τις τιμές από 1 έως 5. Η θέση Received μπορεί να περιέχει τέσσερις διαφορετικές τιμές, που αντιστοιχούν στις περιπτώσεις που έχουν παραληφθεί τα δεδομένα από μηδέν, ένα, δύο ή τρία πακέτα (το πακέτο νούμερο τέσσερα περιέχει τη συμβολοσειρά "###", η οποία δεν αντιγράφεται ποτέ στη θέση Received). Τέλος, η θέση Send έχει ίδια άνω και κάτω όρια πολλαπλών συνόλων. Αυτό σημαίνει ότι το μαρκαρίσμα αυτής της θέσης δεν αλλάζει ποτέ. Τα όρια πολλαπλών συνόλων είναι τα αναμενόμενα και αυτό υποδεικνύει ότι το πρωτόκολλο λειτουργεί σωστά.

Το τρίτο μέρος της αναφοράς χώρου καταστάσεων φαίνεται στο Σχήμα 24 και περιέχει πληροφορίες για τις ιδιότητες της **οικείας κατάστασης (home state)** και της **διάρκειας (liveness)**.

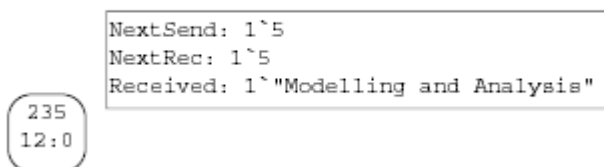
Home Properties
Home Markings: [235]
Liveness Properties
Dead Markings: [235]
Dead Transitions Instances: None
Live Transitions Instances: None

Σχήμα 24: Οι ιδιότητες της οικείας κατάστασης και της διάρκειας

Όπως αναφέρθηκε στο Κεφάλαιο 2, ένα **οικείο μαρκαρίσμα (home marking)** είναι ένα μαρκαρίσμα που είναι προσιτό από όλα τα προσιτά μαρκαρίσματα, δηλαδή μπορούμε να φτάσουμε πάντα σε αυτό το μαρκαρίσμα, ανεξάρτητα με το τι έχει

προηγηθεί. Όπως φαίνεται και από το Σχήμα 24, το πρωτόκολλο επικοινωνίας έχει μόνο ένα οικείο μαρκάρισμα, το M235. Ένα **νεκρό μαρκάρισμα (dead marking)** είναι ένα μαρκάρισμα που δεν έχει ενεργοποιημένες μεταβάσεις. Το πρωτόκολλο έχει μόνο ένα νεκρό μαρκάρισμα και είναι ίδιο με το οικείο μαρκάρισμα, δηλαδή το M235. Το μαρκάρισμα του κόμβου 235 φαίνεται στο Σχήμα 25.

Dead Marking / Home Marking



Σχήμα 25: Το μαρκάρισμα του κόμβου 235

Όπως φαίνεται από το Σχήμα 25, το μαρκάρισμα M235 αντιστοιχεί στην κατάσταση, όπου και τα τέσσερα πακέτα έχουν παραληφθεί επιτυχώς (και δεν υπάρχει καμία μάρκα στις θέσεις A, B, C και D). Το γεγονός ότι το μαρκάρισμα M235 είναι νεκρό, υποδεικνύει ότι το πρωτόκολλο είναι **μερικώς σωστό (partially correct)**, δηλαδή ότι αν τερματίσει, θα έχουμε το σωστό αποτέλεσμα. Από την άλλη μεριά, το γεγονός ότι το μαρκάρισμα M235 είναι οικείο, υποδεικνύει ότι το πρωτόκολλο έχει την ιδιότητα, ότι δεν μπορεί ποτέ να φτάσει σε μια κατάσταση από την οποία είναι αδύνατο να τερματίσει με το σωστό αποτέλεσμα.

Από τις ιδιότητες της διάρκειας στο Σχήμα 24, φαίνεται ότι δεν υπάρχουν **νεκρές μεταβάσεις (dead transitions)**. Αυτό σημαίνει ότι κάθε μετάβαση είναι ενεργοποιημένη σε τουλάχιστον ένα προσιτό μαρκάρισμα. Φαίνεται επίσης ότι δεν υπάρχουν **διαρκείς μεταβάσεις (live transitions)**. Μια διαρκής μετάβαση είναι μια μετάβαση η οποία πάντα, ότι κι αν γίνει, μπορεί να ενεργοποιηθεί άλλη μια φορά. Όταν υπάρχουν νεκρά μαρκάρια, όπως στο πρωτόκολλο επικοινωνίας, δεν μπορούν να υπάρχουν διαρκείς μεταβάσεις. Με μια πρώτη ματιά θα μπορούσε κανείς να σκεφτεί ότι η ύπαρξη νεκρών μαρκαρισμάτων θα απέτρεπε την ύπαρξη οικείων μαρκαρισμάτων. Όμως, εξ ορισμού, ένα μαρκάρισμα είναι πάντα προσιτό από τον εαυτό του, κι έτσι ένα νεκρό μαρκάρισμα μπορεί να είναι ένα οικείο μαρκάρισμα, αν είναι το μοναδικό νεκρό μαρκάρισμα. Αυτή η περίπτωση ισχύει και για το πρωτόκολλο επικοινωνίας.

Το τέταρτο και τελευταίο μέρος της αναφοράς του χώρου καταστάσεων φαίνεται στο Σχήμα 26. Παρέχει πληροφορίες σχετικά με την ιδιότητα της δικαιοσύνης, δηλαδή σχετικά με το πόσο συχνά εκτελείται κάθε μετάβαση. Από το Σχήμα 26 φαίνεται ότι οι μεταβάσεις SendPacket και TransmitPacket είναι **αμερόληπτες (impartial)**, δηλαδή κάθε μία από αυτές εκτελείται απείρως συχνά, σε οποιαδήποτε μη-πεπερασμένη ακολουθία εκτέλεσης. Με άλλα λόγια, εάν μία από αυτές τις μεταβάσεις πάψει να εκτελείται, τότε το πρωτόκολλο, μετά από κάποια επιπρόσθετα βήματα, πρέπει να τερματίσει. Υπάρχουν δύο άλλοι τύποι δικαιοσύνης. Οι μεταβάσεις που ανάκουν στον ένα τύπο ονομάζονται **δίκαιες (fair)**, ενώ οι μεταβάσεις που ανήκουν στο άλλο τύπο ονομάζονται **ακριβοδίκαιες (just)**. Οι τύποι αυτοί είναι πιο αδύναμοι από τον τύπο της αμεροληψίας, κι έτσι οι μεταβάσεις

SendPacket και TransmitPacket αυτόματα τους ικανοποιούν. Στο Σχήμα 26 φαίνεται ότι καμία από τις υπόλοιπες τρεις μεταβάσεις δεν είναι αμερόληπτη, δίκαιη ή ακριβοδίκαιη.

Fairness Properties	
SendPack	Impartial
TranPack	Impartial
RecPack	No Fairness
TranAck	No Fairness
RecAck	No Fairness

Σχήμα 26: Η ιδιότητα της διάρκειας

Η αναφορά του χώρου καταστάσεων δημιουργείται σε λίγα δευτερόλεπτα, εντελώς αυτόματα και περιέχει πολλές εξαιρετικά χρήσιμες πληροφορίες σχετικά με τη δυναμική συμπεριφορά του Χρωματισμένου δικτύου Petri. Μελετώντας την αναφορά του χώρου καταστάσεων, ο σχεδιαστής παίρνει μια πρώτη ιδέα, αν το μοντέλο λειτουργεί όπως ήταν αναμενόμενο. Αν το σύστημα περιέχει σφάλματα, είναι συνήθως εμφανές στην αναφορά του χώρου καταστάσεων. Ο σχεδιαστής μπορεί επίσης να υποβάλλει τα δικά του ερωτήματα (queries), όπως αυτό που φαίνεται στο Σχήμα 27. Η ερώτηση περιέχεται στο ορθογώνιο κουτί, ενώ η απάντηση στο στρογγυλεμένο κουτί. Το συγκεκριμένο ερώτημα ζητάει το μήκος του συντομότερου μονοπατιού, δηλαδή ενός μονοπατιού που δεν έχει αναμεταδόσεις και προσπέρασμα πακέτων, από το μαρκάρισμα M1, στο μαρκάρισμα M235. Η απάντηση είναι ότι πρέπει να εκτελεστούν τουλάχιστον είκοσι μεταβάσεις για να φτάσουμε από το M1 στο M235. Η H ArcsInPath είναι μια συνάρτηση βιβλιοθήκης.

Length of Shortest Path

`length(ArcsInPath(1,235));` > 20 : int

Σχήμα 27: Το μήκος του συντομότερου μονοπατιού από το M1 στο M235

Ένα από τα βασικά μειονεκτήματα της μεθόδου του χώρου καταστάσεων είναι το λεγόμενο πρόβλημα της έκρηξης καταστάσεων. Σε πολλά συστήματα, ο χώρος καταστάσεων γίνεται τόσο μεγάλος που δεν μπορεί να κατασκευαστεί ολόκληρος. Ένα άλλο μειονέκτημα αυτής της μεθόδου είναι το γεγονός ότι ο χώρος καταστάσεων κατασκευάζεται πάντα, μόνο για ένα συγκεκριμένο αρχικό μαρκάρισμα, το οποίο συχνά αντιστοιχεί μόνο σε μία από τις πολλές πιθανές αρχικές καταστάσεις του συστήματος.

4.4 Χρωματισμένα Δίκτυα Petri με Χρόνο

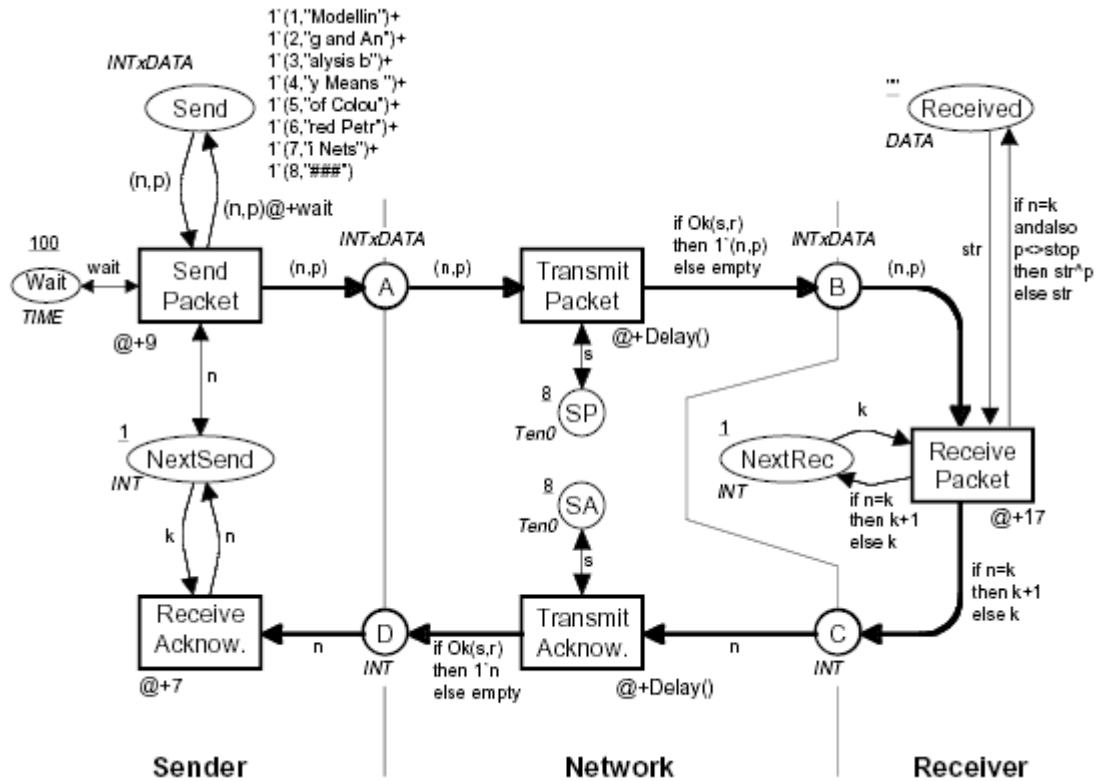
Για να εξετάσουμε την απόδοση των συστημάτων, δηλαδή την ταχύτητα με την οποία λειτουργούν, επεκτείνουμε τα Χρωματισμένα δίκτυα Petri, εισάγοντας την έννοια του χρόνου, σε **Χρωματισμένα δίκτυα Petri με Χρόνο (Timed Coloured Petri nets – TCPNs)**. Για να το πετύχουμε αυτό, εισάγουμε ένα **καθολικό ρολόι (global clock)**, του οποίου οι τιμές παριστάνουν το **χρόνο μοντελοποίησης (model time)** και είναι είτε ακέραιοι αριθμοί, δηλαδή διακριτές τιμές, είτε πραγματικοί αριθμοί, δηλαδή συνεχείς τιμές. Κάθε μάρκα, εκτός από την τιμή της, φέρει μια **τιμή χρόνου (time value)**, η οποία ονομάζεται και **χρονική σφραγίδα (time stamp)**. Η χρονική σφραγίδα υποδεικνύει το μικρότερο χρόνο μοντελοποίησης, στον οποίο μπορεί να χρησιμοποιηθεί η μάρκα, δηλαδή να απομακρυνθεί από ένα στοιχείο καταχώρησης.

Σε ένα Χρωματισμένο δίκτυο Petri με Χρόνο, ένα στοιχείο καταχώρησης λέγεται ότι είναι **χρωματικά ενεργοποιημένο (colour enabled)**, όταν πληρεί τις προϋποθέσεις του **κανόνα ενεργοποίησης (enabling rule)** που ισχύει στα Χρωματισμένα δίκτυα Petri χωρίς χρόνο, όταν δηλαδή υπάρχουν οι απαιτούμενες μάρκες στις θέσεις εισόδου και ο φύλακας είναι αληθής. Για να είναι όμως ενεργοποιημένο ένα στοιχείο καταχώρησης, πρέπει να είναι και **έτοιμο (ready)**. Αυτό σημαίνει ότι οι χρονικές σφραγίδες των μαρκών που είναι να απομακρυνθούν, πρέπει να είναι μικρότερες ή ίσες από τον τρέχων χρόνο μοντελοποίησης.

Για να μοντελοποιηθεί μια δράση που χρειάζεται r μονάδες χρόνου για να ολοκληρωθεί, η αντίστοιχη μετάβαση t δημιουργεί για τις μάρκες εξόδου χρονικές σφραγίδες, που είναι κατά r μονάδες χρόνου μεγαλύτερες από την τιμή του ρολογιού, στην οποία εκτελείται η μετάβαση t . Άρα, οι μάρκες που παράγονται από τη μετάβαση t , δεν είναι διαθέσιμες για r μονάδες χρόνου. Θα ήταν ίσως πιο λογικό, αν καθυστερούσαμε τη δημιουργία των μαρκών, έτσι ώστε να μην υπάρχουν μάρκες, πριν περάσουν r μονάδες χρόνου από τότε που ξεκίνησε η εκτέλεση της μετάβασης t . Μια τέτοια προσέγγιση όμως, θα είχε ως αποτέλεσμα, ένα Χρωματισμένο δίκτυο Petri με Χρόνο να παρουσιάζει “ενδιάμεσα” μαρκαρίσματα που δεν αντιστοιχούν σε μαρκαρίσματα στο αντίστοιχο Χρωματισμένο δίκτυο Petri χωρίς χρόνο, γιατί θα υπήρχαν μαρκαρίσματα, στα οποία οι μάρκες εισόδου έχουν απομακρυνθεί, ενώ οι μάρκες εξόδου δεν έχουν ακόμα παραχθεί. Έτσι, θα υπήρχε μια πιο περίπλοκη σχέση μεταξύ της συμπεριφοράς των Χρωματισμένων δικτύων Petri με Χρόνο και χωρίς χρόνο.

Η εκτέλεση ενός Χρωματισμένου δικτύου Petri με Χρόνο **καθορίζεται με βάση το χρόνο (time driven)** και είναι παρόμοια με την εκτέλεση των **ουρών γεγονότων (event queues)**, οι οποίες συναντώνται σε πολλές άλλες γλώσσες που προσομοιώνουν συστήματα διακριτών γεγονότων. Το σύστημα παραμένει σε ένα δεδομένο χρόνο μοντελοποίησης, όσο υπάρχουν χρωματικά ενεργοποιημένα στοιχεία καταχώρησης, έτοιμα για εκτέλεση. Όταν δεν υπάρχουν άλλα στοιχεία καταχώρησης που μπορούν να εκτελεστούν στον τρέχων χρόνο μοντελοποίησης, το σύστημα αυξάνει την τιμή του ρολογιού στον επόμενο χρόνο μοντελοποίησης που μπορούν να εκτελεστούν στοιχεία καταχώρησης. Κάθε μαρκαρίσμα υπάρχει για ένα πεπερασμένο διάστημα του χρόνου μοντελοποίησης, ενώ η εκτέλεση ενός στοιχείου καταχώρησης είναι στιγμιαία.

Ένα Χρωματισμένο δίκτυο Petri με Χρόνο για το πρωτόκολλο επικοινωνίας φαίνεται στο Σχήμα 28. Το χροнисμένο δίκτυο έχει την ίδια δομή δικτύου με το μη χροнисμένο, με τη μόνη διαφορά ότι έχει προστεθεί μια νέα θέση Wait (στο αριστερό μέρος του Σχήματος). Η θέση αυτή προσδιορίζει, πόσο χρόνο θα πρέπει να περιμένει η μετάβαση SendPacket πριν αναμεταδώσει ένα πακέτο.



Σχήμα 28: Το Χρωματισμένο δίκτυο Petri με Χρόνο που μοντελοποιεί το απλό πρωτόκολλο επικοινωνίας

Από τις δηλώσεις στο Σχήμα 29, φαίνεται ότι ο τύπος INT είναι χροнисμένος. Αυτό σημαίνει, ότι οι μάρκες αυτού του τύπου φέρουν χρονικές σφραγίδες. Οι τύποι DATA, Ten0, Ten1 δεν είναι χροнисμένοι. Οι μάρκες που ανήκουν σε έναν από αυτούς τους τύπους δεν φέρουν χρονικές σφραγίδες και άρα είναι πάντα διαθέσιμες. Ο δομημένος τύπος INTxDATA είναι χροнисμένος, γιατί περιέχει τον τύπο INT που είναι χροнисμένος. Άρα οι τέσσερις θέσεις Received, SP, SA και Wait έχουν μάρκες χωρίς χρονικές σφραγίδες, ενώ οι υπόλοιπες επτά θέσεις έχουν μάρκες με χρονικές σφραγίδες. Στο σύστημα αυτό, χρησιμοποιείται ένα ρολόι που παίρνει ακέραιες τιμές. Επειδή το ρολόι αρχίζει τη μέτρηση από την τιμή μηδέν, στο αρχικό μαρκάρισμα όλες οι χρονικές σφραγίδες είναι ίσες με το μηδέν.

Σε κάθε μία από τις πέντε μεταβάσεις έχει προστεθεί μια χρονική επιγραφή (time inscription), η οποία αρχίζει με το συμβολισμό @ +. Η χρονική επιγραφή υποδεικνύει πόσο χρόνο χρειάζεται η αντίστοιχη δράση για να ολοκληρωθεί. Στη συνέχεια θα αναλυθεί κάθε μετάβαση του πρωτοκόλλου επικοινωνίας ξεχωριστά.

```

color INT = int timed;
color DATA = string;
color INTxDATA = product INT * DATA;
var n, k : INT;
var p, str : DATA;
val stop = "###";

color Ten0 = int with 0..10;
color Ten1 = int with 1..10;
var s : Ten0; var r : Ten1;
fun Ok(s:Ten0,r:Ten1) = (r<=s);

color NetDelay = int with 25..75 declare ran;
fun Delay() = ran!NetDelay();
var wait : TIME;

```

Σχήμα 29: Οι δηλώσεις του Χρωματισμένου δικτύου Petri με Χρόνο του Σχήματος 20

Η μετάβαση SendPacket έχει χρονική επιγραφή : @ + 9. Αυτό σημαίνει ότι οι μάρκες που δημιουργούνται στις θέσεις A και NextSend, παίρνουν χρονικές σφραγίδες που είναι κατά 9 μονάδες χρόνου μεγαλύτερες από το χρόνο r^* , στον οποίο εκτελείται η μετάβαση. Το τόξο εξόδου που καταλήγει στη θέση Send, καθορίζει μια επιπρόσθετη χρονική καθυστέρηση για τις μάρκες που προστίθενται σε αυτή τη θέση. Μία τέτοια μάρκα θα πάρει μια χρονική σφραγίδα $r^* + 9 + 100$, όπου ο αριθμός 100 είναι η τιμή της μεταβλητής wait. Ο αριθμός 9 αντιπροσωπεύει το χρόνο που απαιτείται για την αποστολή ενός πακέτου, ενώ ο αριθμός 100 το χρόνο που πρέπει να περάσει, προτού γίνει μια αναμετάδοση, δηλαδή προτού η θέση SendPacket εκτελεστεί ακόμα μία φορά για το ίδιο πακέτο. Άρα, αναμετάδοση θα γίνει μόνο εάν η τιμή της μεταβλητής n στη θέση NextSend παραμείνει αμετάβλητη για 9 + 100 μονάδες χρόνου, δηλαδή μόνο αν δεν ληφθεί βεβαίωση λήψης πακέτου μέσα σε αυτή τη χρονική περίοδο.

Η μετάβαση TransmitPacket έχει χρονική επιγραφή : @ + Delay(), όπου Delay είναι μια συνάρτηση που επιστρέφει ένα τυχαίο στοιχείο του τύπου NetDelay, δηλαδή έναν τυχαίο ακέραιο στο διάστημα [25, 75]. Αυτό σημαίνει ότι η διάρκεια μιας μετάδοσης μπορεί να ποικίλλει, αλλά πάντα μέσα στο διάστημα [25, 75].

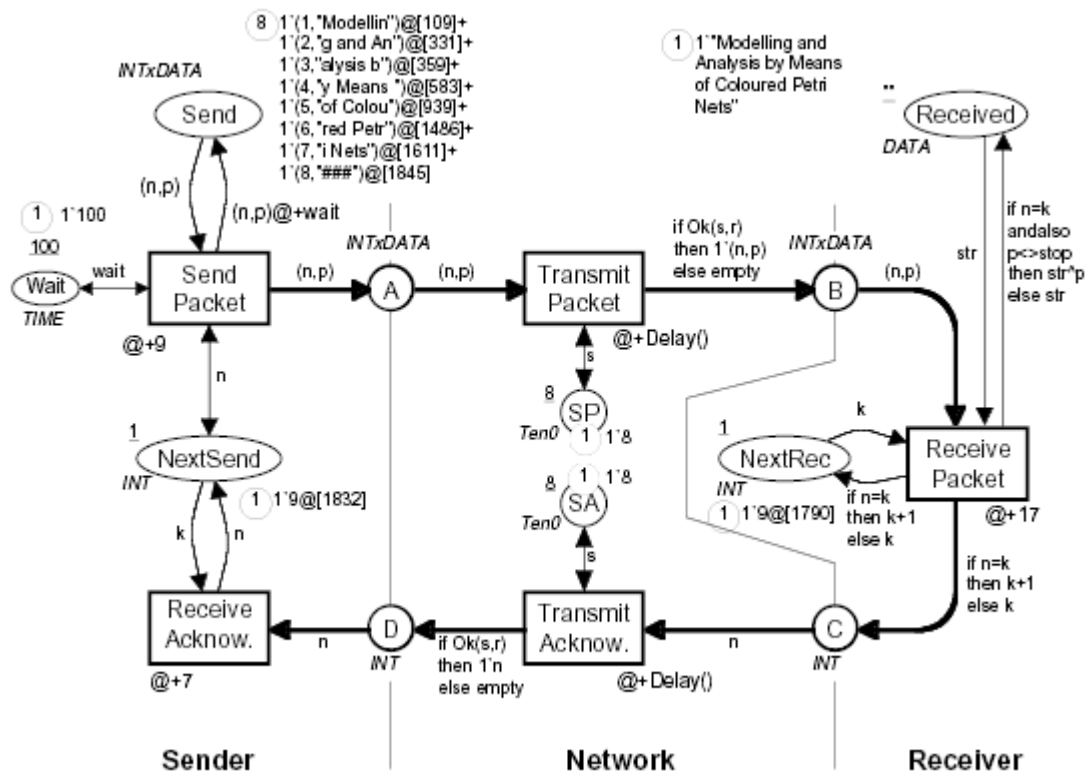
Οι μεταβάσεις ReceivePacket και ReceiveAcknowledgement έχουν χρονικές επιγραφές που ορίζουν μια σταθερή χρονική διάρκεια, 17 και 7 μονάδες χρόνου αντίστοιχα, ενώ η μετάβαση TransmitAcknowledgement έχει μια μεταβλητή χρονική διάρκεια, μεταξύ 25 και 75. Είναι εύκολο να χρησιμοποιηθούν στατιστικές συναρτήσεις, οι οποίες ορίζουν περισσότερο σύνθετους τύπους καθυστερήσεων, όπως για παράδειγμα εκθετικά κατανομημένες καθυστερήσεις.

Στη θέση NextSend η μάρκα φέρει μια χρονική σφραγίδα. Αυτό συνεπάγεται ότι ο αποστολέας Sender δεν μπορεί να ξεκινήσει μια νέα αποστολή πακέτου (SendPacket) ή μια νέα λήψη βεβαίωσης πακέτου (ReceiveAcknowledgement), όσο μία από αυτές τις δράσεις βρίσκεται ήδη σε εξέλιξη. Το ίδιο ισχύει και για τη θέση NextRec του παραλήπτη Receiver.

Σε ένα Χρωματισμένο δίκτυο Petri με Χρόνο, κάθε βήμα πρέπει να αποτελείται από στοιχεία καταχώρησης που είναι, και χρωματικά ενεργοποιημένα, και έτοιμα. Έτσι οι πιθανές ακολουθίες εκτέλεσης ενός Χρωματισμένου δικτύου Petri με Χρόνο αποτελούν πάντα ένα υποσύνολο των πιθανών ακολουθιών εκτέλεσης του αντίστοιχου Χρωματισμένου δικτύου Petri χωρίς χρόνο. Άρα, η σχέση μεταξύ της

συμπεριφοράς του Χρωματισμένου δικτύου Petri με Χρόνο και του αντίστοιχου Χρωματισμένου δικτύου Petri χωρίς χρόνο είναι καλά ορισμένη και γίνεται εύκολα κατανοητή.

Μετά από μερικά βήματα προσομοίωσης το Χρωματισμένο δίκτυο Petri με Χρόνο του Σχήματος 28 μπορεί να φτάσει σε ένα νεκρό τελικό μαρκάρισμα, που φαίνεται στο Σχήμα 30. Στα μαρκάρια των θέσεων χωρίζουμε τις χρονικές σφραγίδες από τις τιμές των μαρκών με ένα σύμβολο @. Από τη χρονική σφραγίδα της θέσης NextRec, φαίνεται ότι το τελευταίο πακέτο παραλήφθηκε σε χρόνο 1790. Ομοίως, η χρονική σφραγίδα στη θέση NextSend υποδεικνύει ότι η τελευταία δράση του αποστολέα Sender τελείωσε σε χρόνο 1832. Οι χρονικές σφραγίδες στη θέση Send μας ενημερώνουν για το χρόνο στον οποίο κάθε πακέτο θα αναμεταδιδόταν, αν αυτό ήταν απαραίτητο. Για παράδειγμα, το πρώτο πακέτο θα αναμεταδιδόταν σε χρόνο 109, το δεύτερο σε χρόνο 331, το τρίτο σε χρόνο 359 και ούτω καθεξής.



Σχήμα 30: Το τελικό μαρκάρισμα του Χρωματισμένου δικτύου Petri με Χρόνο του Σχήματος 28

Με τη βοήθεια του Χρωματισμένου δικτύου Petri με Χρόνο του Σχήματος 28, μπορούμε να εξετάσουμε την απόδοση του πρωτοκόλλου, παραδείγματος χάριν, να δοκιμάσουμε διαφορετικές τιμές για την καθυστέρηση αναμετάδοσης που καθορίζεται από τη θέση Wait. Μια μικρή καθυστέρηση αυξάνει την πιθανότητα να γίνουν αναμεταδόσεις που δεν χρειάζονται. Αυξάνει επίσης την πιθανότητα να προσπεράσει ένα πακέτο, κάποιο άλλο, καθώς και την πιθανότητα να αναβληθεί μια λειτουργία λήψης βεβαίωσης πακέτου (ReceiveAcknowledgement), επειδή ο

αποστολέας Sender είναι απασχολημένος με μια αναμετάδοση. Μια μεγάλη καθυστέρηση σημαίνει ότι μπορεί να περάσει ένα αρκετά μεγάλο χρονικό διάστημα, προτού ο αποστολέας Sender αναγνωρίσει ότι ένα πακέτο ή μια βεβαίωση λήψης πακέτου έχει χαθεί. Εκτελώντας αρκετές προσομοιώσεις, με διαφορετικές τιμές για τη μάρκα στη θέση Wait, μπορούμε να βρούμε την ιδανική τιμή για την καθυστέρηση αναμετάδοσης.

Για να πάρουμε αξιόπιστα αποτελέσματα από την προσομοίωση, αυξάνουμε τον αριθμό των πακέτων από οχτώ σε εκατό. Επίσης επιθυμούμε να περιγράψουμε περισσότερο ρεαλιστικούς ρυθμούς επιτυχίας, κι έτσι αλλάζουμε τους τύπους Ten0 και Ten1, σε Hun0 και Hun1 που περιέχουν όλους τους ακέραιους αριθμούς από 0 ως 100 και από 1 ως 100, αντίστοιχα. Με αυτή την αλλαγή, έχουμε εκατό διαφορετικά στοιχεία καταχώρησης για τη μετάβαση TransmitPacket – για κάθε μάρκα στη θέση A με έτοιμη χρονική σφραγίδα. Αντί να υπολογίσουμε όλες αυτές τις καταχωρήσεις κι έπειτα να επιλέξουμε μία από αυτές, χρησιμοποιούμε τη συνάρτηση ran'Hun1(), η οποία επιλέγει τυχαία ένα στοιχείο από τον τύπο Hun1. Αυτή η νέα μέθοδος τυχαίας επιλογής είναι πολύ αποτελεσματική, γιατί δεν υπολογίζει τις καταχωρήσεις που δε θα χρησιμοποιηθούν.

Με αυτές τις τροποποιήσεις, παίρνουμε τα αποτελέσματα προσομοίωσης που φαίνονται στο Σχήμα 31. Κάθε προσομοίωση διήρκεσε 2-10 δευτερόλεπτα και επαναλήφθηκε δέκα φορές, ώστε να μας δώσει τη μέση τιμή και την τυπική απόκλιση του χρόνου και των βημάτων που χρησιμοποιήθηκαν για να μεταφερθούν εκατό πακέτα. Ο αριθμός των βημάτων είναι ένα μέτρο για τους υπολογιστικούς πόρους που χρησιμοποιεί το πρωτόκολλο και είναι ανάλογος με τον αριθμό των εκτελέσεων των μεταβάσεων TransmitPacket και TransmitAcknowledgement και άρα, ανάλογος με τη χρήση του **εύρους ζώνης (bandwidth)** του δικτύου.

Success Rate: 90 %		
Wait	Time used	Number of steps
20	13,092 ± 345	2,154 ± 54
50	13,931 ± 275	1,260 ± 22
100	14,850 ± 510	871 ± 18
200	16,983 ± 1,275	565 ± 17

Success Rate: 99 %		
Wait	Time used	Number of steps
20	13,750 ± 177	2,405 ± 38
50	13,576 ± 317	1,341 ± 26
100	13,867 ± 246	916 ± 22
200	13,416 ± 308	504 ± 4

Σχήμα 31: Τα αποτελέσματα προσομοίωσης για το απλό πρωτόκολλο επικοινωνίας

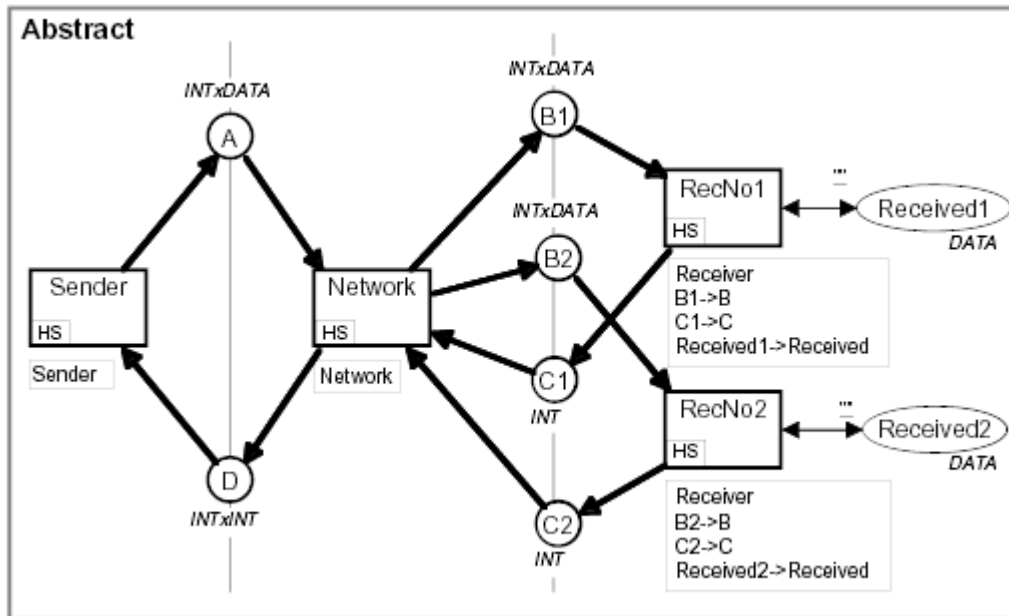
Από τα αποτελέσματα της προσομοίωσης φαίνεται ότι σε ένα όχι και τόσο αξιόπιστο δίκτυο με 10% απώλειες, υπάρχει μια καθαρή αναλογία μεταξύ του εύρους ζώνης και του χρόνου μετάδοσης. Αν γίνονται πιο συχνά αναμεταδόσεις, χρησιμοποιείται περισσότερο εύρος ζώνης κι επιτυγχάνονται μικρότεροι χρόνοι μετάδοσης. Σε ένα πιο αξιόπιστο δίκτυο με 1% απώλειες, η κατάσταση είναι διαφορετική κι έτσι δεν υπάρχει κέρδος σε χρόνο μετάδοσης, όταν γίνονται συχνότερες αναμεταδόσεις.

4.5 Ιεραρχικά Χρωματισμένα Δίκτυα Petri

Η βασική ιδέα πίσω από τα Ιεραρχικά Χρωματισμένα δίκτυα Petri είναι να παρέχεται σε αυτόν που μοντελοποιεί ένα σύστημα, η δυνατότητα να κατασκευάσει ένα μεγάλο μοντέλο, χρησιμοποιώντας μικρότερα Χρωματισμένα δίκτυα Petri, τα οποία συνδέονται το ένα με το άλλο με ένα καλά ορισμένο τρόπο. Με τον ίδιο τρόπο, ένας προγραμματιστής συντάσσει ένα μεγάλο πρόγραμμα από πολλά μικρότερα υποπρογράμματα. Πολλά μοντέλα Ιεραρχικών Χρωματισμένων δικτύων Petri αποτελούνται από περισσότερα από εκατό Χρωματισμένα δίκτυα Petri με μερικές εκατοντάδες θέσεις και μεταβάσεις. Χωρίς τη δυνατότητα ιεραρχικής δομής, ένα τέτοιο μοντέλο θα έπρεπε να σχεδιαστεί σαν ένα πολύ μεγάλο Χρωματισμένο δίκτυο Petri, από το οποίο δεν θα έβγαινε κανένα νόημα.

Σε ένα Ιεραρχικό Χρωματισμένο δίκτυο Petri είναι δυνατό να συσχετιστεί μια μετάβαση και τα τόξα και οι θέσεις που την περιβάλλουν, με ένα ξεχωριστό Χρωματισμένο δίκτυο Petri, το οποίο παρέχει μια ακριβέστερη και πιο λεπτομερή περιγραφή της δράσης που παριστάνεται από τη μετάβαση. Η ιδέα αυτή είναι ανάλογη με τις δομές ιεραρχίας, όπως είναι για παράδειγμα τα **διαγράμματα ροής δεδομένων (data flow charts)**, που υπάρχουν σε πολλές περιγραφικές γλώσσες. Από μία άποψη, είναι ανάλογη και με την έννοια των **αυτοτελών μονάδων (modules)** που συναντώνται σε πολλές σύγχρονες γλώσσες προγραμματισμού. Σε ένα επίπεδο, θέλουμε να δώσουμε μια απλή περιγραφή της δράσης που μοντελοποιείται, ενώ σε ένα άλλο επίπεδο θέλουμε να προσδιορίσουμε με μεγαλύτερη λεπτομέρεια τη συμπεριφορά της δράσης. Επιπλέον, θέλουμε να συνδέσουμε την απλή περιγραφή με την πιο λεπτομερή, με τέτοιο τρόπο ώστε να βγαίνει νόημα από την περιγραφή της συμπεριφοράς που παρουσιάζει συνολικά το σύστημα.

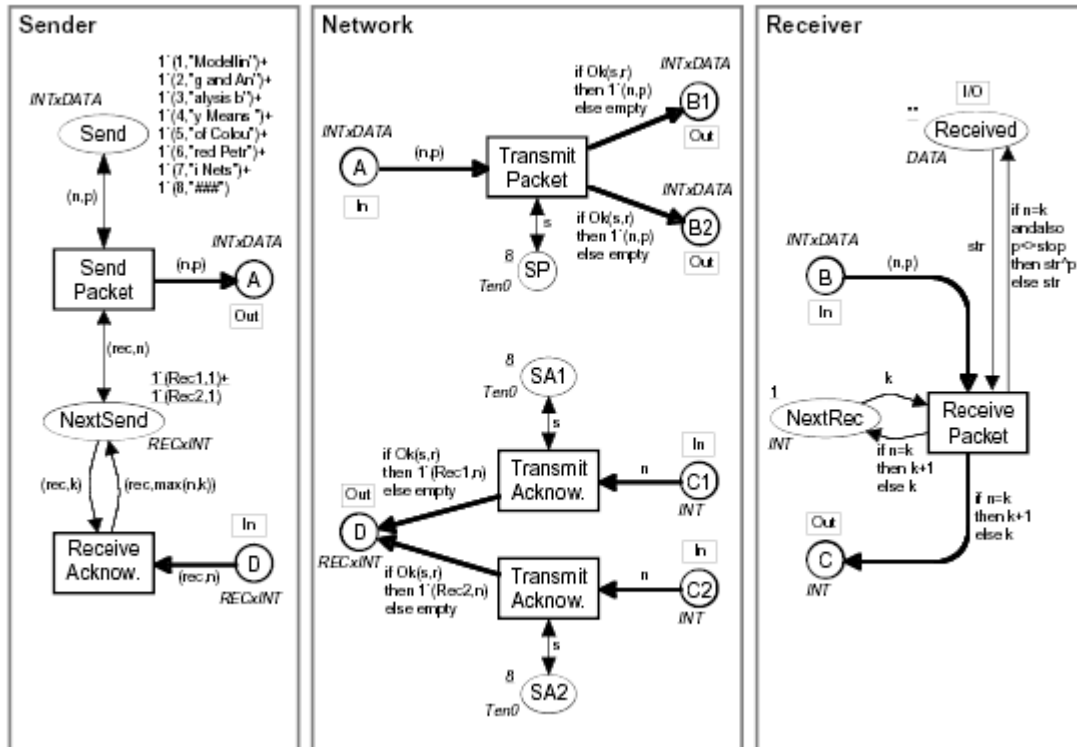
Στη συνέχεια θα περιγράψουμε μια ιεραρχική έκδοση του απλού πρωτοκόλλου επικοινωνίας. Η πιο **συνοπτική (abstract)** περιγραφή του πρωτοκόλλου φαίνεται στο Σχήμα 32. Όπως και προηγουμένως, έχουμε έναν αποστολέα Sender και ένα δίκτυο Network, αλλά τώρα υπάρχουν δύο διαφορετικοί παραλήπτες : ο RecNo1 και ο RecNo2. Το Ιεραρχικό Χρωματισμένο δίκτυο Petri του Σχήματος 32 έχει οχτώ θέσεις και τέσσερις μεταβάσεις. Κάθε μία από τις μεταβάσεις έχει μια **ετικέτα (tag) HS (HS ≈ Hierarchy + Substitution)**, η οποία υποδεικνύει ότι είναι μια **μετάβαση αντικατάστασης (substitution transition)**. Τα διακεκομμένα ορθογώνια δίπλα στις ετικέτες HS, ονομάζονται **επιγραφές ιεραρχίας (hierarchy inscriptions)** και ορίζουν τις λεπτομέρειες των αντικαταστάσεων.



Σχήμα 32: Η πιο συνοπτική σελίδα σε μια ιεραρχική έκδοση του απλού πρωτοκόλλου επικοινωνίας

Η πρώτη γραμμή κάθε επιγραφής ιεραρχίας καθορίζει την **υποσελίδα (subpage)**, δηλαδή το Χρωματισμένο δίκτυο Petri που περιέχει τη λεπτομερή περιγραφή της δράσης που παριστάνεται από την αντίστοιχη μετάβαση αντικατάστασης. Στο παράδειγμα μας, Η μετάβαση Sender έχει μία υποσελίδα με το όνομά της και το ίδιο ισχύει και για τη μετάβαση Network. Οι μεταβάσεις RecNo1 και RecNo2 έχουν αμφότερες μία υποσελίδα με το όνομα Receiver. Κατά τη διάρκεια της εκτέλεσης του Χρωματισμένου δικτύου Petri, θα υπάρχουν δύο διαφορετικά στιγμιότυπα της **σελίδας (page)** Receiver, μία για κάθε μετάβαση αντικατάστασης. Κάθε ένα από αυτά τα **στιγμιότυπα σελίδας (page instances)** θα έχει το δικό του μαρκάρισμα, το οποίο είναι εντελώς ανεξάρτητο από το μαρκάρισμα του άλλου στιγμιότυπου σελίδας, όπως ακριβώς σε μια γλώσσα προγραμματισμού, κάθε **διαδικασία (procedure)** έχει τα δικά της ιδιωτικά αντίγραφα τοπικών μεταβλητών.

Στο Σχήμα 33 φαίνονται οι υποσελίδες των μεταβάσεων αντικατάστασης που είναι παρόμοιες με τα τμήματα Sender, Network και Receiver του πρωτοκόλλου επικοινωνίας του Σχήματος 1. Κάθε υποσελίδα έχει έναν αριθμό από θέσεις, οι οποίες είναι μαρκαρισμένες με μία ετικέτα In, Out ή I/O. Οι θέσεις αυτές ονομάζονται **θέσεις πύλης (port places)** και αποτελούν τη διαπροσωπεία μέσω της οποίας η υποσελίδα επικοινωνεί με τον περίγυρο της. Μία θέση πύλης με ετικέτα In ονομάζεται **πύλη εισόδου (input port)**, ενώ μία θέση πύλης με ετικέτα Out ονομάζεται **πύλη εξόδου (output port)**. Μία θέση πύλης με ετικέτα I/O είναι και πύλη εισόδου και πύλη εξόδου. Μέσω των πυλών εισόδου, η υποσελίδα λαμβάνει μάρκες από το περιβάλλον της, ενώ αντίθετα μέσω των πυλών εξόδου η υποσελίδα στέλνει μάρκες στο περιβάλλον της.



Σχήμα 33: Οι τρεις υποσελίδες που χρησιμοποιούνται από τις μεταβάσεις αντικατάστασης του Σχήματος 32

Η μετάβαση αντικατάστασης Sender στο Σχήμα 32 έχει μια μόνο θέση εισόδου D και μία μόνο θέση εξόδου A. Οι θέσεις αυτές ονομάζονται **θέσεις υποδοχής (socket places)**. Συγκεκριμένα, η θέση D είναι μια **υποδοχή εισόδου (input socket)** για τη μετάβαση Sender, ενώ η θέση A είναι μια **υποδοχή εξόδου (output socket)**. Για να προσδιορίσουμε τη σχέση μεταξύ της μετάβασης αντικατάστασης και της υποσελίδας της, πρέπει να περιγράψουμε, πώς οι θέσεις πύλης της υποσελίδας συνδέονται με τις θέσεις υποδοχής της μετάβασης αντικατάστασης. Αυτό γίνεται με τη βοήθεια της ανάθεσης πύλης (port assignment). Στην υποσελίδα Sender, Η πύλη εισόδου D στο Σχήμα 32 συνδέεται με την υποδοχή εισόδου D στο Σχήμα 33. Ομοίως, η πύλη εξόδου A στο Σχήμα 32 συνδέεται με την υποδοχή εξόδου στο Σχήμα 33. Οι σχέσεις μεταξύ των πυλών και των υποδοχών αναγράφονται στις επιγραφές ιεραρχίας. Όταν η πύλη και η υποδοχή έχουν το ίδιο όνομα, οι αναθέσεις πύλης παραλείπονται, ώστε να αυξηθεί η αναγνωσιμότητα και η συντομία του δικτύου. Έτσι οι αναθέσεις $A \rightarrow A$ και $D \rightarrow D$ δεν συμπεριλαμβάνονται στην επιγραφή ιεραρχίας της μετάβασης αντικατάστασης Sender.

Στην υποσελίδα Network, οι θέσεις A, C1 και C2 είναι υποδοχές εισόδου, ενώ οι θέσεις B1, B2 και D είναι υποδοχές εξόδου. Η υποσελίδα έχει έξι θέσεις πύλης, κάθε μία από τις οποίες έχει το ίδιο όνομα με τη θέση υποδοχής με την οποία συνδέεται κι έτσι, δεν αναγράφουμε τις αναθέσεις πύλης στην επιγραφή ιεραρχίας.

Η κατάσταση είναι λίγο πιο ενδιαφέρουσα για τις υπόλοιπες δύο μεταβάσεις αντικατάστασης. Η μετάβαση αντικατάστασης RecNo1 έχει τις αναθέσεις πύλης B1

→ B, C1 → C και Received1 → Received, ενώ η μετάβαση αντικατάστασης RecNo2 έχει τις αναθέσεις πύλης B2 → B, C2 → C και Received2 → Received.

Όταν μια θέση πύλης ανατίθεται σε μια θέση υποδοχής, οι δύο θέσεις γίνονται ίδιες. Η θέση πύλης και η θέση υποδοχής είναι απλώς δύο διαφορετικές αναπαραστάσεις της ίδιας νοητής θέσης. Αυτό σημαίνει οι θέσεις πύλης και υποδοχής έχουν πάντα το ίδιο μαρκάρισμα. Όταν μια υποδοχή εισόδου λαμβάνει μία μάρκα από το περιβάλλον της μετάβασης αντικατάστασης, η μάρκα αυτή γίνεται επίσης διαθέσιμη και στην πύλη εισόδου της υποσελίδας, και άρα μπορεί να χρησιμοποιηθεί από τις μεταβάσεις της υποσελίδας. Ανάλογα, η υποσελίδα μπορεί να παράγει στην πύλη εξόδου μάρκες, οι οποίες είναι διαθέσιμες και στην αντίστοιχη υποδοχή εξόδου και άρα μπορούν να χρησιμοποιηθούν από το περιβάλλον της μετάβασης αντικατάστασης. Στο παράδειγμα μας έχουμε τρεις διαφορετικές αναπαραστάσεις της θέσης A : μία στο συνοπτικό δίκτυο, μία στην υποσελίδα Sender και μία στην υποσελίδα Network. Το ίδιο ισχύει και για τις θέσεις B1, B2, C1, C2 και D, ενώ οι θέσεις Received1 και Received2 έχουν μόνο δύο αναπαραστάσεις.

Στη συνέχεια της παραγράφου θα αναφερθούμε με μεγαλύτερη λεπτομέρεια στις τρεις υποσελίδες του Σχήματος 33. Η βασική ιδέα είναι ότι ο αποστολέας Sender στέλνει μηνύματα, ενώ το δίκτυο Network **εκπέμπει (broadcasts)** στους δύο παραλήπτες Receivers, οι οποίοι με τη σειρά τους στέλνουν βεβαιώσεις λήψης πακέτων, τις οποίες το δίκτυο Network μεταδίδει στον αποστολέα Sender.

Η υποσελίδα Sender είναι παρόμοια με το τμήμα Sender του Σχήματος 1. Η βασική διαφορά είναι ότι η θέση NextSend μοντελοποιεί δύο μετρητές, ένα για κάθε παραλήπτη Receiver. Κάθε βεβαίωση λήψης πακέτου είναι ένα ζεύγος, όπου το πρώτο στοιχείο καθορίζει εάν η βεβαίωση λήψης πακέτου προήλθε από τον παραλήπτη RecNo1 ή RecNo2, ενώ το δεύτερο στοιχείο περιέχει τον αριθμό του επόμενου πακέτου που θέλει να παραλάβει ο παραλήπτης Receiver. Όταν παραλαμβάνεται μια βεβαίωση λήψης πακέτου (rec, n), ο μετρητής του αντίστοιχου παραλήπτη Receiver ενημερώνεται. Η παλιά τιμή του μετρητή k αντικαθίσταται από τη τιμή max(n, k). Εφ'όσον τα πακέτα αποστέλλονται μέσω εκπομπής, ο αποστολέας Sender πρέπει να στείλει το ίδιο πακέτο και στους δύο παραλήπτες Receivers, ακόμα και στην περίπτωση που οι δύο μετρητές στη θέση NextSend έχουν διαφορετικές τιμές. Μία προφανής λύση είναι να απαιτήσουμε από τον αποστολέα Sender να χρησιμοποιεί τη μικρότερη από τις δύο τιμές του μετρητή, δηλαδή να ακολουθεί τις βεβαιώσεις λήψης πακέτων του πιο άτυχου παραλήπτη Receiver. Παρ'όλα αυτά, στο συγκεκριμένο παράδειγμα θα επιτρέψουμε στον αποστολέα Sender να χρησιμοποιεί τυχαία μία από τις δύο τιμές του μετρητή. Για ένα δίκτυο με πολλές απώλειες, αυτό είναι λιγότερο αποδοτικό, αλλά επιτρέπει στον ένα από τους δύο παραλήπτες να παραλάβουν ολόκληρο το μήνυμα, ακόμα και αν ο άλλος παραλήπτης σταματήσει να λειτουργεί.

Η υποσελίδα Network είναι παρόμοια με το τμήμα Network του Σχήματος 1. Υπάρχουν όμως, πάλι, μερικές διαφορές. Η μετάβαση TransmitPacket παράγει πακέτα σε δύο διαφορετικές θέσεις εξόδου, τις B1 και B2. Τα πακέτα στη θέση B1 είναι για τον παραλήπτη RecNo1, ενώ τα πακέτα στη θέση B2 είναι για τον παραλήπτη RecNo2. Για να καθοριστεί αν τα πακέτα για τις θέσεις B1 και B2 έχουν χαθεί ή όχι, χρησιμοποιείται η ίδια μεταβλητή r. Αυτό σημαίνει ότι μοντελοποιείται μια εκπομπή στην οποία εγγυάται ότι όλοι οι παραλήπτες Receivers θα παραλάβουν

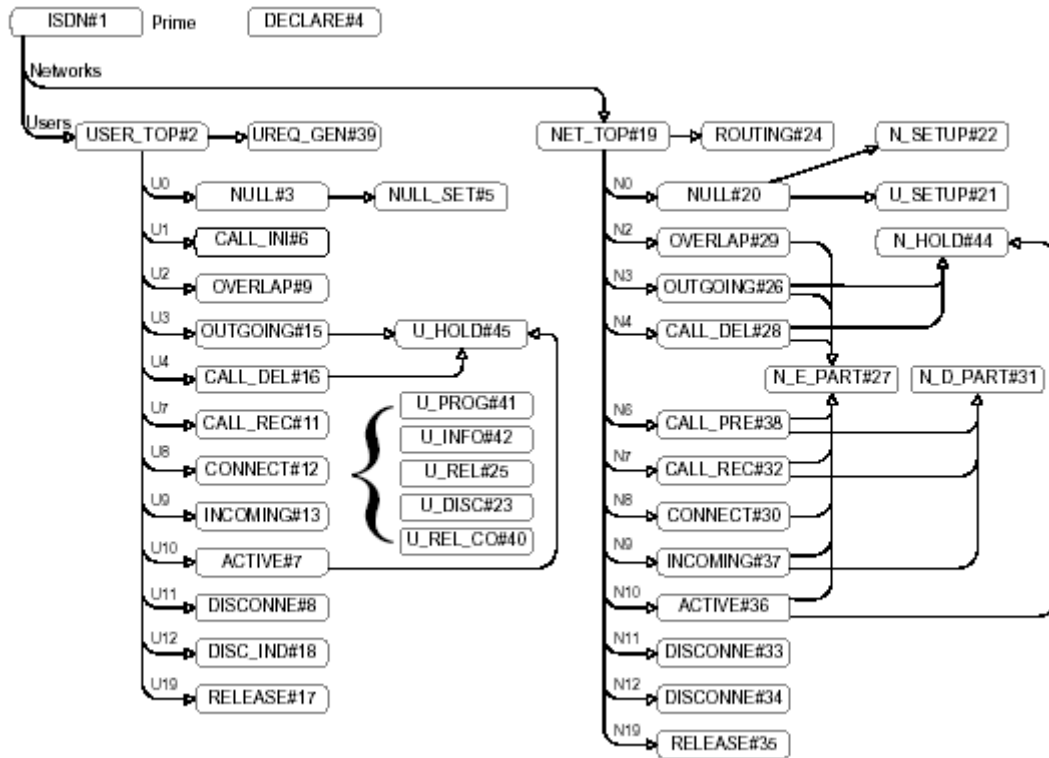
τα ίδια πακέτα. Αν η μεταβλητή r αντικατασταθεί με δύο διαφορετικές μεταβλητές r_1 και r_2 , θα πάρουμε μία εκπομπή όπου ο ένας παραλήπτης Receiver μπορεί να παραλάβει ένα πακέτο, ενώ ο άλλος όχι. Η μετάβαση TransmitAcknowledgement έχει τώρα χωριστεί στα δύο. Η πάνω μετάβαση χειρίζεται βεβαιώσεις λήψης πακέτων από τον παραλήπτη RecNo1, ενώ κάτω μετάβαση χειρίζεται βεβαιώσεις λήψης πακέτων από τον παραλήπτη RecNo2. Αμφότερες οι μεταβάσεις προσθέτουν πληροφορία σχετικά με το από πού προήλθε η βεβαίωση λήψης πακέτου.

Η υποσελίδα Receiver είναι εντελώς ίδια με το τμήμα Receiver του Σχήματος 1. Παρ'όλα αυτά θα πρέπει να σημειωθεί ότι η σελίδα Receiver χρησιμοποιείται από δύο μεταβάσεις αντικατάστασης, τις RecNo1 και RecNo2. Όπως αναφέρθηκα προηγουμένως, αυτό σημαίνει ότι κατά τη διάρκεια μιας εκτέλεσης θα υπάρχουν δύο στιγμιότυπα του **υποδικτύου (subnet)**. Τα δύο στιγμιότυπα μπορεί να έχουν διαφορετικά μαρκαρίσματα και διαφορετική ενεργοποίηση. Αλλιώς θα είναι ίδια.

Στο παράδειγμα του πρωτοκόλλου, υπάρχουν μόνο δύο επίπεδα στην **ιεραρχία σελίδων (page hierarchy)**, η συνοπτική σελίδα του Σχήματος 32 και οι τρεις υποσελίδες του Σχήματος 33. Στην πράξη όμως, υπάρχουν συνήθως μέχρι δέκα διαφορετικά επίπεδα ιεραρχίας. Μία υποσελίδα μπορεί να περιέχει μεταβάσεις αντικατάστασης κι έτσι να περιέχει τις δικές της υποσελίδες. Μια συνηθισμένη περίπτωση είναι, να έχει μια σελίδα και απλές μεταβάσεις και μεταβάσεις αντικατάστασης, δηλαδή κάποιες δράσεις περιγράφονται με όλες τις λεπτομέρειες, ενώ κάποιες άλλες περιγράφονται με ένα πιο απλό τρόπο.

4.5.1 Γράφος Ιεραρχίας Σελίδων

Για να δώσουμε μια συνολική εικόνα της σχέσης μεταξύ των διαφορετικών σελίδων σε ένα Ιεραρχικό Χρωματισμένο δίκτυο Petri, χρησιμοποιούμε ένα **γράφο ιεραρχίας σελίδων (page hierarchy graph)**, όπως αυτός που φαίνεται στο Σχήμα 34. Ένας τέτοιος γράφος περιέχει ένα κόμβο για κάθε σελίδα. Ένα τόξο μεταξύ δύο σελίδων υποδεικνύει ότι η δεύτερη είναι υποσελίδα της πρώτης, δηλαδή ότι η σελίδα πηγής περιέχει μια μετάβαση αντικατάστασης που χρησιμοποιεί τη σελίδα προορισμού ως υποσελίδα. Το κείμενο που είναι γραμμένο μέσα σε κάθε κόμβο προσδιορίζει το όνομα και τον αριθμό της σελίδας. Ανάλογα, το κείμενο που συνοδεύει κάθε τόξο προσδιορίζει το όνομα της προκείμενης μετάβασης αντικατάστασης. Παραδείγματος χάριν, στο Σχήμα 34 η σελίδα User_Top#2 έχει δεκατρείς μεταβάσεις αντικατάστασης. Μία από αυτές, η U4 χρησιμοποιεί τη σελίδα Call_Del#16 ως υποσελίδα.



Σχήμα 34: Παράδειγμα μιας πιο σύνθετης ιεραρχίας σελίδων

Ο γράφος ιεραρχίας σελίδων του Σχήματος 34 ανήκει σε Ιεραρχικό Χρωματισμένο δίκτυο Petri που περιγράφει ένα πρωτόκολλο για τα τηλεφωνικά δίκτυα ISDN. Η μεγάλη αγκύλη υποδεικνύει ότι κάθε μία από τις πέντε σελίδες στα δεξιά της αγκύλης είναι μια υποσελίδα όλων (ή σχεδόν όλων) των δώδεκα σελίδων στα αριστερά της αγκύλης. Άρα, υπάρχουν σχεδόν εξήντα στιγμιότυπα σελίδας μόνο για αυτές τις πέντε σελίδες. Επίσης, μερικές από τις σελίδες στο δεξί μέρος του γράφου ιεραρχίας σελίδων έχουν πολλαπλά στιγμιότυπα. Μία από αυτές έχει οχτώ στιγμιότυπα σελίδας, ενώ άλλες δύο σελίδες έχουν από τρία στιγμιότυπα η κάθε μία. Συνολικά το Ιεραρχικό Χρωματισμένο δίκτυο Petri του Σχήματος 34 έχει σαράντα δύο σελίδες και περίπου εκατό στιγμιότυπα σελίδας. Η σελίδα ISDN#1 (στην πάνω αριστερή γωνία) έχει δίπλα της, τη λέξη Prime. Αυτό υποδεικνύει ότι η σελίδα ISDN#1 είναι μια **κύρια (prime)** σελίδα, δηλαδή μια σελίδα στο πιο αφαιρετικό επίπεδο. Ένα Ιεραρχικό Χρωματισμένο δίκτυο Petri έχει ένα στιγμιότυπο σελίδας για κάθε κύρια σελίδα. Για κάθε μετάβαση αντικατάστασης σε μια κύρια σελίδα, υπάρχει ένα στιγμιότυπο σελίδας της αντίστοιχης υποσελίδας. Εάν αυτά τα στιγμιότυπα σελίδας έχουν μεταβάσεις αντικατάστασης, υπάρχουν στιγμιότυπα σελίδας για αυτές και ούτω καθεξής, μέχρι να φτάσουμε στη βάση της ιεραρχίας σελίδων που δεν πρέπει να είναι κυκλική.

Μπορεί να αποδειχτεί ότι κάθε Ιεραρχικό Χρωματισμένο δίκτυο Petri έχει ένα ισοδύναμο, ως προς τη συμπεριφορά, μη Ιεραρχικό Χρωματισμένο δίκτυο Petri. Για να πάρουμε το μη ιεραρχικό δίκτυο, αντικαθιστούμε απλά κάθε μετάβαση αντικατάστασης και τα περιβάλλοντα τόξα της με ένα αντίγραφο της υποσελίδας της, “κολλώντας” κάθε θέση πύλης με τη θέση υποδοχής με την οποία συνδέεται.

Πρέπει να σημειωθεί ότι οι μεταβάσεις αντικατάστασης δεν ενεργοποιούνται ούτε κι εκτελούνται ποτέ. Οι μεταβάσεις αντικατάστασης λειτουργούν όπως ένας μακρομηχανισμός, δηλαδή επιτρέπουν στις υποσελίδες να εισαχθούν ιδεατά στη θέση των μεταβάσεων αντικατάστασης. Στο Σχήμα 32 δεν υπάρχουν επιγραφές τόξων στα τόξα που περιβάλλουν τις μεταβάσεις αντικατάστασης, γιατί δεν είναι απαραίτητες στην προσομοίωση και την ανάλυση του χώρου καταστάσεων, εφ'όσον οι μεταβάσεις αντικατάστασης ούτε ενεργοποιούνται, ούτε εκτελούνται. Παρ'όλα αυτά μπορεί να είναι πολύ χρήσιμες, γιατί μπορούν να δώσουν στον αναγνώστη ενός μοντέλου μια πρώτη εντύπωση της λειτουργικότητας της υποσελίδας. Επιπλέον, είναι δυνατό να αγνοηθεί προσωρινά μία ή περισσότερες μεταβάσεις αντικατάστασης. Τότε οι μεταβάσεις συμπεριφέρονται σαν κοινές μεταβάσεις. Με αυτόν τον τρόπο και μεταβάλλοντας το σύνολο των κυρίων σελίδων, είναι εύκολο να διορθωθούν επιλεγμένες σελίδες ενός μεγάλου Ιεραρχικού Χρωματισμένου δικτύου Petri, χωρίς να “αποκοπούν” από το μοντέλο.

4.5.2 Θέσεις Συγχώνευσης

Στα Ιεραρχικά Χρωματισμένα δίκτυα Petri υπάρχουν επίσης οι **θέσεις συγχώνευσης (fusion places)**. Αυτός που μοντελοποιεί ένα σύστημα μπορεί να θεωρήσει ότι ένα σύνολο από θέσεις είναι ίδιες, δηλαδή ότι όλες αναπαριστούν μία μόνο ιδεατή θέση, ακόμα και αν σχεδιάζονται σαν ξεχωριστές θέσεις. Όταν μία μάρκα προστίθεται ή αφαιρείται σε μία από τις θέσεις, μία ίδια μάρκα θα προστεθεί ή θα αφαιρεθεί σε όλες τις άλλες θέσεις που ανήκουν στο **σύνολο συγχώνευσης (fusion set)**. Είναι φανερό ότι η σχέση μεταξύ των μελών ενός συνόλου συγχώνευσης είναι παρόμοια με τη σχέση μεταξύ δύο θέσεων, οι οποίες ανατίθενται η μία στην άλλη από μία πύλη ανάθεσης.

Όταν όλα τα μέλη ενός συνόλου συγχώνευσης ανήκουν σε μία μόνο σελίδα και η σελίδα αυτή έχει μόνο ένα στιγμιότυπο σελίδας, η θέση δεν είναι τίποτα άλλο παρά μια βολική σχεδίαση η οποία επιτρέπει στο χρήστη να αποφύγει πολλά τεμνόμενα τόξα. Τα πράγματα αποκτούν περισσότερο ενδιαφέρον, όταν τα μέλη ενός συνόλου συγχώνευσης ανήκουν σε αρκετές διαφορετικές σελίδες ή όταν ανήκουν σε μια σελίδα με αρκετά στιγμιότυπα σελίδας. Σε αυτήν την περίπτωση, τα σύνολα συγχώνευσης επιτρέπουν στο χρήστη να προσδιορίσει μια συμπεριφορά που μπορεί να είναι δύσκολο να περιγραφεί χωρίς συγχώνευση.

Υπάρχουν τρία διαφορετικά είδη συνόλων : τα **καθολικά σύνολα συγχώνευσης (global fusion sets)** μπορούν να έχουν μέλη από πολλές διαφορετικές σελίδες, ενώ τα **σύνολα συγχώνευσης σελίδας (page fusion sets)** και τα **σύνολα συγχώνευσης στιγμιότυπου (instance fusion sets)** έχουν μέλη μόνο από μία σελίδα. Η διαφορά μεταξύ των δύο τελευταίων συνόλων συγχώνευσης είναι η εξής : η συγχώνευση σελίδας ενοποιεί όλα τα στιγμιότυπα των θέσεων της, ανεξάρτητα από το στιγμιότυπο της σελίδας στο οποίο ανήκουν, και αυτό σημαίνει ότι το σύνολο συγχώνευσης έχει μόνο μία **“θέση κατάληξης” (“resulting place”)** την οποία “μοιράζονται” όλα τα στιγμιότυπα της αντίστοιχης σελίδας. Αντίθετα, ένα σύνολο συγχώνευσης στιγμιότυπου αναγνωρίζει μόνο τα στιγμιότυπα θέσης που ανήκουν στο ίδιο στιγμιότυπο σελίδας, και αυτό σημαίνει ότι το σύνολο θέσης έχει μία **“θέση κατάληξης”** για κάθε στιγμιότυπο σελίδας. Η σημασία ενός καθολικού συνόλου συγχώνευσης είναι παρόμοια με αυτή συνόλου συγχώνευσης σελίδας, με την έννοια

ότι υπάρχει μόνο μία “θέση κατάληξης”, κοινή για όλα τα στιγμιότυπα όλων των σελίδων που ανήκουν στο σύνολο συγχώνευσης. Για να είναι δυνατή η σπονδυλωτή ανάλυση των Ιεραρχικών Χρωματισμένων δικτύων Petri, τα καθολικά σύνολα συγχώνευσης πρέπει να χρησιμοποιούνται με μέτρο.

Η βασική ιδέα πίσω από τα Ιεραρχικά Χρωματισμένα δίκτυα Petri είναι να δοθεί η δυνατότητα σε αυτόν που μοντελοποιεί το σύστημα, να δομήσει ένα μεγάλο μοντέλο συνδυάζοντας πολλά μικρά Χρωματισμένα δίκτυα Petri σε ένα μοναδικό μοντέλο. Με παρόμοιο τρόπο, ένας προγραμματιστής συντάσσει ένα μεγάλο πρόγραμμα από ένα σύνολο υποπρογραμμάτων. Η ιδέα όμως είναι διαφορετική από εκείνες τις προσεγγίσεις οι οποίες συσχετίζουν δύο ή περισσότερα υποδίκτυα μεταξύ τους - ώστε να συγκρίνουν τη συμπεριφορά τους - χωρίς όμως να τα συνδυάζουν σε ένα μοναδικό μοντέλο. Σε τέτοιες προσεγγίσεις, τα ξεχωριστά υποδίκτυα είναι εναλλακτικές περιγραφές του ίδιου συστήματος.

4.6 Συμπεράσματα

Τα Χρωματισμένα δίκτυα Petri δεν υπερέχουν όλων των άλλων γλώσσών μοντελοποίησης. Απλά είναι εξαιρετικά χρήσιμα και μαζί με άλλες γλώσσες μοντελοποίησης παίζουν σημαντικό ρόλο στη σχεδίαση και ανάλυση προηγμένων συστημάτων. Συνοπτικά οι λόγοι που χρησιμοποιούνται Χρωματισμένα δίκτυα Petri είναι οι εξής :

1. Τα Χρωματισμένα δίκτυα Petri παριστάνονται γραφικά. Η γραφική απεικόνιση είναι διαιθητικά πολύ ελκυστική. Επειδή τα διαγράμματα των Χρωματισμένων δικτύων Petri μοιάζουν με πολλά από τα σχέδια που κάνουν οι σχεδιαστές και οι μηχανικοί, καθώς κατασκευάζουν και αναλύουν ένα σύστημα, τα Χρωματισμένα δίκτυα Petri γίνονται πολύ εύκολα κατανοητά, ακόμα και από άτομα που δεν είναι εξοικειωμένα με αυτά.
2. Τα Χρωματισμένα δίκτυα Petri έχουν μια καλά ορισμένη σημασιολογία η οποία ορίζει με σαφήνεια τη συμπεριφορά κάθε δικτύου. Λόγω αυτής της σημασιολογίας, γίνεται δυνατή η προσομοίωση των Χρωματισμένων δικτύων Petri και η εφαρμογή των μεθόδων ανάλυσης.
3. Τα Χρωματισμένα δίκτυα Petri είναι πολύ γενικά και μπορούν να χρησιμοποιηθούν για να περιγράψουν μια μεγάλη ποικιλία από διαφορετικά συστήματα. Οι εφαρμογές των Χρωματισμένων δικτύων Petri περιέχουν τόσο άτυπα συστήματα, όπως η περιγραφή μιας διεργασίας, όσο και τυπικά συστήματα, όπως ένα πρωτόκολλο επικοινωνίας. Επίσης, περιέχουν και συστήματα λογισμικού, όπως οι κατανεμημένοι αλγόριθμοι, και συστήματα υλικού, όπως τα τσιπ VLSI.
4. Τα Χρωματισμένα δίκτυα Petri έχουν πολύ λίγες, αλλά ισχυρές αρχές. Ο ορισμός των Χρωματισμένων δικτύων Petri είναι αρκετά σύντομος και δομείται πάνω σε καθιερωμένες έννοιες, τις οποίες γνωρίζουν πολλοί σχεδιαστές συστημάτων, από τα απλά μαθηματικά και από τις γλώσσες προγραμματισμού. Αυτό σημαίνει ότι είναι σχετικά εύκολη η εκμάθηση χρήσης των Χρωματισμένων δικτύων Petri και ότι είναι δυνατό να αναπτυχθούν ισχυρές μέθοδοι ανάλυσης.
5. Τα Χρωματισμένα δίκτυα Petri περιγράφουν με σαφήνεια και τις καταστάσεις και τις δράσεις, αντίθετα με τις περισσότερες γλώσσες περιγραφής

συστημάτων που περιγράφουν είτε τις καταστάσεις, είτε τις δράσεις, αλλά όχι και τα δύο μαζί. Χρησιμοποιώντας Χρωματισμένα δίκτυα Petri, ο αναγνώστης μπορεί εύκολα να μετατοπίσει το επίκεντρο της προσοχής του, από τις καταστάσεις στις δράσεις, και αντίστροφα.

6. Τα Χρωματισμένα δίκτυα Petri έχουν μια σημασιολογία που δομείται πάνω στην πραγματικά ταυτόχρονη εκτέλεση αντί πάνω στη **χρονική διαμεσολάβηση (interleaving)**. Σε μια χρονική διαμεσολάβηση είναι αδύνατο να υπάρχουν δύο δράσεις στο ίδιο βήμα, κι έτσι αυτό σημαίνει ότι οι δράσεις μπορούν να εκτελεστούν μόνο η μία μετά την άλλη, με οποιαδήποτε σειρά. Είναι ευκολότερο να δουλέψει κανείς με πραγματικά ταυτόχρονη εκτέλεση, γιατί είναι πιο κοντά στον τρόπο με τον οποίο σκέφτονται οι άνθρωποι για τις ταυτόχρονες δράσεις.
7. Τα Χρωματισμένα δίκτυα Petri προσφέρουν ιεραρχικές περιγραφές. Αυτό σημαίνει ότι ένα μεγάλο Χρωματισμένο δίκτυο Petri μπορεί να δομηθεί συνδέοντας πολλά μικρά Χρωματισμένα δίκτυα Petri μεταξύ τους, με ένα καλά ορισμένο τρόπο. Οι ιεραρχικές δομές των Χρωματισμένων δικτύων Petri παίζουν παρόμοιο ρόλο με αυτόν των υπορουτινών και των διαδικασιών στις γλώσσες προγραμματισμού. Η ύπαρξη Ιεραρχικών Χρωματισμένων δικτύων Petri κάνει δυνατή τη μοντελοποίηση μεγάλων συστημάτων με ένα εύκολο και σπονδυλωτό τρόπο.
8. Τα Χρωματισμένα δίκτυα Petri συνδυάζουν την περιγραφή του ελέγχου και του συγχρονισμού με την περιγραφή του χειρισμού δεδομένων (data manipulation). Δηλαδή ένας μοναδικός γράφος παρουσιάζει το περιβάλλον, τις συνθήκες ενεργοποίησης και τα αποτελέσματα μιας δράσης. Πολλές άλλες περιγραφικές γλώσσες χρησιμοποιούν γράφους που περιγράφουν μόνο το περιβάλλον μιας δράσης, ενώ η συμπεριφορά της περιγράφεται ξεχωριστά.
9. Τα Χρωματισμένα δίκτυα Petri μπορούν να επεκταθούν, εισάγοντας την έννοια του χρόνου. Αυτό σημαίνει ότι είναι δυνατή η χρήση της ίδιας γλώσσας μοντελοποίησης για τον καθορισμό και την αξιολόγηση, τόσο λειτουργικών και λογικών ιδιοτήτων, όπως είναι, για παράδειγμα, η απουσία αδιεξόδων, όσο και χαρακτηριστικών απόδοσης, όπως είναι η ρυθμοαπόδοση και οι χρόνοι αναμονής.
10. Τα Χρωματισμένα δίκτυα Petri παρουσιάζουν ευστάθεια απέναντι σε μικρές αλλαγές του μοντελοποιημένου συστήματος. Δηλαδή μικρές μετατροπές στο σύστημα που μοντελοποιείται, δεν μεταβάλλουν εντελώς το Χρωματισμένο δίκτυο Petri.
11. Τα Χρωματισμένα δίκτυα Petri προσφέρουν διαδραστικές προσομοιώσεις, των οποίων τα αποτελέσματα παρουσιάζονται απευθείας στο διάγραμμα του Χρωματισμένου δικτύου Petri. Με την προσομοίωση είναι δυνατή η **αποσφαλμάτωση (debugging)** ενός μεγάλου μοντέλου καθώς κατασκευάζεται, όπως ακριβώς ένας προγραμματιστής διορθώνει τα ξεχωριστά τμήματα ενός προγράμματος, αφού τα τελειώσει. Επίσης μπορούν να ελεγχθούν οι τιμές δεδομένων των μετακινούμενων μαρκών.
12. Τα Χρωματισμένα δίκτυα Petri έχουν ένα αριθμό από τυπικές μεθόδους ανάλυσης με τη βοήθεια των οποίων, μπορούν να αποδειχθούν οι ιδιότητες των Χρωματισμένων δικτύων Petri. Οι δύο πιο σημαντικές μέθοδοι ανάλυσης είναι η ανάλυση του χώρου καταστάσεων και τα **αμετάβλητα διανύσματα θέσεων (place invariants)**.
13. Τα Χρωματισμένα δίκτυα Petri έχουν ένα σύνολο από εργαλεία λογισμικού που υποστηρίζουν τη σχεδίαση, την προσομοίωση και την ανάλυση τους. Η

ύπαρξη τέτοιων εργαλείων είναι πολύ σημαντική για την πρακτική χρήση των Χρωματισμένων δικτύων Petri.

ΚΕΦΑΛΑΙΟ 5^ο

CPN Tools

5.1 Τι είναι το πρόγραμμα CPN Tools ;

Το πρόγραμμα CPN Tools είναι ένα εργαλείο για το σχεδιασμό, την προσομοίωση και την ανάλυση των Χρωματισμένων δικτύων Petri. Η **γραφική διεπιφάνεια χρήστη (graphical user interface)** βασίζεται σε προηγμένες τεχνικές διάδρασης και οι υπηρεσίες ανάδρασης παρέχουν μηνύματα σφαλμάτων και υποδεικνύουν τις σχέσεις εξάρτησης μεταξύ των στοιχείων του δικτύου. Το πρόγραμμα παρέχει επίσης συντακτικό έλεγχο και παράγει κώδικα καθώς κατασκευάζεται ένα δίκτυο. Ένας γρήγορος προσομοιωτής χειρίζεται και μη χρονισμένα δίκτυα και χρονισμένα δίκτυα Petri. Οι πλήρεις και οι μερικοί χώροι καταστάσεων μπορούν να παραχθούν και να αναλυθούν και μια τυπική αναφορά του χώρου καταστάσεων περιέχει πληροφορίες όπως είναι οι ιδιότητες της περατότητας και της διάρκειας. Η λειτουργικότητα της μηχανής προσομοίωσης και των χώρων καταστάσεων είναι παρόμοια με αυτή του προγράμματος Design/CPN, το οποίο είναι ένα ευρέως διαδεδομένο εργαλείο για τα Χρωματισμένα δίκτυα Petri και το οποίο αποτελεί πρόδρομο του CPN Tools.

Στο κεφάλαιο αυτό θα γίνει αναφορά στα βασικά στοιχεία του εργαλείου λογισμικού CPN Tools. Για μεγαλύτερη εμβάθυνση στη μελέτη του CPN Tools είναι διαθέσιμη η ιστοσελίδα του εργαλείου :

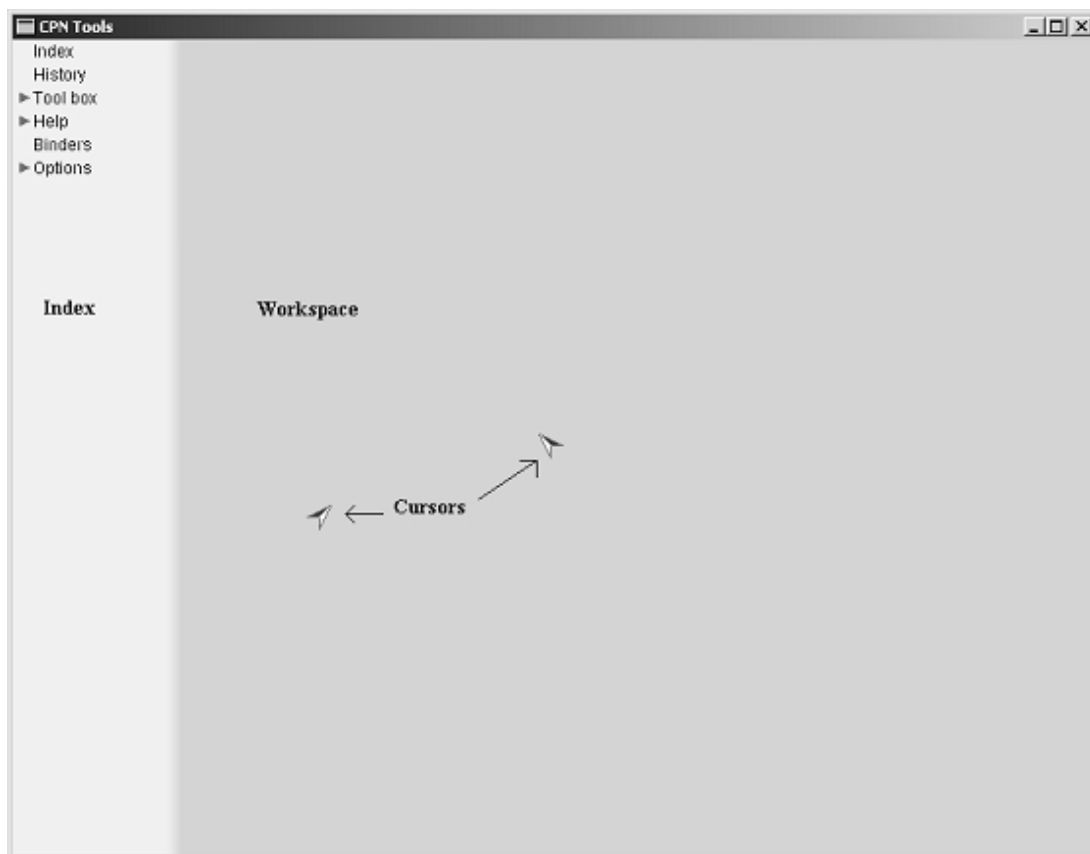
<http://www.daimi.au.dk/CPNTools/>

η οποία περιέχει πληροφορίες για το πρόγραμμα και για το πώς μπορεί κάποιος να το προμηθευτεί.

5.2 Ξεκινώντας με το CPN Tools

5.2.1 Η Διεπιφάνεια

Όταν ανοίγουμε το πρόγραμμα CPN Tools, το πρώτο πράγμα που βλέπουμε είναι ένα παράθυρο, που περιέχει ένα ευρετήριο στα αριστερά και μια επιφάνεια εργασίας στα δεξιά, όπως φαίνεται στην Εικόνα 1.



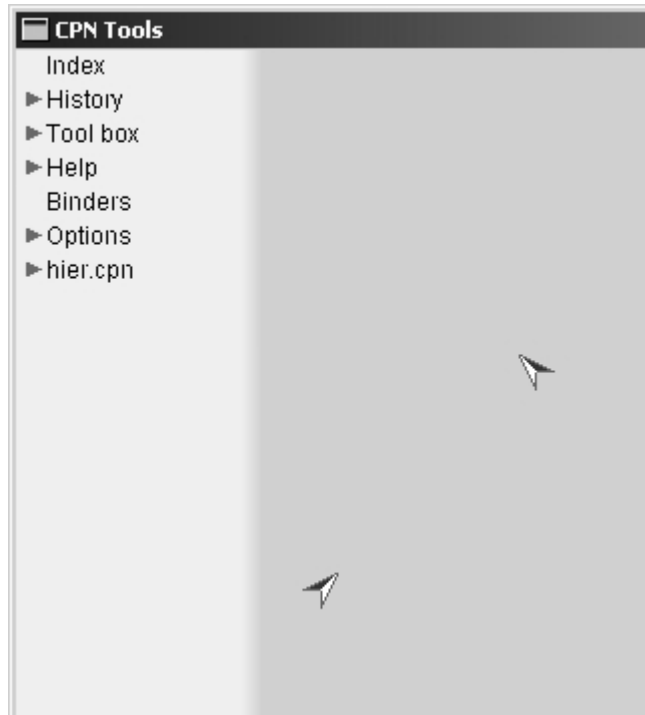
Εικόνα 1: Το παράθυρο του CPN Tools

5.2.2 Το Ευρετήριο

Το **ευρετήριο (index)** βρίσκεται στην αριστερή πλευρά του παραθύρου του CPN Tools και περιέχει :

- **History** : Λίστα με τις εντολές που έχουν εκτελεστεί
- **Tool box** : Λίστα όλων των εργαλείων που είναι διαθέσιμα στο CPN Tools.
- **Help** : Σύνδεσμοι για τις ιστοσελίδες Help, Homepage, Report Bug
- **Options** : Διάφορες επιλογές για, παραδείγματος χάριν, την προσομοίωση
- Τα δίκτυα που είναι ανοιχτά στο δίκτυο.

Ένα μπλε τρίγωνο στα αριστερά κάθε καταχώρησης στο ευρετήριο υποδεικνύει ότι η καταχώρηση μπορεί να ανοιχτεί για να αποκαλυφθούν περισσότερες λεπτομέρειες για την καταχώρηση. Στην Εικόνα 2 φαίνεται το ευρετήριο, από το οποίο μπορούμε να συμπεράνουμε ότι το δίκτυο με όνομα “hier.cpn” είναι ανοιχτό.



Εικόνα 2: Το ευρετήριο του CPN Tools

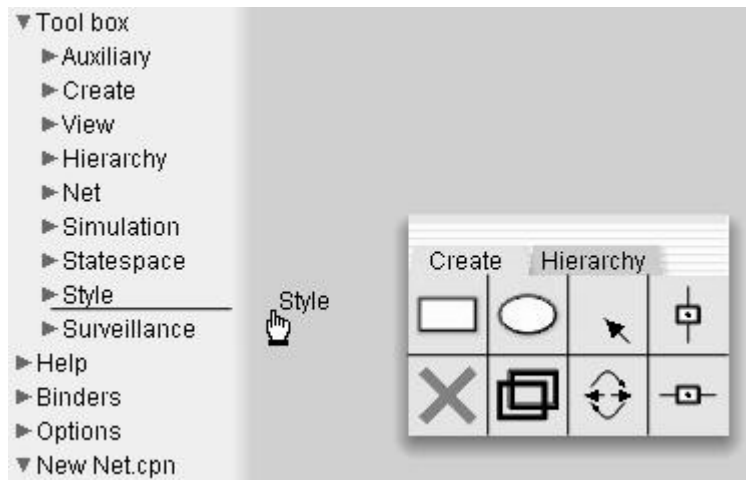
5.2.3 Οι Παλέτες

Οι παλέτες (palettes) είναι μια συλλογή εργαλείων από την οποία μπορούμε να διαλέξουμε ένα εργαλείο για να το εφαρμόσουμε σε κάποιο σημείο της επιφάνειας εργασίας. Όλες οι παλέτες είναι προσβάσιμες από το ευρετήριο κάτω από την καταχώρηση Tool box, όπως φαίνεται στην Εικόνα 3.



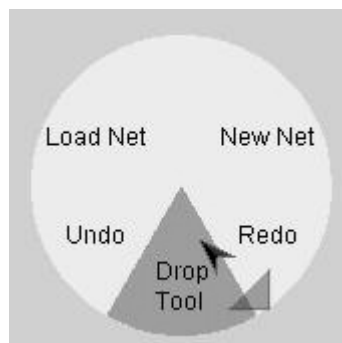
Εικόνα 3: Οι παλέτες εργαλείων στο ευρετήριο

Μπορούμε να τοποθετήσουμε πολλές παλέτες εργαλείων στο ίδιο binder, για να έχουμε όλα τα εργαλεία μαζί και για να εξοικονομήσουμε επιφάνεια οθόνης. Δεν μπορούμε όμως να τοποθετήσουμε παλέτες εργαλείων στα ίδια binders με τις σελίδες του δικτύου ή τις δηλώσεις. Στην Εικόνα 4 έχουμε τοποθετήσει στο ίδιο binder δύο παλέτες εργαλείων, ενώ σύρουμε στην επιφάνεια εργασίας μια τρίτη.



Εικόνα 4: Δύο παλέτες στο ίδιο binder

Όταν διαλέξουμε ένα εργαλείο από μια παλέτα, το εικονίδιο του εργαλείου φωτίζεται και ο κέρσορας αλλάζει, ώστε να δείχνει ότι έχουμε ένα εργαλείο στο χέρι. Για να αφήσουμε το εργαλείο, είτε κάνουμε αριστερό κλικ στο εικονίδιο του εργαλείου στην παλέτα από όπου το πήραμε, είτε επιλέγουμε την εντολή Drop Tool από το μενού μαρκαρισμάτων της Επιφάνειας Εργασίας, που φαίνεται στην Εικόνα 5. Για τα μενού μαρκαρισμάτων θα μιλήσουμε στην επόμενη παράγραφο.



Εικόνα 5: Το μενού μαρκαρισμάτων της Επιφάνειας Εργασίας

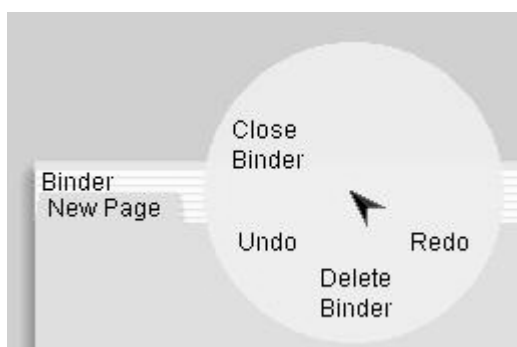
Τα εργαλεία που είναι διαθέσιμα σε παλέτες είναι τα :

- **Βοηθητικά εργαλεία (Auxiliary tools)**
- **εργαλεία Δημιουργίας (Create tools)**
- **εργαλεία Παρουσιάσης (View tools)**
- **εργαλεία Ιεραρχίας (Hierarchy tools)**
- **εργαλεία Δικτύου (Net tools)**
- **εργαλεία Προσομοίωσης (Simulation tools)**
- **εργαλεία Χώρου Καταστάσεων (State space tools)**
- **εργαλεία Στυλ (Style tools)**

Η πρακτική χρήση κάθε κατηγορίας εργαλείων θα περιγραφεί στη συνέχεια του κεφαλαίου, μέσα από παραδείγματα.

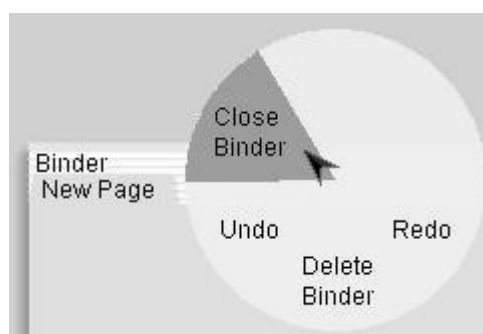
5.2.4 Τα Μενού Μαρκαρισμάτων

Τα μενού μαρκαρισμάτων εμφανίζονται, αν κρατήσουμε για μερικά δευτερόλεπτα πατημένο, το δεξί πλήκτρο του ποντικιού πάνω σε οποιοδήποτε στοιχείο στην οθόνη. Δηλαδή το περιεχόμενο των μενού μαρκαρισμάτων εξαρτάται από το σημείο της οθόνης που βρίσκεται ο κέρσορας. Για παράδειγμα, όπως φαίνεται στην Εικόνα 6, κρατώντας πατημένο το δεξί πλήκτρο του ποντικιού πάνω στο τίτλο ενός binder, εμφανίζουμε το **μενού μαρκαρισμάτων Binder (Binder marking menu)**.



Εικόνα 6: Το μενού μαρκαρισμάτων Binder

Τα μενού μαρκαρισμάτων είναι κυκλικά και οι διάφορες εντολές είναι τοποθετημένες όπως τα κομμάτια της πίτας. Κρατώντας πατημένο το δεξί πλήκτρο και μετακινώντας το πάνω στο μενού, βλέπουμε ότι αλλάζουν χρώμα οι διάφορες εντολές. Για να επιλέξουμε μια εντολή, μετακινούμε τον κέρσορα πάνω από το “κομμάτι” της εντολής ώστε να αλλάξει χρώμα και αφήνουμε το πλήκτρο του ποντικιού. Για να εξαφανίσουμε το μενού, μετακινούμε τον κέρσορα στο κέντρο ή εκτός του μενού, ώστε καμία εντολή να μην έχει διαφορετικό χρώμα, και αφήνουμε το πλήκτρο του ποντικιού. Στην εικόνα 7, επιλέγουμε την εντολή Close Binder από το μενού μαρκαρισμάτων Binder για να κλείσουμε ένα binder.



Εικόνα 7: Επιλογή της εντολής Close Binder

Τα μενού μαρκαρισμάτων έχουν σχεδιαστεί για να βοηθήσουν το χρήστη του προγράμματος να επιλέγει εντολές πολύ γρήγορα. Ένα πλεονέκτημα των κυκλικών μενού είναι ότι το χέρι εξοικειώνεται ευκολότερα με τις κατευθύνσεις παρά με τις αποστάσεις. Αυτό σημαίνει ότι το να σύρουμε το ποντίκι σε μια συγκεκριμένη κατεύθυνση, για να επιλέξουμε μια εντολή από ένα μενού μαρκαρισμάτων είναι πολύ πιο γρήγορο από το να μετακινηθούμε μια συγκεκριμένη απόσταση σε μια λίστα σε

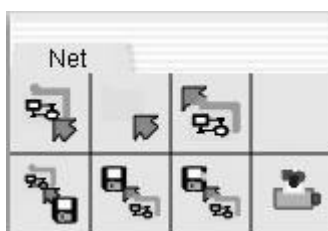
ένα παραδοσιακό, γραμμικό μενού. Υπάρχουν πάρα πολλά μενού μαρκαρισμάτων, τα οποία θα δούμε στη συνέχεια του κεφαλαίου μέσα από τη χρήση τους σε πρακτικά παραδείγματα.

5.3 Σχεδιασμός ενός Χρωματισμένου Δικτύου Petri

5.3.1 Δημιουργία ενός Νέου Δικτύου

Ένας τρόπος για να δημιουργήσουμε ένα νέο δίκτυο είναι να σύρουμε την παλέτα εργαλείων Δικτύου, που φαίνεται στην εικόνα 8, στην επιφάνεια εργασίας και να

κάνουμε αριστερό κλικ στο εργαλείο



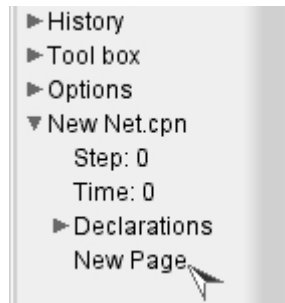
Εικόνα 8: Η παλέτα εργαλείων Δικτύου

Αλλιώς μπορούμε να χρησιμοποιήσουμε την αντίστοιχη εντολή New Net του **μενού μαρκαρισμάτων της Επιφάνειας Εργασίας (Workspace marking menu)**, το οποίο εμφανίζεται, αν κρατήσουμε πατημένο το δεξί πλήκτρο του ποντικιού για μερικά δευτερόλεπτα, πάνω στην επιφάνεια εργασίας. Στην Εικόνα 9 φαίνεται το μενού μαρκαρισμάτων της Επιφάνειας Εργασίας.



Εικόνα 9: Το μενού μαρκαρισμάτων της Επιφάνειας Εργασίας

Μετά τη δημιουργία του νέου δικτύου, μια καταχώρηση “New Net” εμφανίζεται στο ευρετήριο, όπως φαίνεται στην Εικόνα 10. Μπορούμε να έχουμε ανοιχτά περισσότερα από ένα δίκτυα ταυτόχρονα.



Εικόνα 10: Ένα νέο δίκτυο “New Net”

Για να αλλάξουμε το όνομα μιας σελίδας κάνουμε κλικ στο όνομα της σελίδας στο ευρετήριο και εισάγουμε το κείμενο.

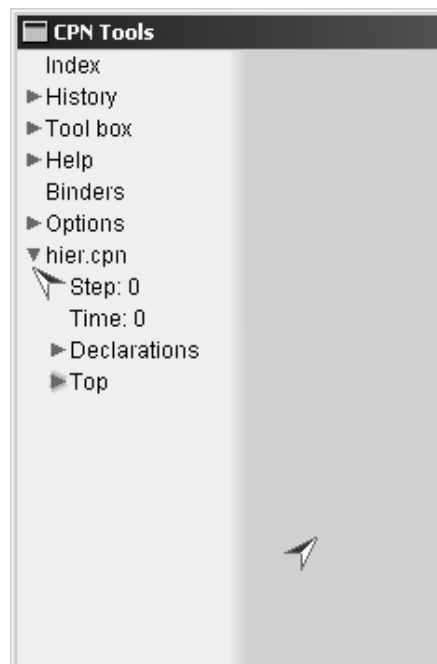
5.3.2 Άνοιγμα ενός Δικτύου

Για να ανοίξουμε ένα αποθηκευμένο δίκτυο, σύρουμε την παλέτα εργαλείων Δικτύου



(Εικόνα 8) στην επιφάνεια εργασίας και κάνουμε αριστερό κλικ στο εργαλείο . Εναλλακτικά μπορούμε να χρησιμοποιήσουμε την αντίστοιχη εντολή Load Net του μενού μαρκαρισμάτων της Επιφάνειας Εργασίας (Εικόνα 9).

Και οι δύο επιλογές θα εμφανίσουν ένα τυπικό παράθυρο διαλόγου “Open file”, από το οποίο μπορούμε να επιλέξουμε το αρχείο που θέλουμε να ανοίξουμε. Στη συνέχεια, το δίκτυο θα εμφανιστεί στο ευρετήριο ως μια καταχώρηση με το όνομα του αντίστοιχου αρχείου, όπως φαίνεται στην Εικόνα 11.



Εικόνα 11: Άνοιγμα του δικτύου “hier.cpn”

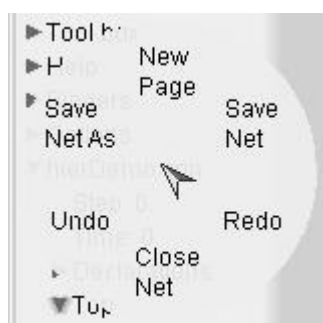
5.3.3 Αποθήκευση ενός Δικτύου

Για να αποθηκεύσουμε ένα δίκτυο, σύρουμε την παλέτα εργαλείων Δικτύου (Εικόνα



8) στην επιφάνεια εργασίας και κάνουμε αριστερό κλικ στο εργαλείο . Τότε το δίκτυο αποθηκεύεται με το τρέχον όνομα του, δηλαδή αυτό που εμφανίζεται στο ευρετήριο.

Μπορούμε επίσης να χρησιμοποιήσουμε το **μενού μαρκαρισμάτων Δικτύου (Net marking menu)**, από το οποίο επιλέγουμε την εντολή Save Net για να αποθηκεύσουμε το δίκτυο με το τρέχον όνομα του. Στην Εικόνα 12 φαίνεται το μενού μαρκαρισμάτων Δικτύου.



Εικόνα 12: Το μενού μαρκαρισμάτων Δικτύου

Μπορούμε να αποθηκεύσουμε το δίκτυο με διαφορετικό όνομα, αλλάζοντας απλά το όνομα του δικτύου στο ευρετήριο. Το δίκτυο θα αποθηκευτεί εκεί που αποθηκεύτηκε τελευταία φορά. Ο προεπιλεγμένος κατάλογος αποθήκευσης των δικτύων είναι ο :

C://Documents and Settings/<username>/Local Settings/Temp.

Μέσω του ευρετηρίου δεν μπορούμε να αλλάξουμε το μέρος που αποθηκεύεται το



Μπορούμε επίσης να χρησιμοποιήσουμε το εργαλείο από την παλέτα Εργαλείων Δικτύου. Θα εμφανιστεί ένα παράθυρο διαλόγου από το οποίο μπορούμε να επιλέξουμε το μέρος που θα αποθηκεύσουμε το δίκτυο και/ή να αλλάξουμε το όνομα του δικτύου.

Μια τρίτη επιλογή είναι να χρησιμοποιήσουμε το μενού μαρκαρισμάτων Δικτύου. Επιλέγοντας την εντολή Save Net As, θα εμφανιστεί, όπως και πριν, ένα παράθυρο διαλόγου από το οποίο μπορούμε να επιλέξουμε το μέρος που θα αποθηκεύσουμε το δίκτυο και/ή να αλλάξουμε το όνομα του δικτύου.

Μπορούμε να δούμε που είναι αποθηκεύμενο το δίκτυο, κρατώντας το ποντίκι (χωρίς να κάνουμε κλικ) για μερικά δευτερόλεπτα πάνω από το όνομα του δικτύου στο ευρετήριο, όπως φαίνεται στην Εικόνα 13.



Εικόνα 13: Εμφάνιση καταλόγου αποθήκευσης του δικτύου

Όλα τα δίκτυα που είναι ανοιχτά, σώζονται περιοδικά σε αρχεία backup. Τα αρχεία αποθηκεύονται αφού έχει εκτελεστεί ένας συγκεκριμένος αριθμός από εντολές. Ο αριθμός των εντολών μεταξύ των αυτόματων αποθηκεύσεων είναι μία από τις επιλογές που μπορεί να αλλάξει, μέσω της επιλογής Options που υπάρχει στο ευρετήριο. Αν το όνομα του δικτύου είναι <όνομα_αρχείου>, τότε το αντίστοιχο αρχείο backup είναι το #<όνομα_αρχείου> και αποθηκεύεται στον ίδιο κατάλογο με το αρχείο του δικτύου. Αν το δίκτυο έχει κάποιες αλλαγές που δεν έχουν αποθηκευτεί, τότε το δίκτυο θα αποθηκευτεί αυτόματα όταν επιλέξουμε την εντολή Close Net από την παλέτα εργαλείων Δικτύου ή όταν κλείσουμε το πρόγραμμα CPN Tools.

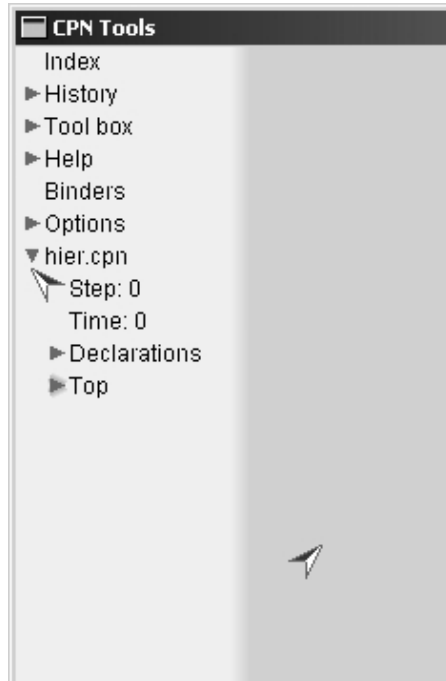
5.3.4 Εικόνα ενός Δικτύου

Το ευρετήριο περιέχει μία καταχώρηση για κάθε δίκτυο που είναι ανοιχτό στο πρόγραμμα. Αν το δίκτυο ήταν ήδη αποθηκευμένο και φορτώθηκε από κάποιο αρχείο, τότε η καταχώρηση του δικτύου στο ευρετήριο έχει ως επιγραφή το όνομα του δικτύου. Στο παρακάτω σχήμα, έχουμε ανοίξει το δίκτυο με όνομα “hier.cpn”. Αν έχει δημιουργηθεί ένα νέο δίκτυο, τότε η καταχώρηση του δικτύου έχει ως επιγραφή το όνομα “New net.cpn”.

Για να αρχίσουμε να δουλεύουμε σε μια σελίδα του δικτύου, πρέπει να ανοίξουμε την καταχώρηση του δικτύου στο ευρετήριο, κάνοντας αριστερό κλικ στο τρίγωνο που βρίσκεται δίπλα στο όνομα του δικτύου.

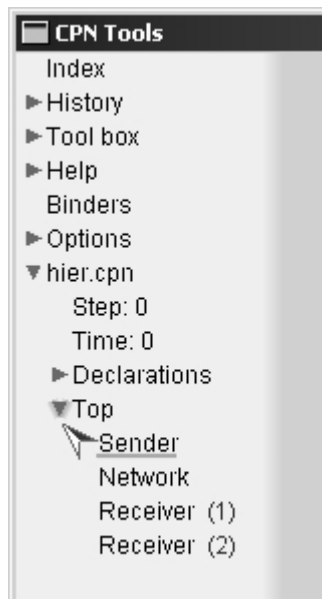
Όπως φαίνεται και στην Εικόνα 14, μία καταχώρηση δικτύου περιέχει τα εξής στοιχεία :

- **Step** : ο αριθμός των βημάτων που έχουν εκτελεστεί στην προσομοίωση
- **Time** : ο τρέχων χρόνος μοντελοποίησης
- **Declarations** : οι δηλώσεις των **συνόλων χρωμάτων (colour sets)**, των συναρτήσεων, των σταθερών τιμών κτλ. Όλες οι δηλώσεις είναι γραμμένες στη γλώσσα CPN ML.
- Καταχωρήσεις σελίδων για τις σελίδες του δικτύου.



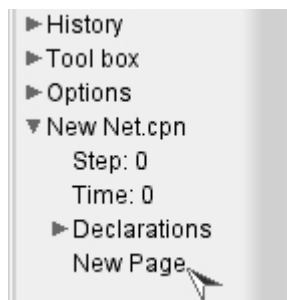
Εικόνα 14: Καταχώρηση δικτύου

Όλες οι σελίδες ενός δικτύου είναι προσβάσιμες μέσω του ευρετηρίου. Η ιεραρχική δομή ενός δικτύου αντικατοπτρίζεται στο ευρετήριο. Η καταχώρηση για μια **υποσελίδα (subpage)** εμφανίζεται κάτω από την καταχώρηση της **υπερσελίδας (superpage)**. Οι καταχωρήσεις των υποσελίδων είναι ορατές στο ευρετήριο, μόνο όταν η αντίστοιχη καταχώρηση της υπερ-σελίδας είναι ανοιχτή. Ένα παράδειγμα ιεραρχικής δομής φαίνεται στην Εικόνα 15. Ανοίγοντας την καταχώρηση για τη σελίδα Top, βλέπουμε ότι έχει τέσσερις υποσελίδες : ένα **στιγμιότυπο (page instance)** των σελίδων Sender και Network, και δύο στιγμιότυπα της σελίδας Receiver.



Εικόνα 15: Ιεραρχική δομή

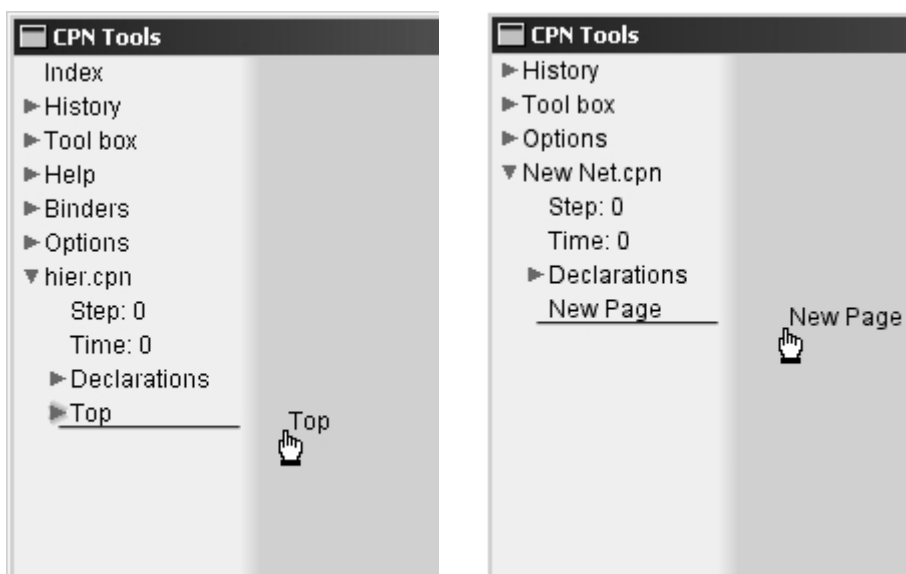
Αν έχουμε δημιουργήσει ένα νέο δίκτυο αντί να ανοίξουμε ένα ήδη υπάρχον, θα υπάρχει μόνο μία σελίδα με το όνομα New Page, όπως φαίνεται και στην Εικόνα 16. Η σελίδα αυτή δημιουργείται αυτόματα με το νέο δίκτυο και είναι κενή, ώστε να μπορούμε να δημιουργήσουμε θέσεις, μεταβάσεις κτλ.



Εικόνα 16: Η σελίδα New Page

5.3.5 Ανοιγμα μιας Σελίδας

Για να ανοίξουμε μια σελίδα στο δίκτυο, μετακινούμε τον κέρσορα πάνω από το όνομα της σελίδας, πιέζουμε το αριστερό πλήκτρο του ποντικιού και κρατώντας το πατημένο, το σύρουμε στην επιφάνεια εργασίας. Στην Εικόνα 17 (α) σύρουμε τη σελίδα με όνομα Top στην επιφάνεια εργασίας, ενώ στην Εικόνα 17 (β) έχουμε ένα νέο δίκτυο και σύρουμε τη σελίδα New Page στην επιφάνεια εργασίας, ώστε να αρχίσουμε να δημιουργούμε ένα νέο δίκτυο.

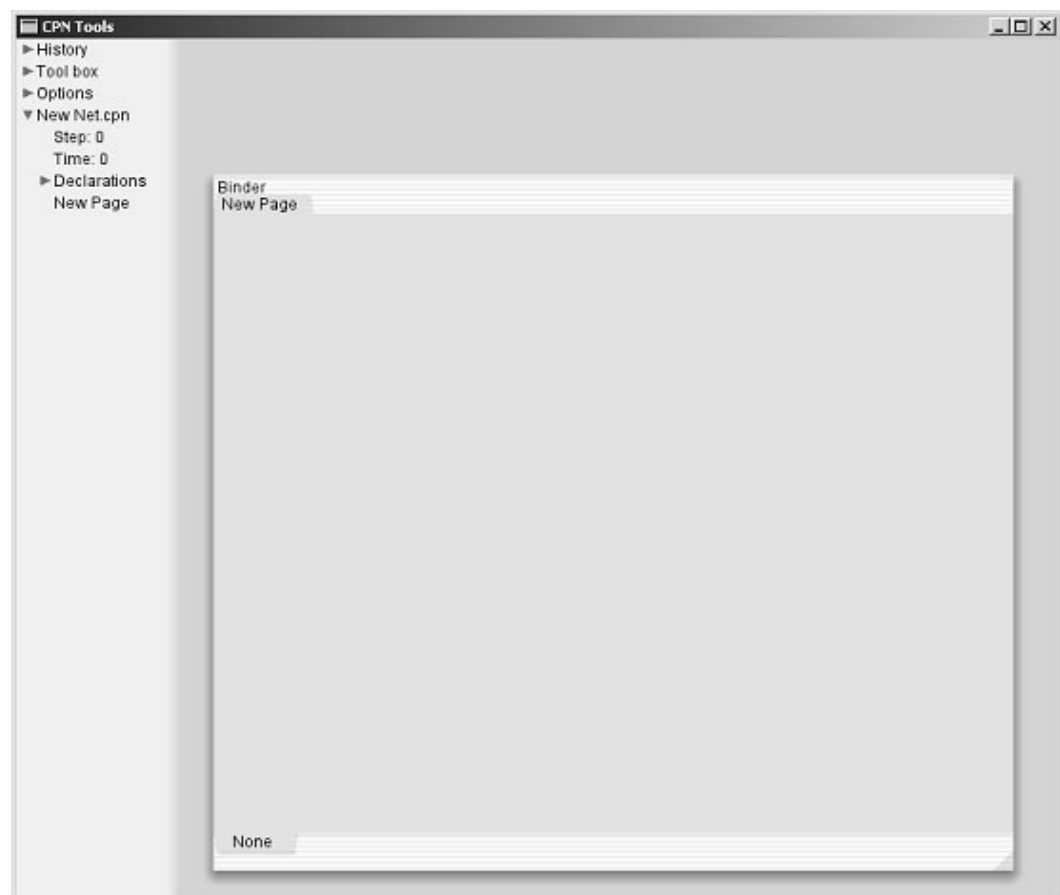
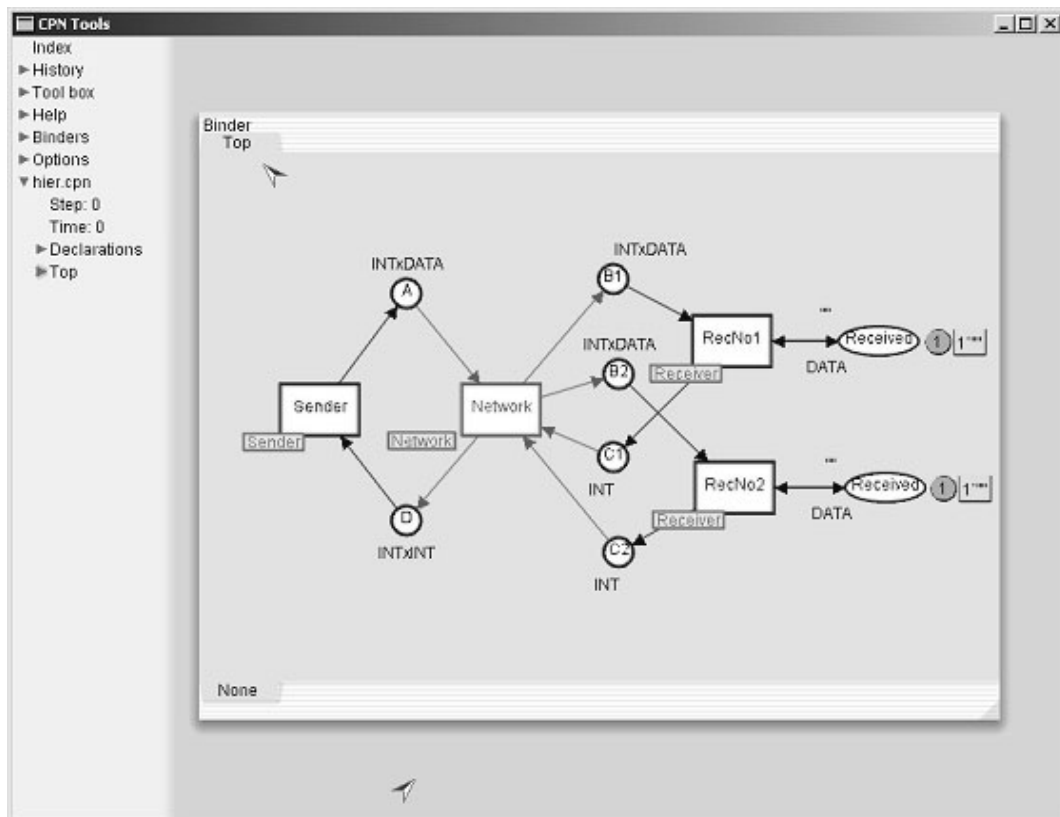


(α)

(β)

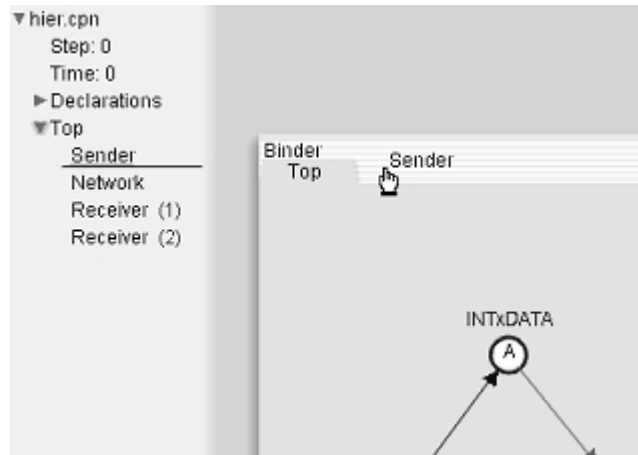
Εικόνα 17: (α)Top (β)New Page

Όταν αφήσουμε το πλήκτρο του ποντικιού, εμφανίζεται η σελίδα στην επιφάνεια εργασίας, σε ένα Binder. Στο πάνω μέρος της Εικόνας 18 φαίνεται η σελίδα Top, ενώ στο κάτω μέρος φαίνεται η σελίδα New Page.



Εικόνα 18: Οι σελίδες Top και New Page

Μπορούμε να σύρουμε τις σελίδες και να τις αφήσουμε στην επιφάνεια εργασίας σε ένα ή περισσότερα binders. Για να αφήσουμε μια σελίδα σε ένα binder, αφήνουμε το πλήκτρο του ποντικιού, όταν το όνομα της σελίδας είναι μέσα στην ορατή περιοχή που βρίσκεται στην κορυφή του binder, όπως φαίνεται στην Εικόνα 19.



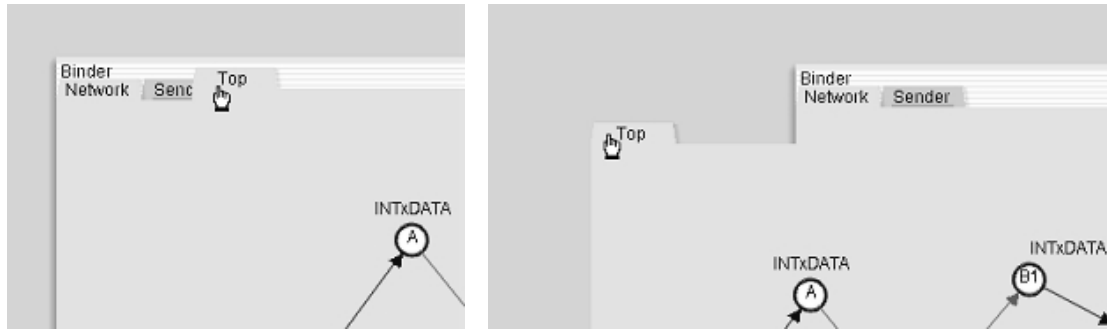
Εικόνα 19: Υπάρχον binder

Για να αφήσουμε τη σελίδα σε ένα νέο binder, απλά αφήνουμε το όνομα της σελίδας έξω από την περιοχή του binder, όπως φαίνεται στην Εικόνα 20.



Εικόνα 20: Νέο binder

Μπορούμε να τοποθετήσουμε πολλές σελίδες στο ίδιο binder ή να έχουμε την ίδια σελίδα σε περισσότερα από ένα binders. Για να μετακινήσουμε μια σελίδα από ένα binder σε κάποιο άλλο, σύρουμε το όνομα της σελίδας που βρίσκεται στην πάνω αριστερή γωνία της σελίδας, σε ένα άλλο υπάρχον binder. Μπορούμε επίσης να τοποθετήσουμε τη σελίδα σε ένα νέο binder, σύροντας όπως πριν το όνομα της σελίδας και αφήνοντας το στην επιφάνεια εργασίας. Στην Εικόνα 21 φαίνεται η μετακίνηση της σελίδας Top από το προηγούμενο binder σε ένα νέο.

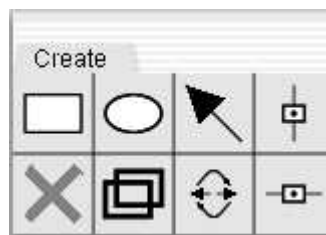


Εικόνα 21: Μετακίνηση της σελίδας Top



5.3.6 Δημιουργία Θέσεων, Μεταβάσεων και Τόξων


Αφού έχουμε σύρει μια κενή σελίδα “New Page” από το ευρετήριο μπορούμε να δημιουργήσουμε θέσεις, μεταβάσεις και τόξα με δύο τρόπους.

Ο πρώτος τρόπος είναι να χρησιμοποιήσουμε την παλέτα εργαλείων Δημιουργίας που φαίνεται στην Εικόνα 22.



Εικόνα 22: Η παλέτα εργαλείων Δημιουργίας

Ανάλογα με το τι στοιχείο θέλουμε να δημιουργήσουμε, επιλέγουμε και το αντίστοιχο εργαλείο. Για μετάβαση επιλέγουμε το εργαλείο  , για θέση το εργαλείο 

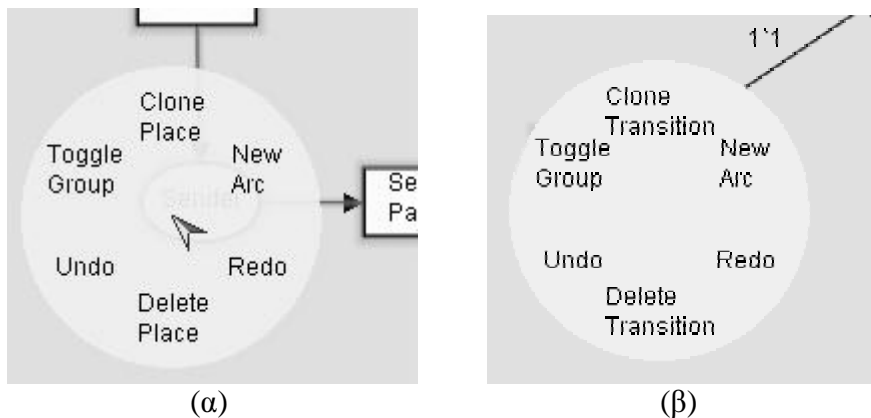
και για τόξο το εργαλείο  και κάνουμε αριστερό κλικ στο σημείο της κενής σελίδας που θέλουμε να δημιουργηθεί το στοιχείο. Αν θέλουμε να ακυρώσουμε την επιλογή ενός στοιχείου, κάνουμε πάλι αριστερό κλικ στο αντίστοιχο εργαλείο.

Ο δεύτερος τρόπος είναι να χρησιμοποιήσουμε το **μενού μαρκαρισμάτων Σελίδας, (Page marking menu)** που φαίνεται στην Εικόνα 23. Για να δημιουργήσουμε μια θέση ή μια μετάβαση, εμφανίζουμε το μενού μαρκαρισμάτων Σελίδας, πατώντας το δεξί πλήκτρο του ποντικιού για μερικά δευτερόλεπτα πάνω στη σελίδα. Επιλέγουμε την εντολή New Place για τη δημιουργία μιας θέσης και την εντολή New Transition για τη δημιουργία μιας μετάβασης.



Εικόνα 23: Μενού μαρκαρισμάτων Σελίδας

Για να δημιουργήσουμε ένα τόξο, πατάμε για μερικά δευτερόλεπτα το δεξί πλήκτρο του ποντικιού, πάνω στη θέση ή στη μετάβαση από την οποία θέλουμε να ξεκινάει το τόξο και εμφανίζεται αντίστοιχα το **μενού μαρκαρισμάτων Θέσης (Place marking menu)** ή το **μενού μαρκαρισμάτων Μετάβασης (Transition marking menu)**. Στην Εικόνα 24(α) φαίνεται το μενού μαρκαρισμάτων Θέσης, ενώ στην Εικόνα 24(β) φαίνεται το μενού μαρκαρισμάτων Μετάβασης



Εικόνα 24: (α) Μενού μαρκαρισμάτων Θέσης (β) Μενού μαρκαρισμάτων Μετάβασης

Στη συνέχεια επιλέγουμε την εντολή New Arc. Ένα τόξο επισυνάπτεται στον κέρσορα και μπορούμε να το σύρουμε στη μετάβαση ή στη θέση στην οποία θέλουμε να καταλήγει το τόξο. Ένα τόξο πρέπει πάντα να πηγαίνει από μια θέση σε μια μετάβαση ή από μια μετάβαση σε μια θέση. Μπορείτε να αλλάξετε τη μορφή ενός τόξου, κάνοντας κλικ πάνω του και σύροντας το ώστε να λυγίσει.

5.3.7 Σύνταξη Δηλώσεων

Οι δηλώσεις είναι τοποθετημένες στο ευρετήριο κάτω από την καταχώρηση Declarations που βρίσκεται κάτω από το όνομα του δικτύου. Κάνοντας αριστερό κλικ στο τρίγωνο δίπλα από την καταχώρηση Declarations, εμφανίζονται οι δηλώσεις και κάνοντας αριστερό κλικ στο τρίγωνο δίπλα από μια δήλωση, εμφανίζει τη δήλωση αυτή. Ένα παράδειγμα δηλώσεων φαίνεται στην Εικόνα 25.

```

▼ Declarations
  ▼ color INT
    color INT = int;
  ► color DATA
  ▼ color INTxDATA
    color INTxDATA = product INT * DATA;
  ► color INTxINT
  ► var n k n1 n2
  ▼ var p str
    var p, str: DATA;
  ► val stop = "#####";
  ► color Ten0
  ► color Ten1
  ► var s
  ► var r r1 r2
  ▼ fun Ok(s:Ten0,r:Ten1) = (r<=s);
    fun Ok(s:Ten0,r:Ten1) = (r<=s);
  ► fun imin(i:int,j:int)=if (i<j) then i else j;

```

Εικόνα 25: Παράδειγμα Δηλώσεων

Για να προσθέσουμε μια δήλωση, εμφανίζουμε το **μενού μαρκαρισμάτων Δηλώσεων, (Declarations marking menu)** πατώντας για μερικά δευτερόλεπτα το δεξί πλήκτρο του ποντικιού, είτε πάνω στην καταχώρηση Declarations (Εικόνα 26(α)), είτε πάνω σε μία από τις δηλώσεις στο ευρετήριο (Εικόνα 26(β)), και επιλέγουμε την εντολή Append Decl.



Εικόνα 26: Μενού μαρκαρισμάτων Δηλώσεων

Η δήλωση που προστέθηκε θα εμφανιστεί είτε στη βάση της λίστας δηλώσεων ή κάτω από τη δήλωση που την προσθέσαμε. Το κοκκινωπό χρώμα υποδεικνύει ότι μπορούμε να συντάξουμε απ'ευθείας τη δήλωση, όπως φαίνεται στην Εικόνα 27.

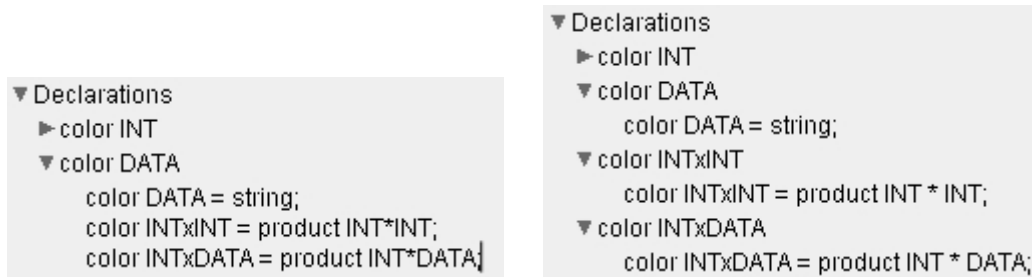
```

► color Ten1
► var s
► var r r1 r2
► fun Ok(s:Ten0,r:Ten1) = (r<=s);
▼ New decl
  (* Empty Decl *)

```

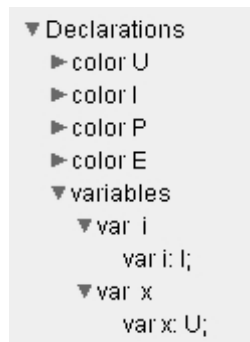
Εικόνα 27: Σύνταξη νέας δήλωσης

Μπορούμε να εισάγουμε νέες γραμμές πατώντας Enter και μπορούμε να δημιουργήσουμε μια νέα δήλωση κάτω από αυτήν που συντάσσουμε, πατώντας Ctrl-Enter. Μπορούμε επίσης να προσθέσουμε μια δήλωση σε μια ήδη υπάρχουσα, αν τις χωρίσουμε με ερωτηματικό ;. Όταν τελειώσουμε τη σύνταξη της δήλωσης, οι δηλώσεις θα διαχωριστούν, όπως φαίνεται στην Εικόνα 28.



Εικόνα 28: Διαχωρισμός δηλώσεων

Μπορούμε επίσης να προσθέσουμε **μπλόκα δηλώσεων (block declarations)**. Τα μπλόκα δηλώσεων χρησιμοποιούνται για να χωρίσουν τις δηλώσεις σε ομάδες, συνήθως σε μεγάλα δίκτυα που έχουν πολλές δηλώσεις. Στην Εικόνα 29 οι μεταβλητές τοποθετούνται μαζί σε ένα μπλοκ με το όνομα variables.



Εικόνα 29: Το μπλοκ δηλώσεων variables

Για να προσθέσουμε ένα μπλοκ δηλώσεων, εμφανίζουμε το μενού μαρκαρισμάτων Δηλώσεων όπως πριν και επιλέγουμε την εντολή Append Block. Το νέο μπλοκ θα εμφανιστεί στη βάση της λίστας ή κάτω από τη δήλωση που το προσθέσαμε, όπως φαίνεται στην Εικόνα 30. Πάλι το κοκκινωπό χρώμα υποδεικνύει ότι μπορούμε να συντάξουμε απ'ευθείας το μπλοκ.



Εικόνα 30: Μπλοκ δηλώσεων έτοιμο για σύνταξη

Για να προσθέσουμε μια δήλωση στο μπλοκ δηλώσεων, εμφανίζουμε το μενού μαρκαρισμάτων Δηλώσεων όπως πριν και επιλέγουμε την εντολή Append Declaration. Επειδή δεν είναι δυνατόν να μετακινούμε τις δηλώσεις μέσα στο ευρετήριο, πρέπει πρώτα να δημιουργήσουμε τα μπλοκ δηλώσεων και στη συνέχεια να προσθέσουμε δηλώσεις σε αυτά.

5.3.8 Εισαγωγή Επιγραφών

Για να προσθέσουμε μια **επιγραφή (inscription)**, κάνουμε αριστερό κλικ στη θέση, στη μετάβαση ή στο τόξο που θέλουμε να προσθέσουμε την επιγραφή και εισάγουμε το κείμενο. Αν έχουμε κάποιο εργαλείο στο χέρι μας, κρατάμε λίγο παραπάνω πατημένο το αριστερό πλήκτρο, ώστε να εμφανιστεί ο κέρσορας εισαγωγής κειμένου. Ο κέρσορας εισαγωγής κειμένου εμφανίζεται αυτόματα μετά από τη δημιουργία μιας θέσης, μιας μετάβασης ή ενός τόξου. Στην Εικόνα 31 φαίνεται η εισαγωγή της επιγραφής Sender σε μια θέση.




Εικόνα 31: Εισαγωγή επιγραφών σε μια θέση

Όταν τελειώσουμε με την εισαγωγή μιας επιγραφής, μπορούμε να χρησιμοποιήσουμε το πλήκτρο TAB για να συνεχίσουμε με την επόμενη επιγραφή. Η σειρά των επιγραφών όταν χρησιμοποιούμε το πλήκτρο TAB είναι για τις θέσεις : όνομα, τύπος/σύνολο χρωμάτων, αρχικό μαρκάρισμα και για τις μεταβάσεις : όνομα, **συνάρτηση φύλακας (guard function)**, χρονική καθυστέρηση, **τμήμα κώδικα (code segment)**. Τα τόξα έχουν μόνο μία επιγραφή. Στην Εικόνα μετά την εισαγωγή του ονόματος της θέσης, Sender, συνεχίζουμε με τον τύπο της θέσης TYPE (όπως υποδεικνύεται από το κοκκινωπό χρώμα).

5.4 Ιεραρχικά Δίκτυα

Η δημιουργία μεγάλων και πολύπλοκων δικτύων είναι μια δύσκολη εργασία. Όπως και στο σπονδυλωτό προγραμματισμό, η κατασκευή των Χρωματισμένων δικτύων Petri μπορεί να σπάσει σε μικρότερα κομμάτια, χρησιμοποιώντας τις δυνατότητες που παρέχει το πρόγραμμα CPN Tools για τη δημιουργία **μεταβάσεων αντικατάστασης (substitution transitions)** και **θέσεων συγχώνευσης (fusion places)**. Τα δίκτυα με μεταβάσεις αντικατάστασης είναι δίκτυα με πολλαπλά επίπεδα λεπτομέρειας – μπορούμε να έχουμε ένα απλοποιημένο δίκτυο που να δίνει μια γενική εικόνα του συστήματος που μοντελοποιούμε και αντικαθιστώντας τις μεταβάσεις αυτής της σελίδας με περισσότερο λεπτομερείς σελίδες, μπορούμε να φέρνουμε όλο και περισσότερη λεπτομέρεια στο δίκτυο.

5.4.1 Ετικέτες Ιεραρχίας

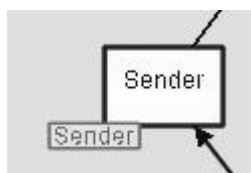
Οι **ετικέτες ιεραρχίας (Hierarchy Tags)** είναι μικρές μπλε ετικέτες, όπως για παράδειγμα η , οι οποίες συσχετίζονται με κάποιες θέσεις και μεταβάσεις στα Ιεραρχικά Χρωματισμένα δίκτυα Petri. Υπάρχουν τρία διαφορετικά είδη ετικετών ιεραρχίας :

- **Ετικέτες υποσελίδας (Subpage tags)**
- **Ετικέτες τύπου πύλης (Port-type tags)**
- **Ετικέτες συνόλου συγχώνευσης (Fusion-set tags)**

Οι ιεραρχικές δομές μπορούν να διαγράφονται ή να τροποποιούνται με τη διαγραφή των ετικετών ιεραρχίας.

5.4.1.1 Ετικέτες Υποσελίδας

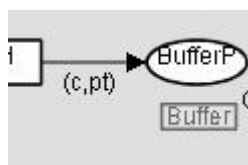
Οι ετικέτες υποσελίδας συχετίζονται με τις μεταβάσεις αντικατάστασης. Το κείμενο μιας ετικέτας υποσελίδας είναι το ίδιο με το όνομα της αντίστοιχης υποσελίδας και τις περισσότερες φορές είναι ίδιο και με το όνομα της αντίστοιχης μετάβασης αντικατάστασης. Στην Εικόνα 32 φαίνεται η μετάβαση αντικατάστασης Sender με την ομώνυμη ετικέτα υποσελίδας.



Εικόνα 32: Η μετάβαση αντικατάστασης Sender

5.4.1.2 Ετικέτες Συνόλου Συγχώνευσης

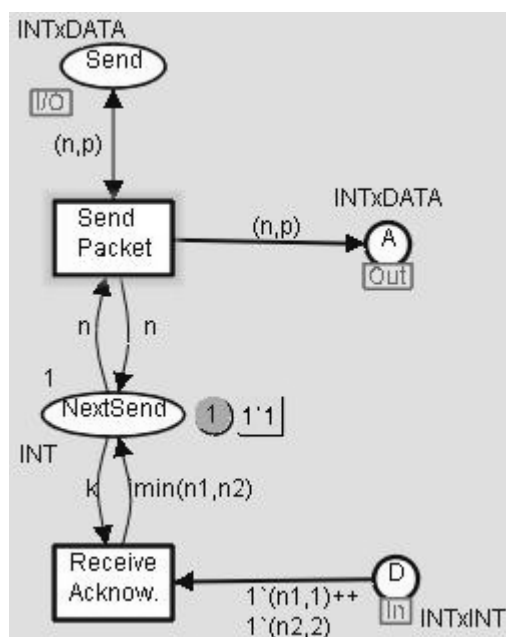
Οι ετικέτες συνόλου συγχώνευσης συσχετίζονται με τις θέσεις συγχώνευσης. Το κείμενο σε μια ετικέτα συνόλου συγχώνευσης υποδεικνύει το όνομα του συνόλου συγχώνευσης στο οποίο ανήκει η αντίστοιχη θέση συγχώνευσης. Στην Εικόνα 33 φαίνεται η θέση συγχώνευσης Buffer με την ομώνυμη ετικέτα.



Εικόνα 33: Η θέση συγχώνευσης Buffer

5.4.1.3 Ετικέτες Τύπου Πύλης

Οι ετικέτες τύπου πύλης συσχετίζονται με τις θέσεις πύλης στις υποσελίδες. Το κείμενο σε μια ετικέτα τύπου πύλης υποδεικνύει αν η θέση πύλης είναι μια πύλη εισόδου (In), μια πύλη εξόδου (Out) ή μια πύλη εισόδου/εξόδου (I/O). Στην Εικόνα φαίνονται τρεις θέσεις πύλης με τις αντίστοιχες ετικέτες : η πύλη εισόδου D, η πύλη εξόδου A και η πύλη εισόδου/εξόδου Send. Η Εικόνα 34 παρουσιάζει τον αποστολέα Sender του απλού πρωτοκόλλου επικοινωνίας που περιγράψαμε αναλυτικά στο προηγούμενο κεφάλαιο.



Εικόνα 34: Θέσεις πύλης

5.4.2 Δημιουργία Μεταβάσεων Αντικατάστασης


Υπάρχουν δύο τρόποι δημιουργίας ιεραρχικών δικτύων :

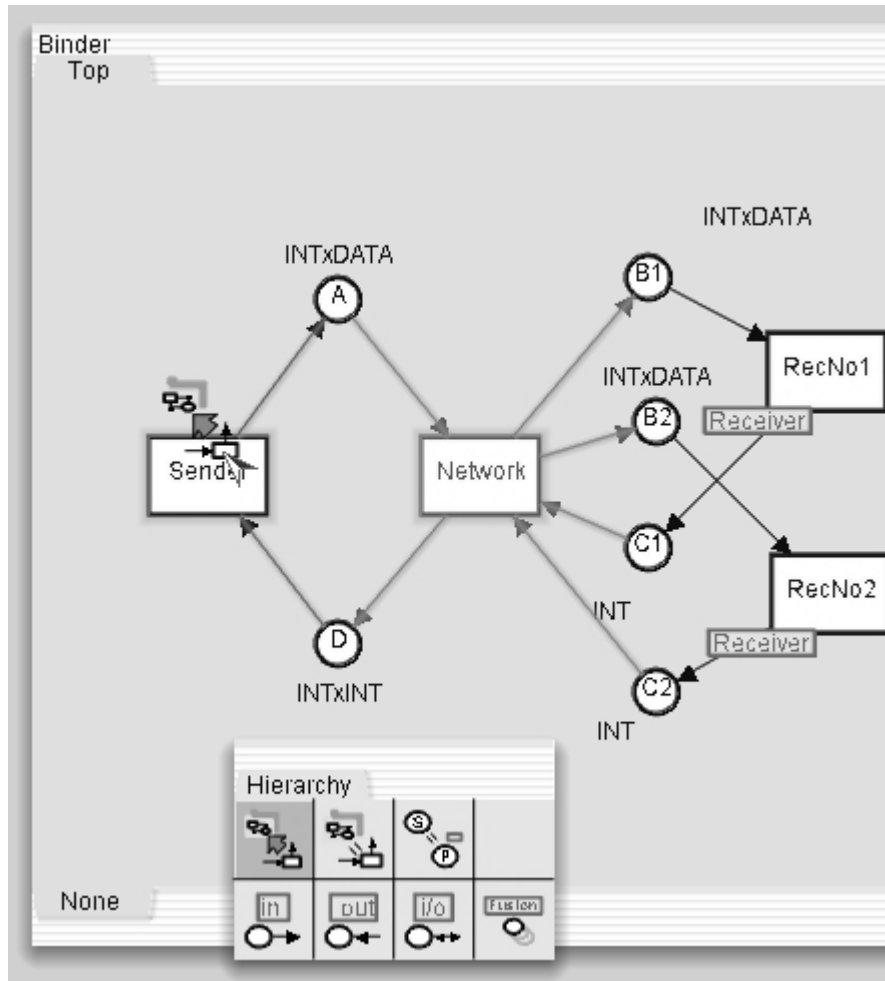
- η από πάνω-προς-τα-κάτω ανάπτυξη (top-down development)
- η από κάτω-προς-τα-πάνω ανάπτυξη (bottom-up development)

5.4.2.1 Από Πάνω-προς-τα-Κάτω Ανάπτυξη

Όταν δημιουργούμε ένα ιεραρχικό δίκτυο από πάνω-προς-τα-κάτω, αρχίζουμε με τη δημιουργία μιας σελίδας η οποία δείχνει τις υποσελίδες και πώς αυτές συνδέονται. Το παράδειγμα που θα χρησιμοποιηθεί είναι το Ιεραρχικό Χρωματισμένο δίκτυο Petri που μοντελοποιεί το απλό πρωτόκολλο επικοινωνίας που παρουσιάστηκε στο Κεφάλαιο 4.

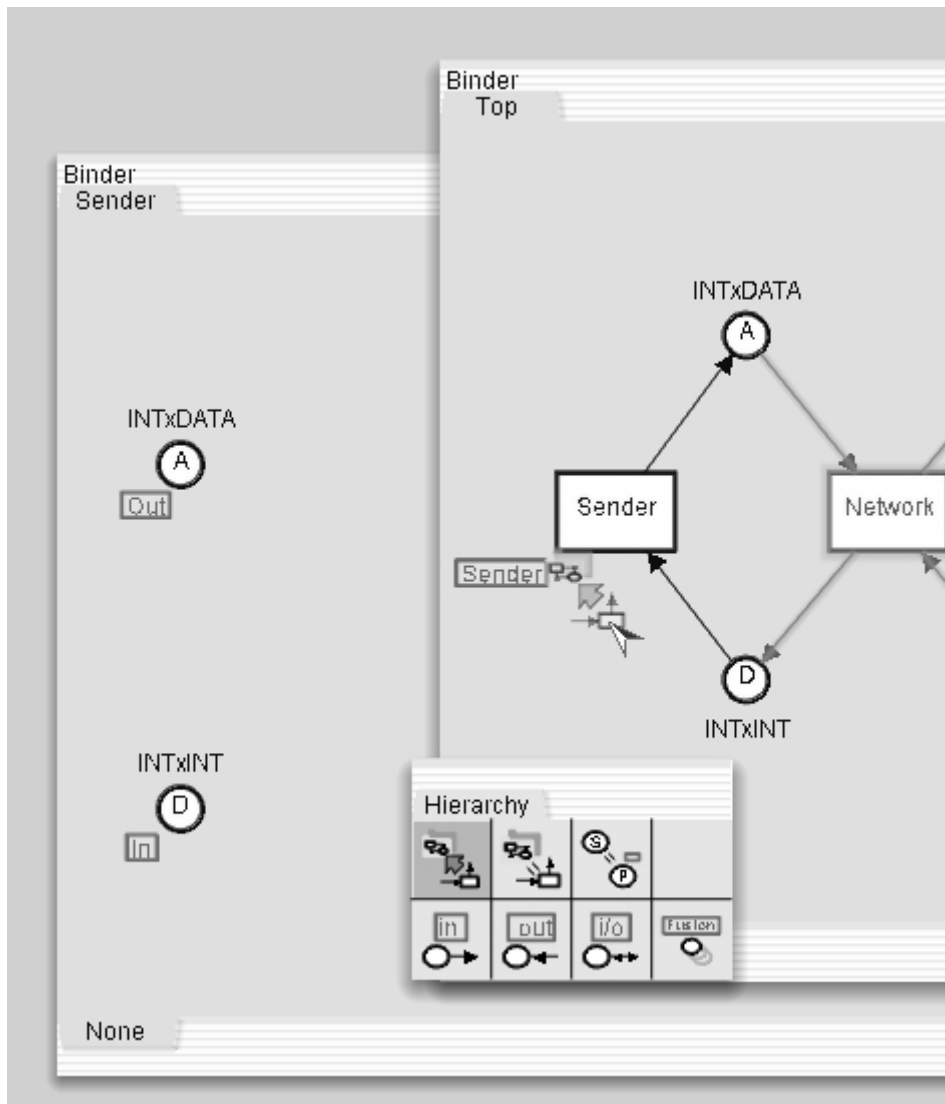
Στην Εικόνα 35 φαίνεται η σελίδα Top του δικτύου. Από το ευρετήριο έχουμε σύρει την παλέτα εργαλείων Ιεραρχίας, η οποία φαίνεται στο κάτω μέρος της Εικόνας 35.

Πρώτα επιλέγουμε το εργαλείο  και το εφαρμόζουμε στη μετάβαση Sender της σελίδας. Η σελίδα Top γίνεται τώρα υπερσελίδα.



Εικόνα 35: Η υπερσελίδα Top

Μία ετικέτα υποσελίδας προστίθεται στη μετάβαση Sender, η οποία είναι τώρα μια μετάβαση αντικατάστασης. Στη συνέχεια δημιουργείται μια νέα σελίδα που έχει αντίγραφο όλων των θέσεων που περιβάλλουν τη μετάβαση αντικατάστασης Sender. Η νέα σελίδα παίρνει το όνομα της μετάβασης αντικατάστασης, όπως φαίνεται στην Εικόνα 36. Οι ετικέτες τύπου πύλης προστίθενται αυτόματα στις θέσεις στη νέα υποσελίδα και οι αναθέσεις των πυλών στις υποδοχές γίνεται αυτόματα.






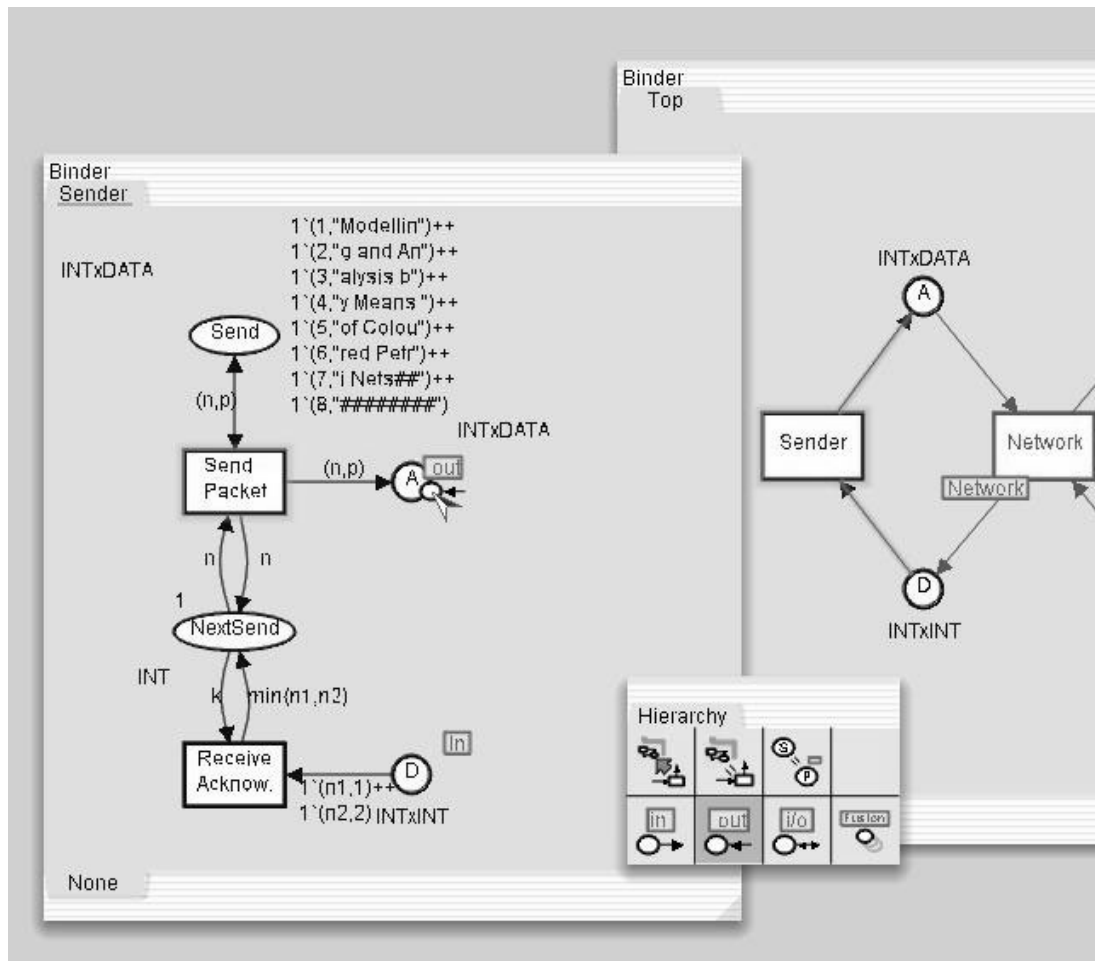
Εικόνα 36: Η νέα υποσελίδα Sender

5.4.2.2 Από Κάτω-προς-τα-Πάνω Ανάπτυξη

Όταν δημιουργούμε ένα ιεραρχικό δίκτυο από κάτω-προς-τα-πάνω, αρχίζουμε με τη δημιουργία των ξεχωριστών σελίδων. Σε αντίθεση με την από πάνω-προς-τα-κάτω ανάπτυξη, αυτή η προσέγγιση περιλαμβάνει τη δημιουργία των πιο λεπτομερών τμημάτων του δικτύου πρώτα. Στη συνέχεια, οι υπάρχουσες σελίδες ορίζονται ως υποσελίδες για τις μεταβάσεις αντικατάστασης. Θα χρησιμοποιηθεί πάλι το παράδειγμα του πρωτοκόλλου επικοινωνίας. Η σελίδα Sender θα οριστεί ως υποσελίδα της μετάβασης Sender στη σελίδα Top.


Η διαδικασία της σύνδεσης σελίδων χωρίζεται σε τρεις φάσεις. Πρώτα, όπως φαίνεται στην Εικόνα 37, αναθέτουμε ετικέτες τύπου πύλης στις θέσεις της


υποσελίδας Sender, χρησιμοποιώντας τα εργαλεία    από την παλέτα εργαλείων Ιεραρχίας, την οποία έχουμε σύρει από το ευρετήριο.




Εικόνα 37: Ανάθεση ετικετών τύπου πύλης

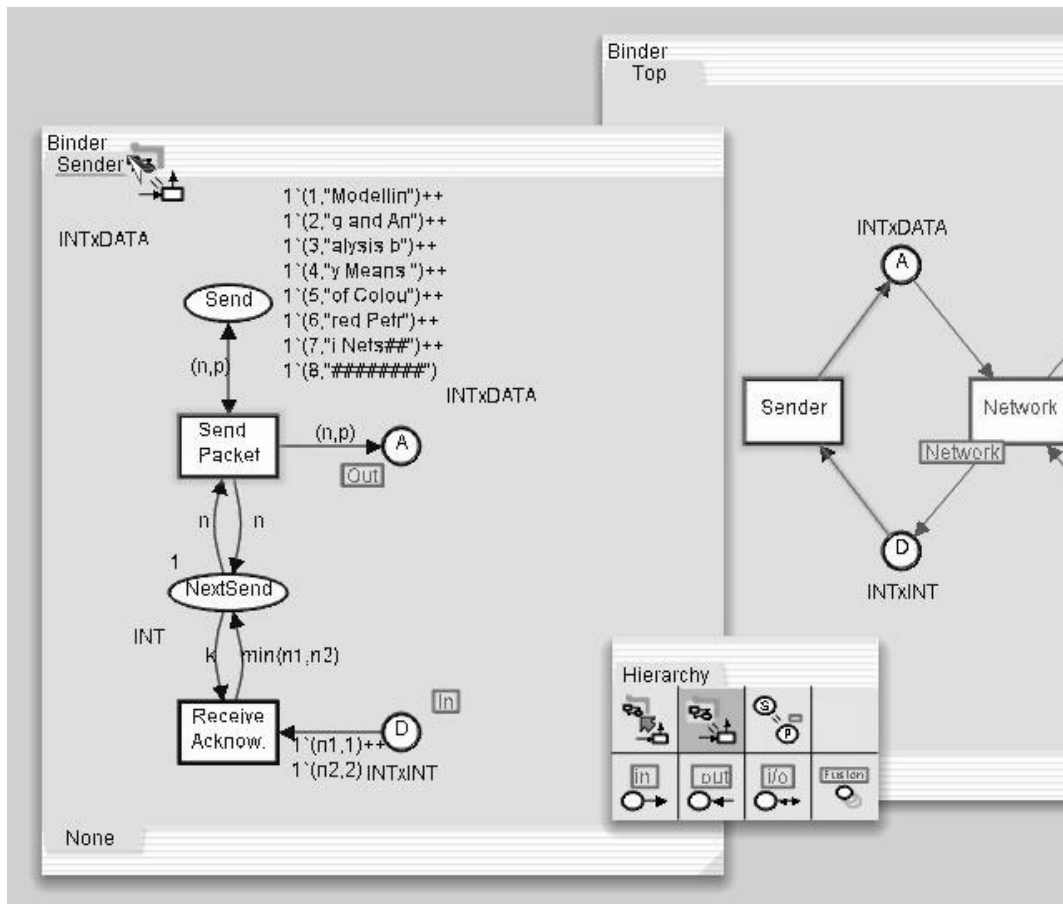
Έπειτα, αναθέτουμε υποσελίδες στις μεταβάσεις αντικατάστασης της υπερσελίδας,

χρησιμοποιώντας το εργαλείο  από την παλέτα εργαλείων Ιεραρχίας. Πρώτα εφαρμόζουμε το εργαλείο σε μια μετάβαση η οποία γίνεται μια μετάβαση αντικατάστασης. Στη συνέχεια το εργαλείο πρέπει να εφαρμοστεί στη σελίδα η οποία θα γίνει η υποσελίδα της μετάβασης αντικατάστασης. Το εικονίδιο του κέρσορα αλλάζει μετά την εφαρμογή του εργαλείου σε μια μετάβαση και υποδεικνύει ότι πρέπει να εφαρμοστεί και σε μια σελίδα. Στη Εικόνα 38 φαίνεται η εφαρμογή του εργαλείου στη σελίδα Sender.

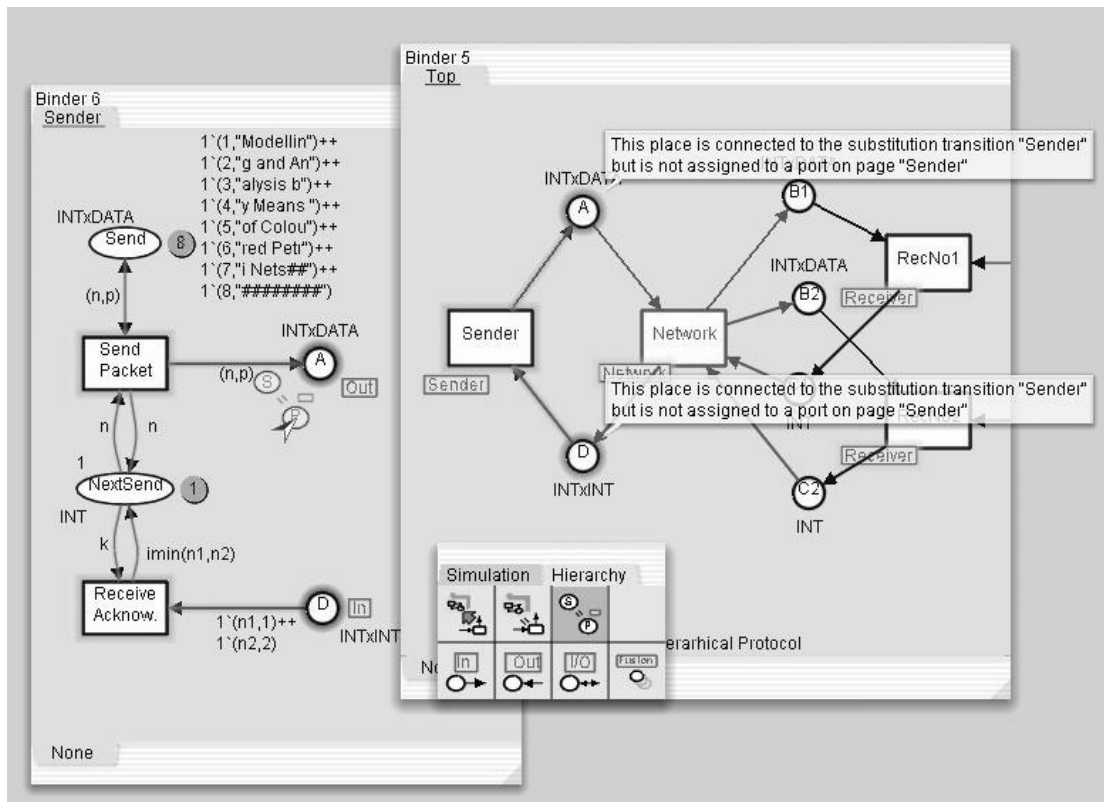
Μετά την εφαρμογή του εργαλείου , θα εμφανιστούν **μηνύματα σφαλμάτων (error messages)**, τα οποία υποδεικνύουν ποιες υποδοχές της υπερσελίδας δεν έχουν ακόμα ανατεθεί στις πύλες της υποσελίδας. Οι μη ανατιθέμενες πύλες έχουν ένα σκούρο γκρι περίγραμμα, όπως φαίνεται στην Εικόνα 39.

Το τελικό βήμα για να συνδέσουμε μια υποσελίδα σε μια υπερσελίδα, είναι να

αναθέσουμε κάθε πύλη σε μια υποδοχή, χρησιμοποιώντας το εργαλείο  από την παλέτα εργαλείων Ιεραρχίας. Αρχικά εφαρμόζουμε το εργαλείο σε μια πύλη. Το εικονίδιο του κέρσορα τώρα αλλάζει και υποδεικνύει ότι το εργαλείο πρέπει να



Εικόνα 38: Ανάθεση της υποσελίδας Sender



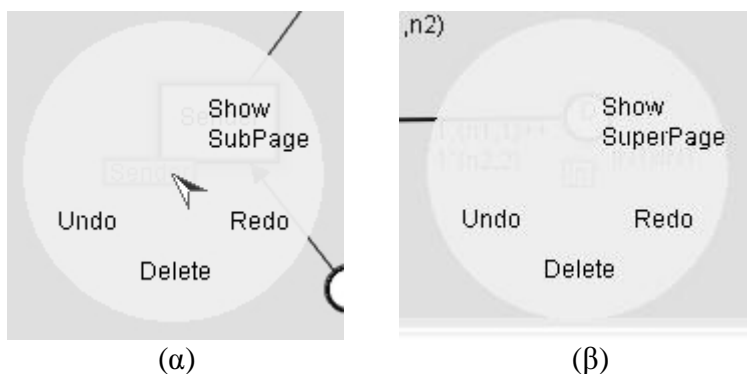
Εικόνα 39: Μηνύματα σφαλμάτων

εφαρμοστεί σε μια υποδοχή. Τέλος εφαρμόζουμε το εργαλείο σε μια υποδοχή της υπερσελίδας.

Μια υποσελίδα μπορεί να χρησιμοποιηθεί περισσότερες από μία φορά, δηλαδή μπορεί να συσχετιστεί με περισσότερες από μία μετάβαση αντικατάστασης. Για να δημιουργήσουμε πολλαπλά στιγμιότυπα μιας υποσελίδας, απλά χρησιμοποιούμε τη μέθοδο που περιγράφηκε παραπάνω, για να αναθέσουμε σε μια μετάβαση, μια σελίδα, η οποία είναι ήδη υποσελίδα μιας άλλης μετάβασης. Αν το όνομα της σελίδας ακολουθείται από αριθμούς, παραδείγματος χάριν (1), (2) κτλ, τότε υπάρχουν πολλαπλά στιγμιότυπα της συγκεκριμένης σελίδας. Τα μαρκάρια ενός στιγμιότυπου μιας υποσελίδας είναι εντελώς ανεξάρτητα από το μαρκάρισμα ενός άλλου στιγμιότυπου της ίδιας σελίδας.

5.4.3 Πλοήγηση μεταξύ των Υποσελίδων και των Υπερσελίδων

Για να πλοηγηθούμε γρήγορα μεταξύ των σελίδων ενός Ιεραρχικού δικτύου Petri, χρησιμοποιούμε το **μενού μαρκαρισμάτων Υποσελίδας (Subpage marking menu)**, που φαίνεται στην Εικόνα 40(α) και το **μενού μαρκαρισμάτων Τύπου Πύλης (Port type marking menu)**, που φαίνεται στην Εικόνα 40(β). Για να εμφανιστεί το μενού μαρκαρισμάτων Υποσελίδας, κρατάμε πατημένο το δεξί πλήκτρο του ποντικιού πάνω σε μια επιγραφή υποσελίδας, ενώ για να εμφανιστεί το μενού μαρκαρισμάτων Τύπου Πύλης, κρατάμε πατημένο το δεξί πλήκτρο του ποντικιού πάνω σε μια επιγραφή τύπου πύλης. Για την εμφάνιση μιας υποσελίδας επιλέγουμε την εντολή Show Subpage από το μενού μαρκαρισμάτων Υποσελίδας και για την εμφάνιση μιας υπερσελίδας επιλέγουμε αντίστοιχα την εντολή Show Superpage από το μενού μαρκαρισμάτων Τύπου Πύλης. Μετά την εφαρμογή αυτών των εργαλείων, θα εμφανιστεί η αντίστοιχη σελίδα.




Εικόνα 40: (α) Μενού μαρκαρισμάτων Υποσελίδας
(β) Μενού μαρκαρισμάτων Τύπου Πύλης


5.4.4 Θέσεις Συγχώνευσης

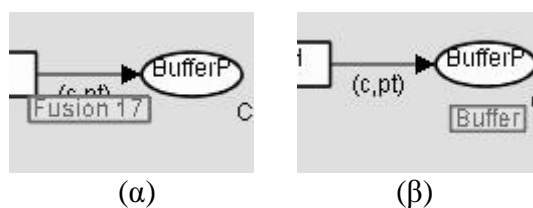
Το πρόγραμμα CPN Tools υποστηρίζει μια μέθοδο για τον ορισμό των θέσεων συγχώνευσης. Οτιδήποτε συμβεί σε μια θέση που ανήκει στο σύνολο συγχώνευσης, συμβαίνει και σε όλες τις άλλες θέσεις που ανήκουν στο σύνολο συγχώνευσης. Οι θέσεις συγχώνευσης είναι λειτουργικά ίδιες.

Αν όλα τα μέλη ενός συνόλου συγχώνευσης βρίσκονται στην ίδια σελίδα, μπορούμε να αντικαταστήσουμε το σύνολο μόνο με μία θέση και να τη συνδέσουμε με όλα τα τόξα που ήταν συνδεδεμένα σε κάθε μέλος του συνόλου. Αν τα μέλη βρίσκονται σε διαφορετικές σελίδες, μπορούμε να αντιγράψουμε οτιδήποτε υπάρχει σε αυτές τις σελίδες σε μία σελίδα και να επαναλάβουμε την προηγούμενη διαδικασία. Αντιστρόφως, αν ένα δίκτυο περιέχει μια θέση που έχει πολλά τόξα συνδεδεμένα σε αυτήν, ή απαιτεί πολύ μακριά τόξα για τη συνδέσουμε, μπορούμε να τη διασπάσουμε σε πολλές θέσεις, στην ίδια ή σε διαφορετικές σελίδες, κι έτσι να απλοποιήσουμε τη γραφική απεικόνιση του δικτύου χωρίς να αλλάξουμε το νόημα του. Αυτή η διαδικασία της διάσπασης είναι συνηθισμένο γεγονός κατά τη διάρκεια της ανάπτυξης ενός Χρωματισμένου δικτύου Petri.

5.4.4.1 Δημιουργία Θέσεων Συγχώνευσης

Οι θέσεις συγχώνευσης δημιουργούνται, χρησιμοποιώντας το εργαλείο  από την παλέτα εργαλείων Ιεραρχίας. Μετά την εφαρμογή του εργαλείου πάνω σε μια θέση, μια ετικέτα συγχώνευσης προστίθεται στη θέση. Η ετικέτα συγχώνευσης αρχικά τοποθετείται σε μια προεπιλεγμένη θέση και έχει ένα προεπιλεγμένο όνομα για το σύνολο συγχώνευσης. Στη συνέχεια μπορούμε να αλλάξουμε τη θέση της ετικέτας και το όνομα του συνόλου συγχώνευσης. Στην Εικόνα 41(α) μετά την

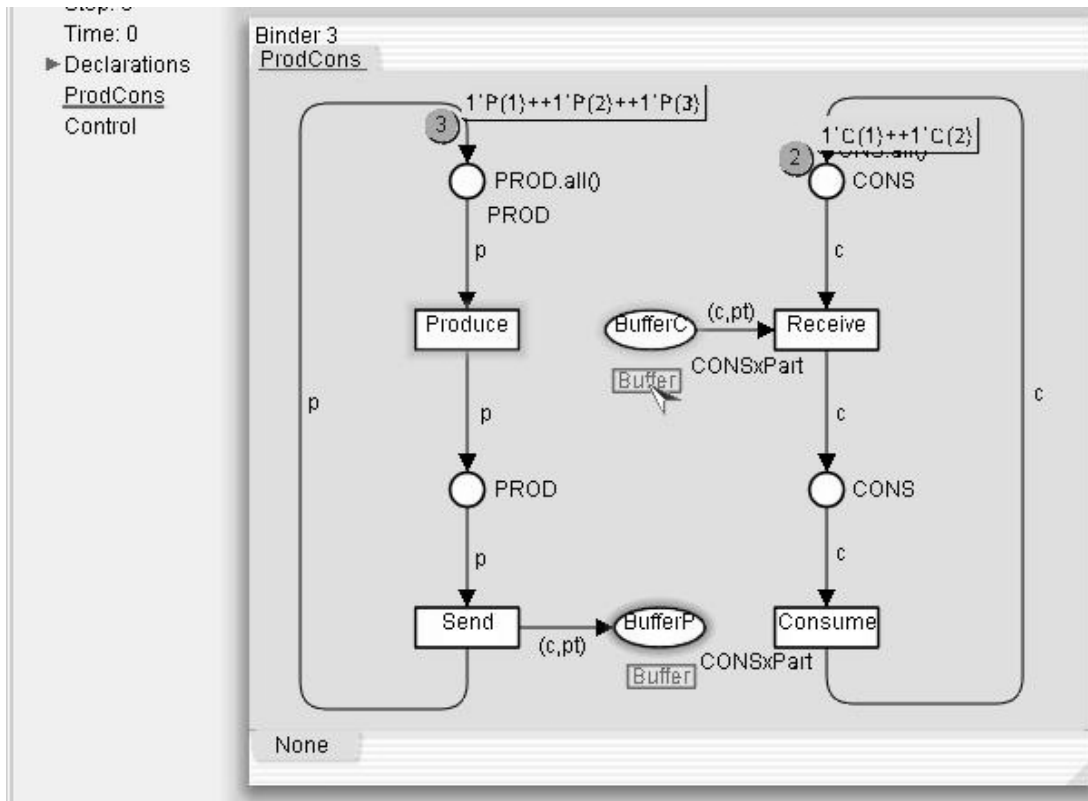
εφαρμογή του εργαλείου , έχει προστεθεί στη θέση Buffer η ετικέτα συγχώνευσης με όνομα Fusion 17. Στην Εικόνα 41(β) έχουμε αλλάξει το όνομα της ετικέτας συγχώνευσης σε Buffer.



Εικόνα 41: Η θέση συγχώνευσης Buffer

5.4.4.2 Μέλη ενός Συνόλου Συγχώνευσης

Μπορούμε να βρούμε τα μέλη ενός συνόλου συγχώνευσης, τοποθετώντας τον κέρσορα πάνω από μια ετικέτα συγχώνευσης, όπως φαίνεται στην Εικόνα 42. Το γαλάζιο περίγραμμα γύρω από μια θέση υποδεικνύει ότι στη θέση αυτή ανήκει η ετικέτα συγχώνευσης. Τα ροζ περιγράμματα γύρω από κάποιες θέσεις υποδεικνύουν τις άλλες θέσεις που ανήκουν στο ίδιο σύνολο συγχώνευσης, ενώ οι ροζ υπογραμμίσεις υποδεικνύουν τις σελίδες που περιέχουν αυτές τις θέσεις.



Εικόνα 42: Τα μέλη του συνόλου συγχώνευσης Buffer

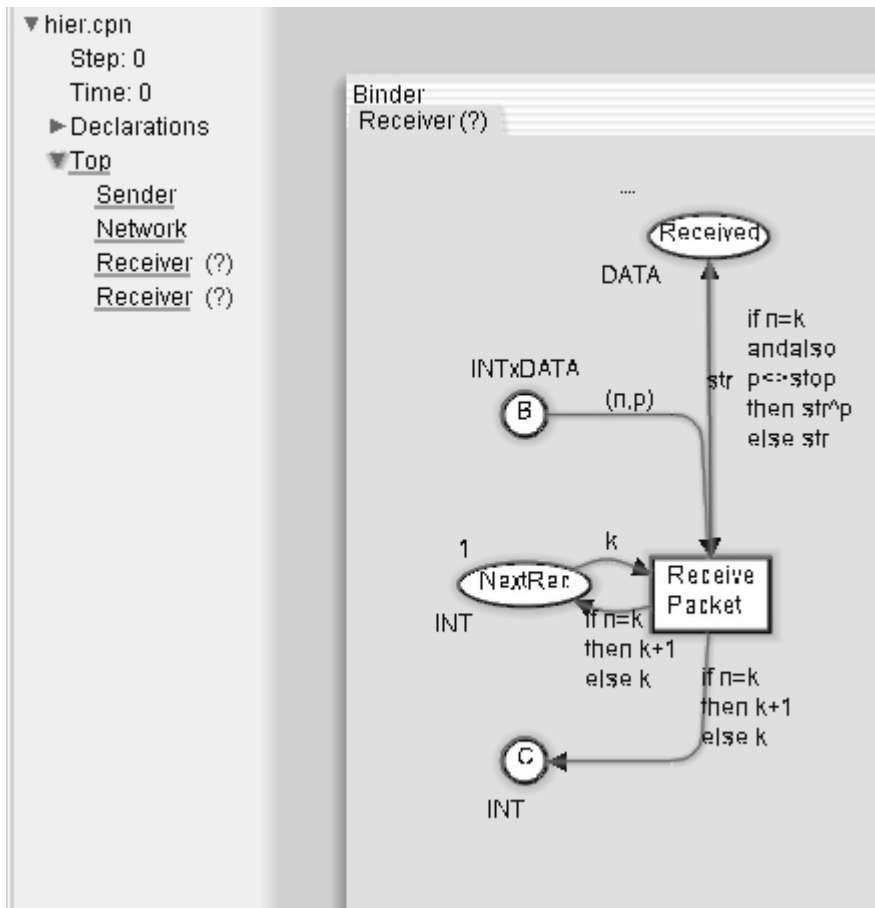
5.5 Αναλύση ενός Χρωματισμένου Δικτύου Petri

5.5.1 Συντακτικός Έλεγχος

Το πρόγραμμα CPN Tools κάνει αυτόματα συντακτικό έλεγχο των δικτύων, καθώς δημιουργούμε ένα δίκτυο ή όταν ανοίγουμε ένα αποθηκευμένο. Από τις χρωματικές υποδείξεις μπορούμε να δούμε σε ποιο σημείο βρίσκεται ο έλεγχος. Όταν γίνεται συντακτικός έλεγχος μιας σελίδας, το όνομα της σελίδας στο ευρετήριο είναι υπογραμμισμένο με μια χρωματιστή γραμμή. Αν η σελίδα είναι ανοιχτή, τότε είναι χρωματισμένα και τα στοιχεία του δικτύου Petri στα οποία ανήκει το χρώμα.

Στην Εικόνα 43 φαίνεται η κατάσταση του αποστολέα Sender του πρωτοκόλλου επικοινωνίας πριν το συντακτικό έλεγχο. Το πορτοκαλί χρώμα υποδεικνύει ότι το στοιχείο δεν έχει ακόμα ελεγχθεί για συντακτικά σφάλματα.

Όταν ανοίγουμε ένα αποθηκευμένο δίκτυο, ο συντακτικός έλεγχος κάνει δυο-τρία λεπτά για να ολοκληρωθεί. Κατά τη διάρκεια αυτή της φάσης, τα στοιχεία θα αλλάξουν χρώμα από πορτοκαλί σε κίτρινο και από κίτρινο σε κόκκινο, αν υπάρχουν σφάλματα ή σε καθόλου χρώμα, αν το στοιχείο είναι σωστό.

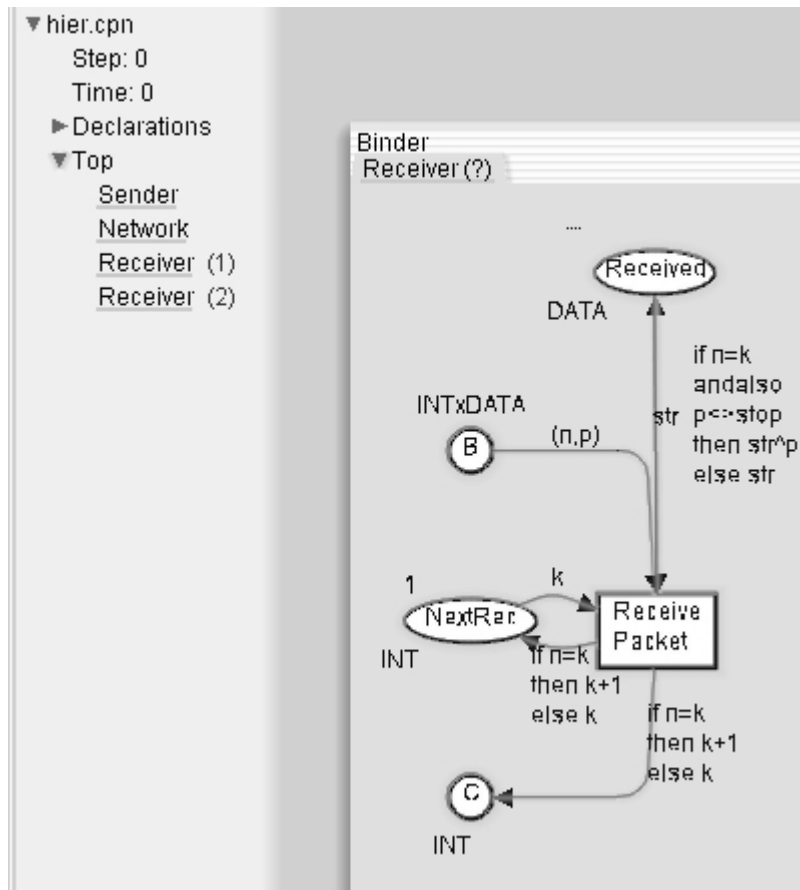


Εικόνα 43: Πριν το συντακτικό έλεγχο

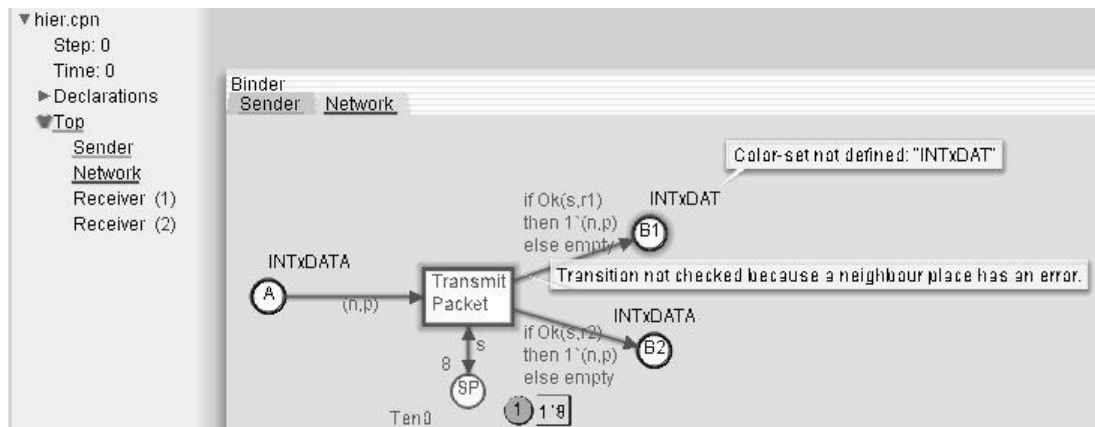
Αν το πορτοκαλί χρώμα παραμείνει, αυτό σημαίνει ότι είτε λείπει κάτι, είτε ότι υπάρχει σφάλμα σε ένα συσχετιζόμενο στοιχείο δικτύου. Στη συνέχεια εξηγείται πότε ελέγχονται τα στοιχεία του δικτύου :

- Μια θέση θα ελεγχθεί όταν :
 - έχει μια επιγραφή συνόλου χρωμάτων και
 - καμία από τις συσχετιζόμενες πύλες ή υποδοχές δεν έχει σφάλμα
- Ένα τόξο θα ελεγχθεί όταν :
 - έχει ελεγχθεί η θέση εισόδου του και
 - έχει μια επιγραφή τόξου
- Μια μετάβαση θα ελεγχθεί όταν :
 - όλες οι θέσεις που την περιβάλλουν έχουν ελεγχθεί και
 - όλα τα τόξα που την περιβάλλουν έχουν ελεγχθεί

Κατά τη διάρκεια του συντακτικού ελέγχου, όπως φαίνεται στην Εικόνα 44, τα στοιχεία του δικτύου έχουν κίτρινο περίγραμμα και οι σελίδες του δικτύου έχουν κίτρινη υπογράμμιση. Αντίθετα το κόκκινο χρώμα υποδεικνύει ότι το στοιχείο έχει ελεγχθεί αλλά έχει κάποιο σφάλμα. Στην περίπτωση σφάλματος εμφανίζεται ένα μήνυμα λάθους, όπως φαίνεται στην Εικόνα 45. Τα στοιχεία που συνδέονται σε ένα στοιχείο που έχει σφάλμα, παραδείγματος χάριν οι μεταβάσεις που συνδέονται σε μια θέση με σφάλμα, δεν ελέγχονται μέχρι να διορθωθεί το σφάλμα.



Εικόνα 44: Κατά τη διάρκεια του συντακτικού ελέγχου



Εικόνα 45: Ύπαρξη σφάλματος

Αν υπάρχει σφάλμα σε κάποια δήλωση, το όνομα της λανθασμένης δήλωσης στο ευρετήριο αλλάζει σε ErrorDecl και υπογραμμίζεται με κόκκινο χρώμα. Η καταχώρηση δικτύου και όλες οι σελίδες που χρησιμοποιούν τη δήλωση θα υπογραμμιστούν και αυτές με κόκκινο χρώμα, όπως φαίνεται στην Εικόνα 46. Για να δούμε το μήνυμα σφάλματος, μετακινούμε το ποντίκι πάνω από την καταχώρηση ErrorDecl.

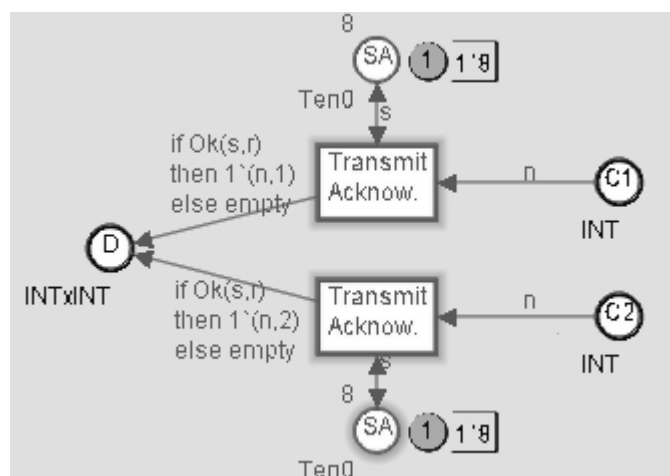
```

▼ HierarchicalProtocol.cpn
  Step: 0
  Time: 0
  ▼ Declarations
    ► color INT
    ► color DATA
    ▼ ErrorDecl
      color INTxDATA = product INT DATA;
    ► color INTxINT
    ► var n k n1 n2
    ► var p str
    ► val stop = "#####";
    ► color Ten0
    ► color Ten1
    ► var s
    ► var r r1 r2
    ► fun Ok(s:Ten0,r:Ten1) = (r<=s);
    ► fun imin(i:int,j:int)=if (i<j) then i else j;
  ▼ Top
    Sender
    Network
    Receiver (1)
    Receiver (2)

```

Εικόνα 46: Σφάλμα σε μια δήλωση

Στην Εικόνα 47 υπάρχει ένα κοκκινωπό περίγραμμα γύρω από τις μεταβάσεις TransmitAcknowledgement και τη θέση SA. Το κοκκινωπό χρώμα υποδεικνύει ότι το πρόθεμα του ονόματος (μέχρι το πρώτο κενό διάστημα) ενός κόμβου δεν είναι μοναδικό και/ή έγκυρο. Τα προθέματα πρέπει να είναι μοναδικά για το εργαλείο του χώρου καταστάσεων. Το χρώμα αυτό παρουσιάζεται μόνο στα στοιχεία του δικτύου και όχι στο ευρετήριο.




Εικόνα 47: Μη μοναδικά ονόματα

5.5.2 Προσομοίωση

Αφού έχουμε ανοίξει ένα αποθηκευμένο δίκτυο ή έχουμε δημιουργήσει ένα νέο δίκτυο και έχει γίνει με επιτυχία ο συντακτικός έλεγχος, δηλαδή δεν υπάρχουν σφάλματα, μπορούμε να προσομοιώσουμε το δίκτυο.

Για να αρχίσουμε μια προσομοίωση, σύρουμε την παλέτα εργαλείων Προσομοίωσης, που φαίνεται στην Εικόνα 48, από το ευρετήριο στην επιφάνεια εργασίας και εφαρμόζουμε ένα από τα εργαλεία προσομοίωσης. Αν εφαρμόσουμε για παράδειγμα

το εργαλείο  , θα αρχίσει η εκτέλεση της προσομοίωσης. Ο αριθμός των βημάτων προσομοίωσης καθορίζεται από τον αριθμό που αναγράφεται στο εικονίδιο του εργαλείου.



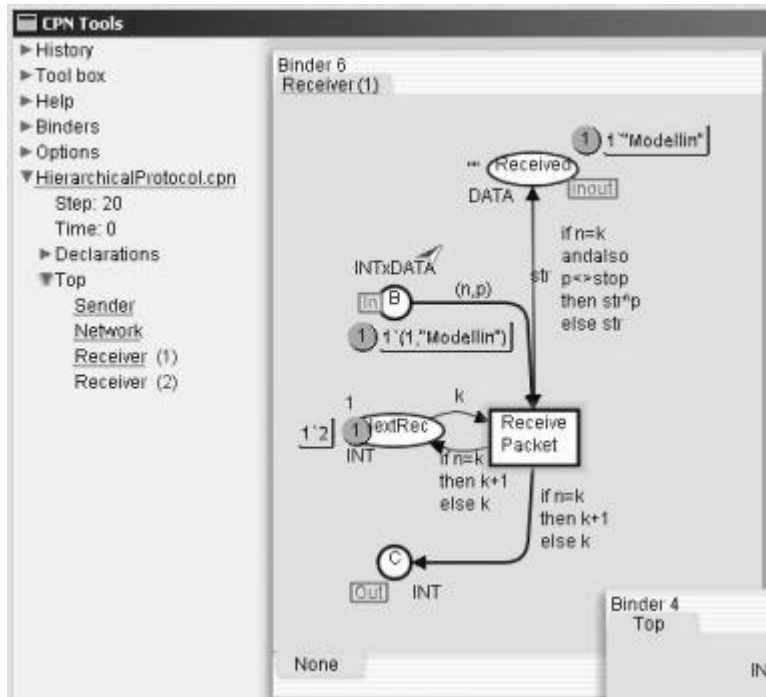
Εικόνα 48: Η παλέτα εργαλείων Προσομοίωσης

Όταν τρέχει μια προσομοίωση, έχουμε την ακόλουθη ανάδραση (feedback) προσομοίωσης :

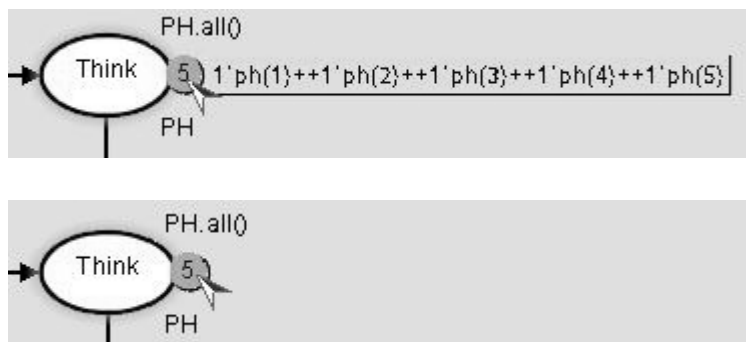
- Τα τρέχοντα μαρκαρίσματα των θέσεων φαίνονται δίπλα στις θέσεις
 - ο αριθμός των μαρκών σε μια θέση φαίνεται σε ένα πράσινο κύκλο
 - οι αντίστοιχες τιμές των μαρκών φαίνονται σε πράσινα κουτιά
- Οι ενεργοποιημένες μεταβάσεις έχουν πράσινα περιγράμματα και οι σελίδες που περιέχουν ενεργοποιημένες μεταβάσεις έχουν πράσινες υπογραμμίσεις, οι οποίες φαίνονται στο ευρετήριο και στις ετικέτες των σελίδων
- Τα βήματα και ο χρόνος της προσομοίωσης φαίνονται στο ευρετήριο κάτω από το όνομα του δικτύου που προσομοιώνεται

Στην Εικόνα 49 φαίνεται η ανάδραση της προσομοίωσης του Ιεραρχικού Χρωματισμένου δικτύου Petri που μοντελοποιεί το απλό πρωτόκολλο επικοινωνίας.


Μερικές φορές η ανάδραση της προσομοίωσης κρύβει τμήματα του δικτύου, δυσκολεύοντας την ανάγνωση τους. Για αυτό το λόγο, έχουμε τη δυνατότητα να μετακινήσουμε τις τρέχουσες πληροφορίες μαρκαρισμάτων, σύροντας τις σε κάποια άλλη θέση. Αν σύρουμε ένα πράσινο κύκλο, θα μετακινηθεί επίσης και το πράσινο κουτί με τις τιμές των μαρκών. Το αντίστροφο όμως δεν ισχύει, δηλαδή αν σύρουμε ένα κουτί, το κουτί θα μετακινηθεί ανεξάρτητα από τον κύκλο. Μπορούμε επίσης να κρύψουμε τις τιμές των μαρκών, κάνοντας αριστερό κλικ στο πράσινο κύκλο που περιέχει τις αντίστοιχες τιμές των μαρκών, όπως φαίνεται στην Εικόνα 50.



Εικόνα 49: Ανάδραση προσομοίωσης



Εικόνα 50: Απόκρυψη των τιμών μαρκών

Για να επαναφέρουμε μετά την προσομοίωση, ένα Χρωματισμένο δίκτυο Petri πίσω στην αρχική του κατάσταση, μπορούμε να χρησιμοποιήσουμε το εργαλείο  από την παλέτα εργαλείων Προσομοίωσης.

5.5.2.1 Αναφορά Προσομοίωσης

Η **αναφορά προσομοίωσης (simulation report)** είναι ένα αρχείο κειμένου που περιέχει πληροφορίες σχετικά με τις μεταβάσεις που εκτελούνται κατά τη διάρκεια μιας προσομοίωσης. Μια αναφορά προσομοίωσης αποθηκεύεται στον κατάλογο C:\tmp. Αν ο κατάλογος αυτός δεν υπάρχει, τότε η αναφορά προσομοίωσης δεν θα αποθηκευτεί. Οι αναφορές προσομοιώσεων αποθηκεύονται προεπιλεγμένα ως C:\tmp\simrep-<process id>.txt.

Μια αναφορά προσομοίωσης θα δημιουργηθεί μόνο αν ενεργοποιήσουμε την επιλογή “Save Report” από τις επιλογές Options που βρίσκονται στο ευρετήριο, όπως φαίνεται στην Εικόνα 51. Αφού ενεργοποιηθεί η επιλογή αυτή, τα ονόματα των μεταβάσεων καταγράφονται στην αναφορά προσομοίωσης. Για κάθε βήμα προσομοίωσης, αποθηκεύεται μία γραμμή με την εξής σύνταξη :

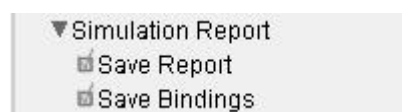
```
step      time      TransitionName @ (Instance:PageName)
```



Εικόνα 51: Η επιλογή Save Report

Μια πιο λεπτομερής αναφορά προσομοίωσης θα αποθηκευτεί, αν ενεργοποιήσουμε και την επιλογή Save Bindings, όπως φαίνεται στην Εικόνα 52. Σε αυτήν την περίπτωση, αποθηκεύονται στην αναφορά προσομοίωσης και οι αναθέσεις των μεταβλητών. Για κάθε βήμα της προσομοίωσης, αποθηκεύεται μία γραμμή για κάθε μεταβλητή της μετάβασης. Η σύνταξη της γραμμής είναι η εξής :

```
- variable = value
```



Εικόνα 52: Η επιλογή Save Bindings

Παρακάτω φαίνεται ένα παράδειγμα μιας αναφοράς προσομοίωσης του συστήματος **κατανομής πόρων (resource allocation)**, η οποία περιέχει πληροφορίες για τα τέσσερα πρώτα βήματα της προσομοίωσης.

```
1 0 T2 @ (1:Top)
2 0 T1 @ (1:Top)
3 0 T3 @ (1:Top)
- i = 0
- x = p
4 0 T4 @ (1:Top)
- i = 0
- x = p
```

Η πρώτη γραμμή προσδιορίζει ότι στο Βήμα 1, εκτελέστηκε μια μετάβαση με όνομα T2 στο στιγμιότυπο 1 της σελίδας Top. Ομοίως μια μετάβαση με όνομα T1 στην ίδια σελίδα εκτελέστηκε στο Βήμα 2. Η τρίτη γραμμή υποδεικνύει ότι η μετάβαση T3 (επίσης στην ίδια σελίδα) εκτελέστηκε στο Βήμα 3. Επιπλέον, όταν η μετάβαση T3 εκτελέστηκε, η μεταβλητή i είχε την τιμή 0 και η μεταβλητή x είχε την τιμή p. Ομοίως και για το Βήμα 4.

Η παραπάνω αναφορά προσομοίωσης δημιουργήθηκε με τον εξής τρόπο :

1. Ανόιξαμε το δίκτυο με όνομα Resource Allocation

2. Ενεργοποιήσαμε την επιλογή Save Report
3. Εκτελέστηκαν δύο βήματα προσομοίωσης
4. Ενεργοποιήσαμε την επιλογή Save Bindings
5. Δύο επιπρόσθετα βήματα προσομοίωσης εκτελέστηκαν

5.5.3 Ανάλυση του Χώρου Καταστάσεων

5.5.3.1 Είσοδος στο Εργαλείο του Χώρου Καταστάσεων

Προτού υπολογίσουμε και αναλύσουμε το χώρο καταστάσεων, πρέπει να παράγουμε τον κώδικα του χώρου καταστάσεων, δηλαδή τον κώδικα της γλώσσας CPN ML που χρησιμοποιείται για να υπολογίσουμε και να αναλύσουμε τους χώρους καταστάσεων. Ο κώδικας αυτός παράγεται όταν εισαχθούμε στο **εργαλείο του χώρου καταστάσεων (state space tool)**.

Για να μπορέσουμε να εισαχθούμε στο εργαλείο του χώρου καταστάσεων, πρέπει να ικανοποιούνται ορισμένες συνθήκες :

- (i) Το δίκτυο δεν πρέπει να περιέχει συντακτικά σφάλματα.
- (ii) Όλες οι θέσεις, οι μεταβάσεις και οι σελίδες του δικτύου πρέπει να έχουν ονόματα.
- (iii) Τα ονόματα των θέσεων, των μεταβάσεων και των σελίδων πρέπει να έχουν μοναδικά προθέματα μέχρι το πρώτο κενό διάστημα, και τα προθέματα πρέπει να είναι αναγνωριστικά (identifiers) της γλώσσας CPN ML. Μία θέση μπορεί να έχει το ίδιο πρόθεμα ονόματος με μια μετάβαση, αλλά δύο θέσεις (ή δύο μεταβάσεις) δεν μπορούν να έχουν το ίδιο πρόθεμα ονόματος.

Τα συντακτικά σφάλματα και τα μη-μοναδικά προθέματα ονομάτων θα αναγνωριστούν κατά τη διάρκεια του συντακτικού ελέγχου.

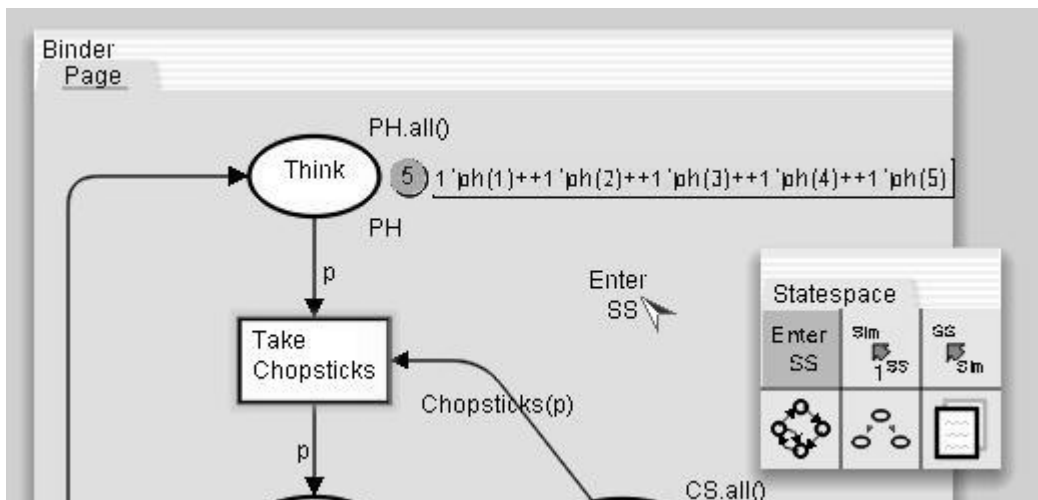
Για να εισαχθούμε στο εργαλείο του χώρου καταστάσεων, εφαρμόζουμε το εργαλείο



από την παλέτα εργαλείων του Χώρου Καταστάσεων, σε ένα από τα **φύλλα (sheets)** που περιέχουν μια σελίδα του δικτύου, όπως φαίνεται στην Εικόνα 53. Η παλέτα εργαλείων του Χώρου Καταστάσεων φαίνεται στην Εικόνα 53 κάτω δεξιά.

Η είσοδος στο εργαλείο του χώρου καταστάσεων παίρνει κάποιο χρόνο. Στο **παράθυρο της κονσόλας (console window)** εμφανίζεται κάποιο κείμενο (Εικόνα 54), το οποίο υποδεικνύει την πρόοδο που έχει γίνει μετά την είσοδο στο εργαλείο του χώρου καταστάσεων.

Όταν εισαχθούμε με επιτυχία στο εργαλείο του χώρου καταστάσεων, μια πράσινη **φούσκα κατάστασης (state bubble)** εμφανίζεται στο ευρετήριο. Αν συμβεί ένα σφάλμα κατά την είσοδο μας στο εργαλείο του χώρου καταστάσεων, μία κόκκινη φούσκα κατάστασης, όπως αυτή της Εικόνας 55, θα εμφανιστεί στο ευρετήριο. Αν τοποθετήσουμε τον κέρσορα πάνω από την κόκκινη φούσκα, θα εμφανιστεί το μήνυμα σφάλματος.



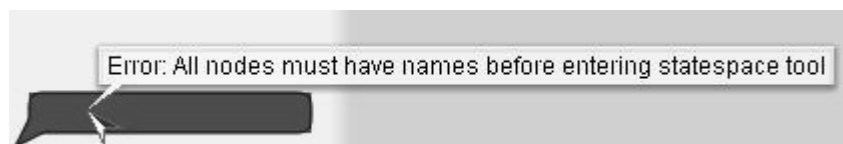
Εικόνα 53: Άνοιγμα του εργαλείου του χώρου καταστάσεων

```

CPN Tools
log_handle = 1996
Entering state space tool [1:8]
Entering state space tool [2:8]
Entering state space tool [3:8]
Entering state space tool [4:8]
Entering state space tool [5:8]

```



Εικόνα 54: Ανάδραση με μορφή κειμένου



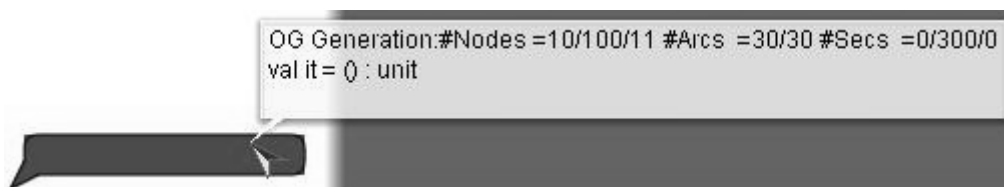
Εικόνα 55: Κόκκινη φούσκα κατάστασης

5.5.3.2 Υπολογισμός του Χώρου Καταστάσεων και του SCC-γράφου

Μετά την είσοδο στο εργαλείο του χώρου καταστάσεων, υπολογίζουμε ένα χώρο καταστάσεων. Αν ο χώρος καταστάσεων αναμένεται να είναι μικρός, μπορούμε απλά


να εφαρμόσουμε το εργαλείο  από την παλέτα εργαλείων του Χώρου Καταστάσεων, σε ένα φύλλο που περιέχει μια σελίδα του δικτύου. Αν ο χώρος καταστάσεων αναμένεται να είναι μεγάλος, ίσως χρειαστεί να αλλάξουμε τις επιλογές  για το εργαλείο .

Αν ο υπολογισμός ήταν επιτυχής, θα εμφανιστεί μια πράσινη φούσκα κατάστασης στο ευρετήριο. Όταν τοποθετούμε τον κέρσορα πάνω από την πράσινη φούσκα κατάστασης, εμφανίζονται τα στατιστικά της παραγωγής του χώρου καταστάσεων, όπως φαίνεται στην Εικόνα 56. Αντίθετα, αν ο υπολογισμός απέτυχε, εμφανίζεται μια κόκκινη φούσκα κατάστασης, η οποία εμφανίζει όπως και πριν το μήνυμα σφάλματος, όταν τοποθετήσουμε τον κέρσορα πάνω από αυτήν.




Εικόνα 56: Πράσινη φούσκα κατάστασης


Μετά τον υπολογισμό του χώρου καταστάσεων, μπορούμε να υπολογίσουμε το **γράφο των ισχυρά συνδεδεμένων τμημάτων (strongly connected components)** ή συντομογραφικά το **SCC-γράφο (SCC-graph)**, εφαρμόζοντας από την παλέτα

εργαλείων του Χώρου Καταστάσεων, το εργαλείο  , σε ένα φύλλο που περιέχει μια σελίδα του δικτύου. Αν ο υπολογισμός του SCC-γράφου είναι επιτυχής, εμφανίζεται μια πράσινη φούσκα κατάστασης στο ευρετήριο, ενώ αν αποτύχει, εμφανίζεται μια κόκκινη φούσκα κατάστασης, η οποία με την κατάλληλη τοποθέτηση του κέρσορα εμφανίζει το μήνυμα σφάλματος.

5.5.3.3 Αποθήκευση της Αναφοράς του Χώρου Καταστάσεων

Αφού έχουμε υπολογίσει ένα χώρο καταστάσεων και προαιρετικά το SCC-γράφο, μπορούμε να σώσουμε την **αναφορά του χώρου καταστάσεων (state space report)** με τους υπολογισμούς. Για να σώσουμε μια αναφορά, εφαρμόζουμε από την παλέτα


εργαλείων του Χώρου Καταστάσεων, το εργαλείο  , σε ένα φύλλο που περιέχει μια σελίδα του δικτύου. Τότε εμφανίζεται ένα παράθυρο διαλόγου, το οποίο μας ζητάει να καθορίσουμε το όνομα του αρχείου αναφοράς που θέλουμε να δημιουργήσουμε και το φάκελο που θα αποθηκευτεί.

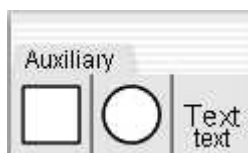
Τα περιεχόμενα της αναφοράς καθορίζονται από τις επιλογές του εργαλείου . Επίσης μπορεί να τροποποιηθεί η μορφοποίηση κειμένου για τα ονόματα των θέσεων και των μεταβάσεων.

Στο Κεφάλαιο 4 παρουσιάστηκε λεπτομερώς η αναφορά του χώρου καταστάσεων του Χρωματισμένου δικτύου Petri που μοντελοποιεί το απλό πρωτόκολλο επικοινωνίας.


5.5.3.4 Υποβολή Ερωτημάτων

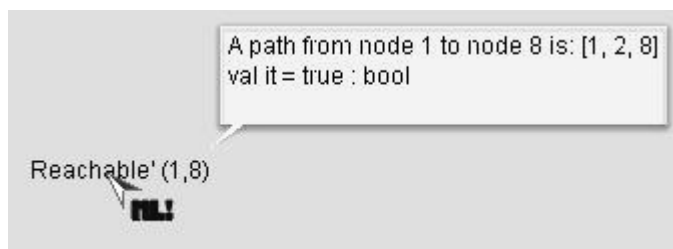
Τα **ερωτήματα (queries)** χρησιμοποιούνται για να εξεταστούν οι ιδιότητες ενός Χρωματισμένου δικτύου Petri. Μπορούμε, παραδείγματος χάριν, να εξετάσουμε τις ιδιότητες της περατότητας, της διάρκειας, της προσιτότητας, της οικείας κατάστασης και της δικαιοσύνης χρησιμοποιώντας τυποποιημένα ερωτήματα. Μπορούμε επίσης να συντάξουμε τα δικά μας ερωτήματα με τη βοήθεια της γλώσσας CPN ML.

Τα ερωτήματα υποβάλλονται, εφαρμόζοντας πρώτα το εργαλείο  από την παλέτα των Βοηθητικών εργαλείων, η οποία φαίνεται στην Εικόνα 57, για να συντάξουμε το βοηθητικό κείμενο που περιέχει τη συνάρτηση του ερωτήματος. Οι διαθέσιμες συναρτήσεις ερωτημάτων περιέχονται σε ένα αρχείου κειμένου που περιέχεται μέσα στο εργαλείο CPN Tools.



Εικόνα 57: Η παλέτα των Βοηθητικών εργαλείων

Στη συνέχεια χρησιμοποιούμε το εργαλείο , από την παλέτα των εργαλείων προσομοίωσης. Στην Εικόνα 58 φαίνεται ένα παράδειγμα υποβολής ερωτήματος.

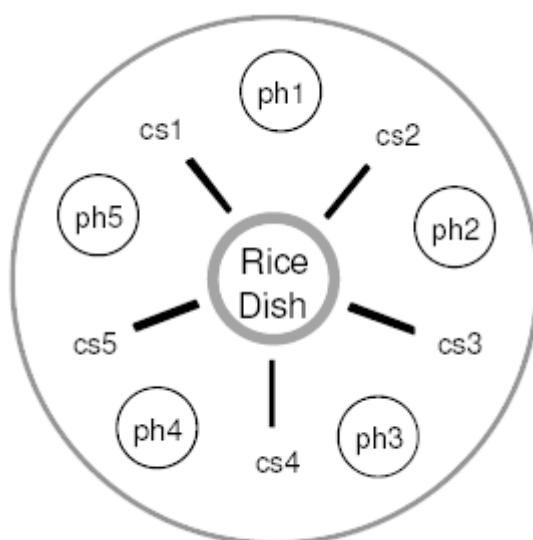


Εικόνα 58: Παράδειγμα υποβολής ερωτήματος

5.5.4 Το Δείπνο των Φιλοσόφων

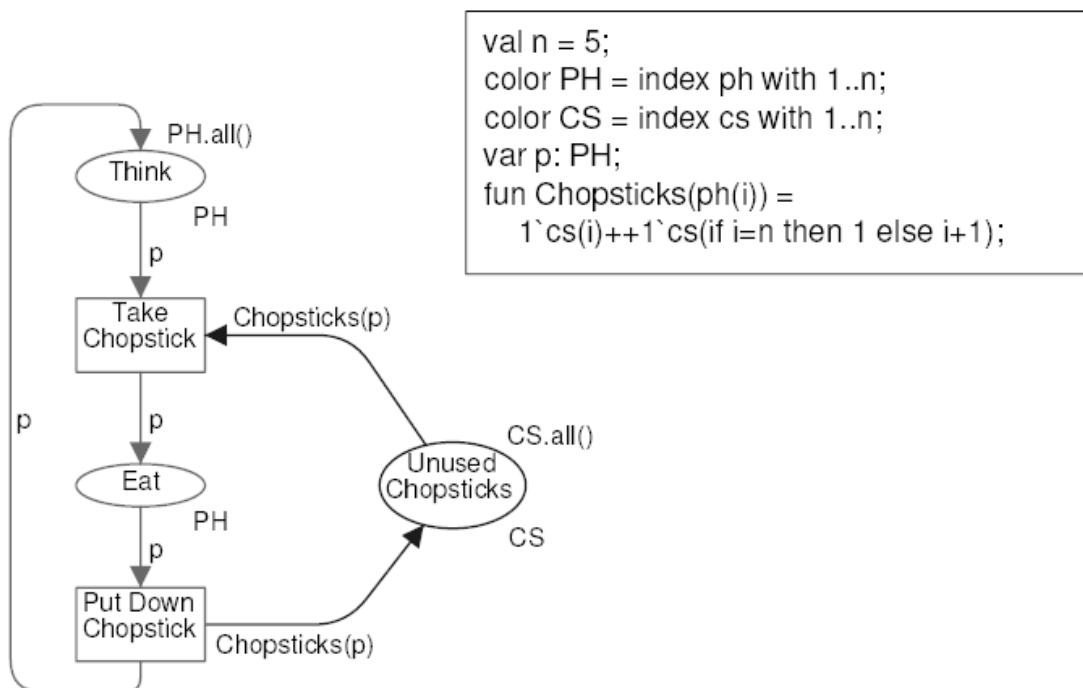
Σε αυτήν την παράγραφο θα χρησιμοποιήσουμε το πρόβλημα του **Δείπνου των Φιλοσόφων (Dining Philosophers)**, για να δώσουμε ένα παράδειγμα χρήσης του προγράμματος CPN Tools. Θα περιγράψουμε το γράφο εκτέλεσης με λεπτομέρεια και θα δούμε ένα μεγάλο αριθμό από διαφορετικά ερωτήματα. Το μοντέλο του Χρωματισμένου δικτύου Petri περιγράφει, πώς ένας αριθμός από διεργασίες (οι φιλόσοφοι) μοιράζονται κοινούς πόρους (τα ξυλάκια φαγητού - chopsticks). Το Δείπνο των φιλοσόφων είναι ένα από τα κλασικά παραδείγματα που χρησιμοποιούνται από τους επιστήμονες, για να απεικονίσουν νέες ιδέες σχετικά με το συγχρονισμό και τον ταυτοχρονισμό.

Πέντε Κινέζοι φιλόσοφοι κάθονται γύρω από ένα κυκλικό τραπέζι. Στη μέση του τραπεζιού υπάρχει ένα νοστιμότατο πιάτο ρύζι, και μεταξύ κάθε ζεύγους φιλοσόφων υπάρχει μόνο ένα ξυλάκι φαγητού. Κάθε φιλόσοφος είτε σκέφτεται, είτε τρώει. Για να φάει ένας φιλόσοφος, χρειάζεται δύο ξυλάκια φαγητού, κι επιτρέπεται να χρησιμοποιήσει μόνο τα δύο ξυλάκια που βρίσκονται δίπλα του, το ένα από την αριστερή και το άλλο από τη δεξιά μεριά. Επειδή τα ξυλάκια φαγητού πρέπει να μοιραστούν, δύο φιλόσοφοι που κάθονται ο ένας δίπλα στον άλλον, δεν μπορούν να φάνε ταυτόχρονα. Στην Εικόνα 59 απεικονίζεται το δείπνο των φιλοσόφων.



Εικόνα 59: Το Δείπνο των Φιλοσόφων

Το Χρωματισμένο δίκτυο Petri που μοντελοποιεί το σύστημα των φιλοσόφων και οι αντίστοιχες δηλώσεις φαίνονται στην Εικόνα 60. Το σύνολο χρωμάτων PH παριστάνει τους φιλοσόφους, ενώ το σύνολο χρωμάτων CS παριστάνει τα ξυλάκια φαγητού. Η συνάρτηση Chopsticks αντιστοιχεί σε κάθε φιλόσοφο, τα δύο ξυλάκια φαγητού που βρίσκονται δίπλα του.



Εικόνα 60: Το Χρωματισμένο δίκτυο Petri που μοντελοποιεί το σύστημα των φιλοσόφων και οι δηλώσεις του δικτύου

Στην Εικόνα 61 φαίνεται ένας γράφος εκτέλεσης του Δείπνου των Φιλοσόφων. Η τρέχουσα έκδοση του CPN Tools δεν μπορεί να σχεδιάσει γράφους εκτέλεσης. Κάθε κόμβος αναπαριστά ένα προσιτό μαρκάρισμα, ενώ κάθε τόξο αναπαριστά την εκτέλεση ενός στοιχείου καταχώρησης, το οποίο οδηγεί από το μαρκάρισμα του κόμβου από τον οποίο ξεκινάει το τόξο, στο μαρκάρισμα του κόμβου στον οποίο καταλήξει το τόξο. Για να βελτιώσουμε την αναγνωσιμότητα, δείχνουμε μόνο τα περιεχόμενα ορισμένων μαρκαρισμάτων και ορισμένων στοιχείων καταχώρησης. Όλα τα τόξα του γράφου εκτέλεσης είναι διπλά τόξα, δηλαδή αναπαριστούν δύο ξεχωριστά τόξα.

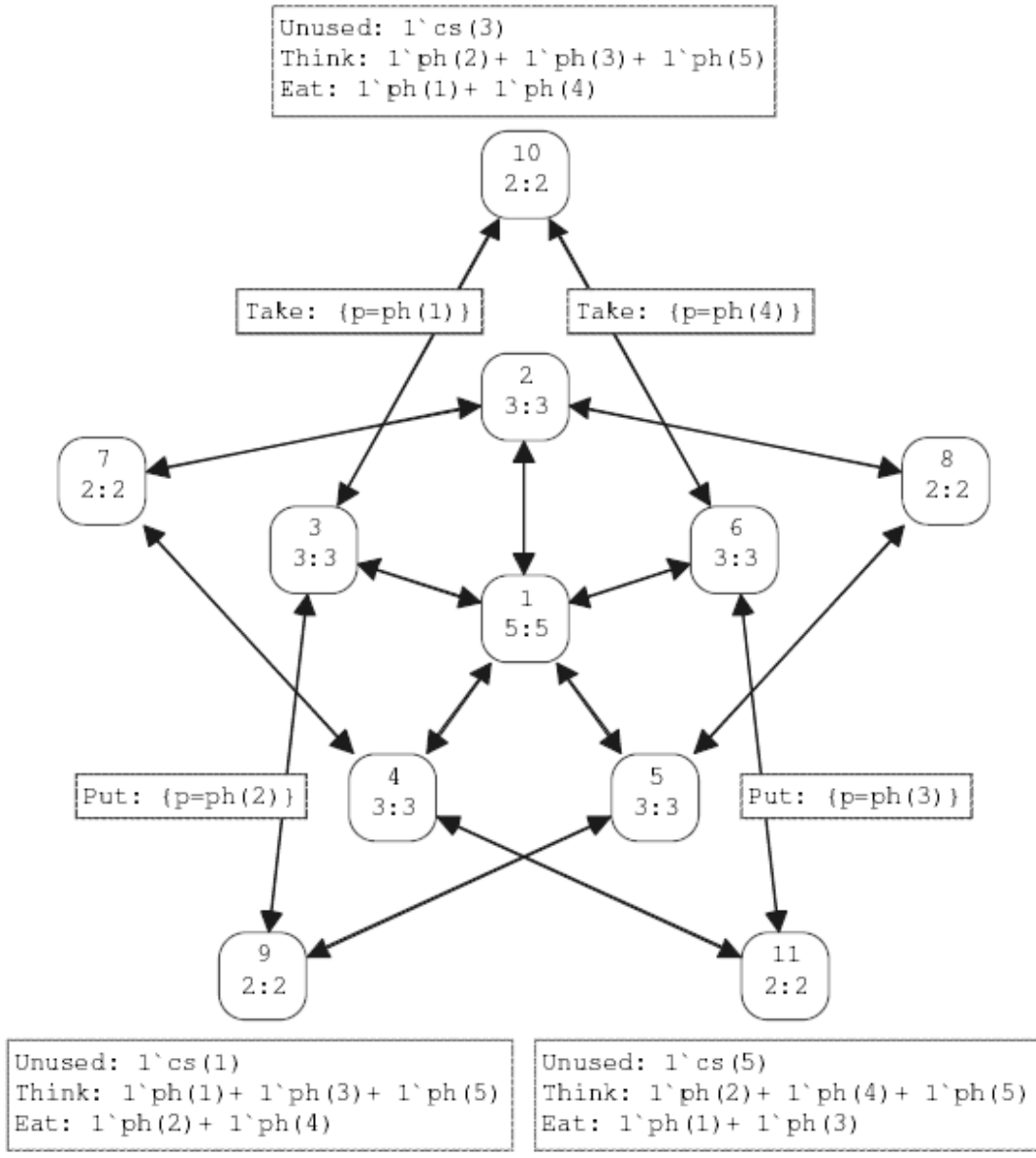
Μια τυπική αναφορά προσομοίωσης του Δείπνου των Φιλοσόφων φαίνεται στην Εικόνα 62. Για να βελτιώσουμε τηνη αναγνωσιμότητα των ορίων πολλαπλών συνόλων (multi-set bounds), αντικαταστήσαμε την έκφραση :

$$1`ph(1) + 1`ph(2) + 1`ph(3) + 1`ph(4) + 1`ph(5)$$

με PH, και την έκφραση :

$$1`cs(1) + 1`cs(2) + 1`cs(3) + 1`cs(4) + 1`cs(5)$$

με CS.



Εικόνα 61: Ένας γράφος εκτέλεσης του Δείπνου των Φιλοσόφων

Statistics

Occurrence Graph

Nodes: 11
Arcs: 30
Secs: 1
Status: Full

Scc Graph

Nodes: 1
Arcs: 0
Secs: 0

Boundedness Properties

Best Integer Bounds

	Upper	Lower
Eat	2	0
Think	5	3
Unused	5	1

Best Upper Multi-set Bounds

Eat	PH
Think	PH
Unused	CS

Best Lower Multi-set Bounds

Eat	empty
Think	empty
Unused	empty

Home Properties

Home Markings: All

Liveness Properties

Dead Markings: None
Dead Transitions Instances: None
Live Transitions Instances: All

Fairness Properties	
Put	Impartial
Take	Impartial

Εικόνα 62: Μια αναφορά προσομοίωσης του Δείπνου των Φιλοσόφων

Στη συνέχεια θα περιγράψουμε κάποιες συναρτήσεις που χρησιμοποιούν τα ερωτήματα της γλώσσας CPN ML, για να εξετάσουν τις ιδιότητες ενός Χρωματισμένου δικτύου Petri.

Reachable: καθορίζει αν υπάρχει μια **ακολουθία εκτέλεσης (occurrence sequence)** που οδηγεί από το μαρκάρισμα του πρώτου κόμβου, στο μαρκάρισμα του δεύτερου. Για το Δείπνο των Φιλοσόφων η συνάρτηση :

Reachable (5, 3)

είναι **αληθής (true)**. Αυτό σημαίνει ότι υπάρχει μια ακολουθία εκτέλεσης που οδηγεί από το μαρκάρισμα M5 (του κόμβου 5), στο μαρκάρισμα M3 (του κόμβου 3). Η συνάρτηση έχει και άλλη μία έκδοση :

Reachable' (5, 3)

η οποία επιστρέφει το ίδιο αποτέλεσμα με την εξήγηση :

"A path from node 5 to node 3 is: [5, 9, 3]"

που μας πληροφορεί ότι υπάρχει μια ακολουθία εκτέλεσης που περιέχει τα μαρκάρια M5, M9, M3 (με αυτή τη σειρά). Το μονοπάτι έχει ελάχιστο μέγεθος.

SccReachable: επιστρέφει το ίδιο αποτέλεσμα με τη συνάρτησης Reachable, αλλά χρησιμοποιεί το SCC-γράφο, δηλαδή τα ισχυρά συνδεδεμένα τμήματα. Αυτό σημαίνει ότι είναι πιο γρήγορη από τη Reachable, τουλάχιστον για τους γράφους εκτέλεσης που περιέχουν κύκλους. Η συνάρτηση έχει και άλλη μία έκδοση :

SccReachable' (5, 3)

η οποία επιστρέφει το ίδιο αποτέλεσμα μαζί με την εξήγηση :

"A path from the SCC of node 5 to the
SCC of node 3 is: [~1]"

που μας πληροφορεί ότι και το μαρκάρισμα M5 και το μαρκάρισμα M3 ανήκουν στο ισχυρά συνδεδεμένο τμήμα ~1, δηλαδή στο ισχυρά συνδεδεμένο τμήμα του αρχικού μαρκαρίσματος.

AllReachable: καθορίζει αν όλα τα προσिता μαρκαρίσματα είναι προσिता από κάθε άλλο. Αυτό ισχύει αν υπάρχει μόνο ένα ισχυρά συνδεδεμένο τμήμα, οπότε και η συνάρτηση :

AllReachable ()

είναι αληθής.

UpperInteger: χρησιμοποιεί μια συγκεκριμένη συνάρτηση F του τύπου :

Node -> 'a ms

που υπολογίζει έναν ακέραιο αριθμό |F(n)|. Αυτό γίνεται για κάθε κόμβο n του γράφου εκτέλεσης, και επιστρέφεται ο μεγαλύτερος από τους ακέραιους αριθμούς που έχει υπολογιστεί. Το ερώτημα :

UpperInteger (Mark.System' Eat 1)

υπολογίζει το μέγιστο αριθμό των μαρκών στη θέση Eat στο πρώτο στιγμιότυπο της σελίδας System. Το αποτέλεσμα είναι 2 και υποδεικνύει ότι το πολύ δύο φιλόσοφοι μπορούν να φάνε ταυτόχρονα.

LowerInteger: είναι ανάλογο με το UpperInteger, αλλά επιστρέφει το μικρότερο από τους ακέραιους αριθμούς |F(n)|. Το ερώτημα :

LowerInteger (Mark.System' Think 1)

υπολογίζει τον ελάχιστο αριθμό των μαρκών στη θέση Think στο πρώτο στιγμιότυπο της σελίδας System. Το αποτέλεσμα είναι 3 και υποδεικνύει ότι υπάρχουν πάντα τουλάχιστον τρεις φιλόσοφοι που σκέφτονται.

UpperMultiset: είναι ανάλογο με το UpperInteger, αλλά αντί για το F(n) υπολογίζει το &F(n)&. Το αποτέλεσμα είναι το μικρότερο πολλαπλό σύνολο που είναι μεγαλύτερο ή ίσο από όλα τα πολλαπλά σύνολα που έχουν υπολογιστεί. Το ερώτημα:

UpperMultiSet (Mark.System' Eat 1)

επιστρέφει :

[ph 1, ph 2, ph 3, ph 4, ph 5] : PH ms

που είναι η αναπαράσταση ML για το πολλαπλό σύνολο :

$1^{\text{ph}}(1) + 1^{\text{ph}}(2) + 1^{\text{ph}}(3) + 1^{\text{ph}}(4) + 1^{\text{ph}}(5)$

Αυτό σημαίνει ότι κάθε ένας από τους πέντε φιλοσόφους μπορεί να φάει. Για να πάρουμε τη δεύτερη, πιο αναγνώσιμη μορφή του αποτελέσματος, χρησιμοποιούμε τον ακόλουθο ML κώδικα :

```
mkst_ms'PH (UpperMultiset (Mark.System'Eat 1))
```

LowerMultiset: είναι ανάλογο με το UpperInteger, αλλά επιστρέφει το μεγαλύτερο πολλαπλό σύνολο που είναι μικρότερο ή ίσο από όλα τα πολλαπλά σύνολα που έχουν υπολογιστεί. Το ερώτημα

```
LowerMultiset (Mark.System'Eat 1)
```

επιστρέφει το κενό πολλαπλό σύνολο, το οποίο αναπαρίσταται σαν μια άδεια λίστα. Αυτό σημαίνει ότι κάθε ένας από τους πέντε φιλοσόφους μπορεί να σκεφτεί γιατί υπάρχει ένα μαρκάρισμα στο οποίο ο φιλόσοφος δεν τρώει.

HomeSpace: καθορίζει αν το σύνολο των μαρκαρισμάτων που προσδιορίζεται στη λίστα των κόμβων είναι ένας **οικείος χώρος (home space)**, δηλαδή καθορίζει αν, από κάθε προσιτό μαρκάρισμα, είναι δυνατό να μεταβούμε σε τουλάχιστον ένα από τα μαρκάρια. Για το Δείπνο των Φιλοσόφων η συνάρτηση:

```
HomeSpace [2, 6]
```

είναι αληθής. Η συνάρτηση έχει επίσης και μια έκδοση που επιστρέφει το αποτέλεσμα μαζί με μια εξήγηση.

MinimalHomeSpace: επιστρέφει το μικρότερο αριθμό μαρκαρισμάτων που χρειάζονται για να σχηματίσουμε ένα οικείο χώρο. Αυτός ο αριθμός είναι ίδιος με τον αριθμό των τερματικών ισχυρά συνδεδεμένων τμημάτων. Για το Δείπνο των Φιλοσόφων το ερώτημα :

```
MinimalHomeSpace ()
```

επιστρέφει 1.

HomeMarking: καθορίζει αν το μαρκάρισμα του συγκεκριμένου κόμβου είναι ένα οικείο μαρκάρισμα, δηλαδή αν είναι προσιτό από όλα τα προσιτά μαρκάρια. Αυτό ισχύει αν υπάρχει μόνο ένα τερματικό ισχυρά συνδεδεμένο τμήμα και το συγκεκριμένο μαρκάρισμα ανήκει στο τμήμα αυτό. Για το Δείπνο των Φιλοσόφων η συνάρτηση:

```
HomeMarking (6)
```

είναι αληθής. Η συνάρτηση έχει επίσης και μια έκδοση, που επιστρέφει το αποτέλεσμα μαζί με μια εξήγηση.

ListHomeMarkings: επιστρέφει μια λίστα με όλους τους κόμβους που είναι οικεία μαρκαρίσματα. Για το Δείπνο των Φιλοσόφων το ερώτημα:

ListHomeMarkings ()

επιστρέφει μια λίστα που περιέχει και τους 11 κόμβους του γράφου εκτέλεσης.

ListHomeScc: είναι παρόμοιο με το ListHomeMarkings, αλλά το αποτέλεσμα δίνεται με ένα πιο συμμαζεμένο τρόπο. Το αποτέλεσμα είναι είτε ένα μόνο SCC και τότε τα οικεία μαρκαρίσματα είναι τα μαρκαρίσματα που ανήκουν στο SCC, είτε το αποτέλεσμα είναι μηδέν και δεν υπάρχουν οικεία μαρκαρίσματα. Για το Δείπνο των Φιλοσόφων το ερώτημα :

ListHomeScc ()

επιστρέφει ~1, δηλαδή το SCC στο οποίο ανήκει το αρχικό μαρκαρίσμα. Αυτό υποδεικνύει ότι όλα τα προσιτά μαρκαρίσματα είναι οικεία μαρκαρίσματα.

HomeMarkingExists: καθορίζει αν το Χρωματισμένο δίκτυο Petri έχει κάποιο οικείο μαρκαρίσμα, το οποίο ισχύει όταν υπάρχει ακριβώς ένα τερματικό ισχυρά συνδεδεμένο τμήμα. Για το Δείπνο των Φιλοσόφων η συνάρτηση:

HomeMarkingExists ()

είναι αληθής.

InitialHomeMarking: καθορίζει αν το αρχικό μαρκαρίσμα του γράφου εκτέλεσης είναι ένα οικείο μαρκαρίσμα, δηλαδή αν είναι προσιτό από όλα τα προσιτά μαρκαρίσματα. Αυτό ισχύει όταν υπάρχει ακριβώς ένα τερματικό ισχυρά συνδεδεμένο τμήμα. Για το Δείπνο των Φιλοσόφων το αποτέλεσμα αυτής της συνάρτησης είναι ίδιο με αυτό της συνάρτησης AllReachable, δηλαδή η συνάρτηση :

InitialHomeMarking ()

είναι αληθής.

DeadMarking: καθορίζει αν το μαρκαρίσμα του συγκεκριμένου κόμβου είναι νεκρό, δηλαδή δεν έχει ενεργοποιημένα στοιχεία καταχώρησης. Για το Δείπνο των Φιλοσόφων η συνάρτηση:

DeadMarking (8)

είναι ψευδής. Αυτό υποδεικνύει ότι το μαρκάρισμα M8 έχει κάποια ενεργοποιημένα στοιχεία καταχώρησης.

ListDeadMarkings: επιστρέφει μια λίστα με όλους τους κόμβους του γράφου εκτέλεσης που είναι νεκροί, δηλαδή δεν έχουν ενεργοποιημένα στοιχεία καταχώρησης. Για το Δείπνο των Φιλοσόφων το ερώτημα :

```
ListDeadMarkings ()
```

επιστρέφει την κενή λίστα.

SccListDeadMarkings: επιστρέφει το ίδιο αποτέλεσμα με τη συνάρτησης ListDeadMarkings, αλλά χρησιμοποιεί το SCC-γράφο, δηλαδή τα ισχυρά συνδεδεμένα τμήματα. Αυτό σημαίνει ότι είναι πιο γρήγορη από τη ListDeadMarkings, τουλάχιστον για τους γράφους εκτέλεσης που περιέχουν κύκλους.

TIsDead: καθορίζει αν το σύνολο των στιγμιότυπων των μεταβάσεων, που προσδιορίζεται στη λίστα, είναι νεκρό στο μαρκάρισμα του συγκεκριμένου κόμβου, δηλαδή αν είναι αδύνατο να βρεθεί μια ακολουθία εκτέλεσης που να αρχίζει στο συγκεκριμένο μαρκάρισμα και να περιέχει ένα από τα στιγμιότυπα μεταβάσεων που προσδιορίζονται. Για το Δείπνο των Φιλοσόφων η συνάρτηση:

```
TIsDead ([TI.System'Take 1], 4)
```

είναι ψευδής. Αυτό υποδεικνύει ότι υπάρχει μια ακολουθία εκτέλεσης που ξεκινάει στο μαρκάρισμα M4 και περιέχει μια εκτέλεση της μετάβασης Take στο πρώτο στιγμιότυπο της σελίδας System. Η συνάρτηση έχει επίσης και μια έκδοση που επιστρέφει το ίδιο αποτέλεσμα μαζί με την εξήγηση :

```
"A transition instance from the given list  
is contained in the SCC: ~1 (which is  
reachable from the SCC of the given node)"
```

BEsDead: είναι ανάλογο με το TIsDead, εκτός από το γεγονός ότι το όρισμα είναι μια λίστα στοιχείων καταχώρησης αντί για στιγμιότυπα μεταβάσεων. Για το Δείπνο των Φιλοσόφων η συνάρτηση:

```
BEsDead ([Bind.System'Take (1, {p=ph(3)})], 4)
```

είναι ψευδής. Αυτό υποδεικνύει ότι υπάρχει μια ακολουθία εκτέλεσης που ξεκινάει στο μαρκάρισμα M4 και περιέχει μια εκτέλεση της μετάβασης Take στο πρώτο στιγμιότυπο της σελίδας System, με τη μεταβλητή p να έχει τιμή ph(3). Η συνάρτηση έχει επίσης και μια έκδοση, που επιστρέφει το αποτέλεσμα μαζί με μια εξήγηση.

ListDeadTIs: επιστρέφει μία λίστα με όλα τα στιγμιότυπα μεταβάσεων που είναι νεκρά, δηλαδή δεν εμφανίζονται σε καμία ακολουθία εκτέλεσης που αρχίζει στο

αρχικό μαρκάρισμα του γράφου εκτέλεσης. Για το Δείπνο των Φιλοσόφων το ερώτημα :

ListDeadTIs ()

επιστρέφει την κενή λίστα.

TIsLive: καθορίζει αν το σύνολο των στιγμιότυπων των μεταβάσεων, που προσδιορίζεται στη λίστα, είναι διαρκές, δηλαδή αν, από κάθε προσιτό μαρκάρισμα, είναι δυνατό να βρεθεί μια ακολουθία εκτέλεσης που να περιέχει ένα από τα στιγμιότυπα μεταβάσεων που προσδιορίζονται. Για το Δείπνο των Φιλοσόφων η συνάρτηση:

TIsLive [TI.System'Take 1]

είναι αληθής. Αυτό υποδεικνύει ότι δεν υπάρχει μαρκάρισμα στο οποίο η μετάβαση Take στο πρώτο στιγμιότυπο της σελίδας System, να μην μπορεί να εκτελεστεί. Η συνάρτηση έχει επίσης και μια έκδοση, που επιστρέφει το αποτέλεσμα μαζί με μια εξήγηση.

BEsLive: είναι ανάλογο με το TIsLive, εκτός από το γεγονός ότι το όρισμα είναι μια λίστα στοιχείων καταχώρησης αντί για στιγμιότυπα μεταβάσεων. Για το Δείπνο των Φιλοσόφων η συνάρτηση:

BEsLive [Bind.System'Take (1, {p=ph(3)})]

είναι αληθής. Αυτό υποδεικνύει ότι ο φιλόσοφος ph(3) έχει πάντα μια πιθανότητα να πάρει (Take) τα ξυλάκια φαγητού του. Δεν μπορεί να το κάνει αυτό σε όλα τα προσιτά μαρκάρια, αλλά είναι πάντα δυνατό να επιλεγεί μια ακολουθία βημάτων, ώστε να συμβεί. Η συνάρτηση έχει επίσης και μια έκδοση, που επιστρέφει το αποτέλεσμα μαζί με μια εξήγηση.

BEsStrictlyLive: καθορίζει αν το σύνολο των στιγμιότυπων των μεταβάσεων, που προσδιορίζεται στη λίστα, είναι αυστηρά διαρκές, δηλαδή αν κάθε στιγμιότυπο μετάβασης στη λίστα είναι διαρκές. Για το Δείπνο των Φιλοσόφων η συνάρτηση:

```
BEsStrictlyLive [  
    Bind.System'Take (1, {p=ph(1)}),  
    Bind.System'Take (1, {p=ph(2)}),  
    Bind.System'Take (1, {p=ph(3)}),  
    Bind.System'Take (1, {p=ph(4)}),  
    Bind.System'Take (1, {p=ph(5)})]
```

είναι αληθής. Αυτό υποδεικνύει ότι κάθε φιλόσοφος έχει πάντα μια πιθανότητα να πάρει (Take) τα ξυλάκια φαγητού του. Δεν μπορεί να το κάνει αυτό σε όλα τα προσιτά μαρκάρια, αλλά είναι πάντα δυνατό να επιλεγεί μια ακολουθία βημάτων, ώστε να συμβεί.

ListLiveTIs: επιστρέφει μία λίστα με όλα τα στιγμιότυπα μεταβάσεων που είναι διαρκή. Για το Δείπνο των Φιλοσόφων το ερώτημα:

```
ListLiveTIs ()
```

επιστρέφει :

```
[System'Put 1, System'Take 1]
```

Αυτό υποδεικνύει ότι είναι αδύνατο να μεταβούμε σε ένα μαρκάρισμα τέτοιο ώστε ένα από τα στιγμιότυπα μεταβάσεων να μην μπορεί να εκτελεστεί.

TIsFairness: καθορίζει αν το σύνολο των στιγμιότυπων των μεταβάσεων, που προσδιορίζεται στη λίστα, είναι αμερόληπτο, δίκαιο ή ακριβοδίκαιο. Για το Δείπνο των Φιλοσόφων το ερώτημα:

```
TIsFairness [TI.System'Take 1]
```

επιστρέφει `Impartial`. Αυτό σημαίνει ότι δεν μπορούμε να έχουμε μια μη πεπερασμένη ακολουθία εκτέλεσης, εκτός και αν η μετάβαση `Take` στο πρώτο στιγμιότυπο της σελίδας `System` συνεχίζει να εκτελείται.

BEsFairness: είναι ανάλογο με το `TIsFairness`, εκτός από το γεγονός ότι το όρισμα είναι μια λίστα στοιχείων καταχώρησης αντί για στιγμιότυπα μεταβάσεων. Για το Δείπνο των Φιλοσόφων η συνάρτηση:

```
BEsFairness [Bind.System'Take (1, {p=ph(3)})]
```

επιστρέφει `No_Fairness`. Αυτό σημαίνει ότι είναι δυνατό να έχουμε μια μη πεπερασμένη ακολουθία εκτέλεσης (που αρχίζει από ένα προσιτό μαρκάρισμα) στην οποία ο φιλόσοφος `ph(3)` ποτέ δεν παίρνει ξυλάκια φαγητού.

ListImpartialTIs: επιστρέφει μία λίστα με όλα τα στιγμιότυπα μεταβάσεων που είναι αμερόληπτα. Για το Δείπνο των Φιλοσόφων το ερώτημα:

```
ListImpartialTIs ()
```

επιστρέφει τη λίστα :

```
[System'Put 1, System'Take 1]
```

Αυτό σημαίνει ότι όλες οι μη πεπερασμένες ακολουθίες εκτέλεσης (που αρχίζουν από το αρχικό μαρκάρισμα) περιέχουν ένα μη πεπερασμένο αριθμό και από τα δύο στιγμιότυπα μεταβάσεων.

ListFairTIs και ListJustTIs: είναι ανάλογα με το `ListImpartialTIs`, με τη διαφορά ότι επιστρέφουν αυτά τα στιγμιότυπα μεταβάσεων που είναι δίκαια και ακριβοδίκαια, αντίστοιχα. Η αμεροληψία υπονοεί τη δικαιοσύνη η οποία υπονοεί την

ακριβοδιακιοσύνη . Έτσι είναι προφανές ότι για το Δείπνο των Φιλοσόφων και τα δύο ερωτήματα :

```
ListFairTIs ()  
ListJustTIs ()
```

επιστρέφουν την ίδια λίστα :

```
[System'Put 1, System'Take 1]
```

Στη συνέχεια θα δούμε ερωτήματα που εξαρτώνται από το εκάστοτε μοντέλο, στη συγκεκριμένη περίπτωση δηλαδή από το μοντέλο του Δείπνου των Φιλοσόφων.

Όλοι οι κόμβοι στους οποίους ένας συγκεκριμένος φιλόσοφος τρώει μπορούν να υπολογιστούν με τον εξής τρόπο :

```
fun Eating (p:PH) : Node list  
    = PredAllNodes (fn n => cf(p,Mark.System'Eat 1  
n) > 0)
```

όπου η συνάρτηση cf επιστρέφει το συντελεστή του συγκεκριμένου χρώματος στο συγκεκριμένο πολλαπλό σύνολο.

Ο μέγιστος αριθμός των ταυτόχρονα ενεργοποιημένων στιγμιότυπων μεταβάσεων μπορεί να υπολογιστεί με τον εξής τρόπο :

```
fun MaxTIEnabled () : int  
    = SearchAllNodes(  
        fn _ => true,  
        fn n =>  
            List.length(remdupl(List.map  
                                ArcToTI(OutArcs n))),  
        0,  
        Int.max)
```

όπου η συνάρτηση remdupl απομακρύνει τις διπλές εμφανίσεις από μια λίστα, ενώ η συνάρτηση map χρησιμοποιεί τη συγκεκριμένη συνάρτηση σε όλα τα στοιχεία της συγκεκριμένης λίστας.

Για να εξετάσουμε αν υπάρχουν προσιτά μαρκαρίσματα στα οποία δύο γειτονικοί φιλόσοφοι τρώνε ταυτόχρονα, γράφουμε την εξής συνάρτηση, όπου next είναι μια συνάρτηση που αντιστοιχεί κάθε φιλόσοφο στο διάδοχο του, ext_col είναι μια συνάρτηση που επεκτείνει τη συνάρτηση 'a -> 'b στη συνάρτηση 'a ms -> 'b ms, ενώ <=< είναι ο τελεστής μικρότερο-ίσο στα πολλαπλά σύνολα :

```

fun next (ph i: PH) : PH
  = if i<n then ph(i+1) else ph 1;
fun EatingNeighbours () : Node list
  = PredAllNodes(fn n =>
    let
      val Eating = Mark.System'Eat 1 n
    in
      not(Eating + ext_col next Eating <== PH)
    end)

```

Η παρακάτω συνάρτηση επιστρέφει όλα τα τόξα στα οποία η μετάβαση Take εκτελείται στο πρώτο στιγμιότυπο της σελίδας System με τη μεταβλητή p να παριστάνει ένα συγκεκριμένο φιλόσοφο.

```

fun TakeChopsticks (p:PH) : Arc list
  = PredAllArcs(fn a =>
    case ArcToBE a
    of Bind.System'Take (1, {p=pí}) => p=p'
    | _ => false)

```

Για το σύστημα του Δείπνου των φιλοσόφων ο γράφος εκτέλεσης αναπτύσσεται σχετικά αργά, όταν αυξάνουμε τον αριθμό των φιλοσόφων, όπως φαίνεται στην Εικόνα 63.

PH	Nodes	Arcs
2	3	4
3	4	6
4	7	16
5	11	30
6	18	60
7	29	112
8	47	208
9	76	378
10	123	680
15	1,364	11,310

Εικόνα 63: Το μέγεθος του γράφου εκτέλεσης σε σχέση με τον αριθμό των φιλοσόφων

Βιβλιογραφία - Αναφορές

1. A. H. Lewis: "An Introduction to Petri Nets", v.3.3, 23/11/1999.
2. Ξ. Γωγουβίτης: "Εισαγωγή στα δίκτυα Petri (Petri-nets, PNs)", Διπλωματική Εργασία, επιβλέπων Γ. Βοσνιάκος, 2002.
3. Β. Χ. Γερογιάννης: "Ενοποιημένο Τυπικό Μοντέλο για την Αναπαράσταση της Λειτουργικότητας και την Ανάλυση της Χρονοδρομολογησιμότητας Εφαρμογών Πραγματικού Χρόνου", Διδακτορική Διατριβή, επιβλέπων καθηγητής Π. Πιντέλας, Τμήμα Μαθηματικών, Πανεπιστήμιο Πατρών, Μάιος 2001.
4. G. Eirea: "Petri Nets: Properties, Analysis and Applications", UC Berkeley, August 2002.
5. Π. Λιναρδής: "Προηγμένες Αρχιτεκτονικές Υπολογιστικών Συστημάτων", Σημειώσεις, Τμήμα Πληροφορικής, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Θεσσαλονίκη 2003.
6. G. Balbo: "An Introduction to Generalized Stochastic Petri Nets", Introductory Tutorial Petri Nets, 21st International Conference of Application and Theory of Petri Nets, Aarhus, Denmark, June 2000.
7. G. Balbo: "Introduction to Stochastic Petri Nets", Dipartimento di Informatica, Università di Torino, Torino, Italy.
8. Ε. Καρατζά: "Απόδοση Υπολογιστικών Συστημάτων", Σημειώσεις, Τμήμα Πληροφορικής, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Θεσσαλονίκη 2002.
9. N. Chapman: "Petri Net Models", ISE-2 Surprise 97 Project, Imperial College London
10. M. A. Marsan, A. Bobbio, S. Donatelli: "Petri Nets in Performance Analysis: An Introduction", Lectures on Petri Nets I: Basic Models, Springer-Verlag, Berlin, 1998.
11. K. Jensen: "An Introduction to the Practical Use of Coloured Petri Nets", Lectures on Petri Nets II: Applications, Springer-Verlag, Berlin, 1998.
12. <http://inf-server.inf.uth.gr/courseware/CE636/>
13. <http://www.cse.fau.edu/~maria/COURSES/CEN4010-SE/C10/10-7.html>
14. http://ls4-www.informatik.uni-dortmund.de/QPN/QPN_article/qpn_final/qpn_final.html
15. <http://www.daimi.au.dk/CPnets>
16. <http://www.daimi.au.dk/PetriNets>
17. <http://wiki.daimi.au.dk/cpntools/cpntools.wiki>