

Ατομική Διπλωματική Εργασία

**ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΓΙΑ ΠΡΟΣΟΜΟΙΩΣΗ ΚΑΙ ΕΛΕΓΧΟ
ΙΚΑΝΟΠΟΙΗΣΗΣ ΠΟΛΙΤΙΚΩΝ ΙΔΙΩΤΙΚΟΤΗΤΑΣ ΣΤΟ π GP-CALCULUS**

Χρίστος Πιτσιλλίδης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2014

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Τίτλος Ατομικής Διπλωματικής Εργασίας

Χρίστος Πιτσιλλίδης

Επιβλέπουσα Καθηγήτρια

Άννα Φιλίππου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2014

Ευχαριστίες

Καταρχήν θέλω να ευχαριστήσω την επιβλέπον καθηγήτριά μου, Δρ. Άννα Φιλίππου, η οποία καθ' όλη την διάρκεια εκπόνησης της διπλωματικής μου εργασίας παρείχε συνεχή υποστήριξη βοηθώντας με να εμπλουτίσω τις γνώσεις καθώς και την κρίση μου, γύρω από τον τομέα της έρευνας.

Επιπλέον, θα ήθελα να ευχαριστήσω τους γονείς μου, για την υποστήριξη που μου παρείχαν όλα τα χρόνια των σπουδών μου, τα οποία συντέλεσαν στην δημιουργία αυτής της ατομικής διπλωματικής εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω τον συμφοιτητή μου Κωνσταντίνο Ηροδότου, για την υποστήριξη και την βοήθεια που μου έδωσε στην αρχή της δημιουργίας του συστήματος το οποίο υλοποιήθηκε.

Περίληψη

Ζούμε σε ένα κόσμο ο οποίος είναι συνεχώς αναπτυσσόμενος. Αυτή η ανάπτυξη είναι περισσότερο εμφανής στον τομέα της τεχνολογίας. Η τεχνολογία, όμως, αναπτυσσόμενη διασφαλίζει περισσότερες ανέσεις, προσφέρει περισσότερες και καλύτερες υπηρεσίες, αυξάνει όμως και τα ρίσκα τα οποία παίρνει το κάθε άτομο το οποίο θέλει να χρησιμοποιήσει τις υπηρεσίες αυτές. Οι άνθρωποι όμως είναι διατεθειμένοι να δώσουν προσωπικές πληροφορίες με σκοπό να τους παρατεθούν αυτές οι υπηρεσίες, οι οποίες όμως πληροφορίες, τις περισσότερες φορές, αφορούν προσωπικά τους δεδομένα τα οποία θέλουν να είναι σίγουροι ότι δεν θα βρεθούν σε χέρια τα οποία οι ίδιοι δεν εξουσιοδότησαν.

Μέχρι σήμερα δεν υπάρχει τρόπος να πει ένα σύστημα με ακρίβεια αν διατηρείται η ιδιωτικότητα μέσα σε ένα άλλο δεδομένο σύστημα. Σκοπός της εργασίας αυτής ήταν η μελέτη διαφόρων αλγεβρών διεργασιών και η ανάπτυξη ενός συστήματος το οποίο αποσκοπεί στην αυτοματοποίηση της λειτουργίας αυτής. Έτσι, για να λύσουμε αυτό το πρόβλημα, θέσαμε ως στόχο την δημιουργία ενός συστήματος, το οποίο θα υπακούει στην άλγεβρα διεργασιών πGP-Calculus. Αυτό το σύστημα θα δέχεται ένα άλλο πρόγραμμα γραμμένο σε πGP-Calculus και θα εκτελεί 2 λειτουργίες. Είτε θα εκτελεί προσομοίωση των κανόνων μετάβασης, είτε θα εκτελεί έλεγχο των τύπων. Με τον έλεγχο των τύπων μπορούμε να διαπιστώσουμε κατά πόσο το σύστημα διαφυλάσσει ή όχι την ιδιωτικότητα ως προς τους τομείς της διάδοσης πληροφοριών, επεξεργασίας πληροφοριών και συλλογής πληροφοριών.

Περιεχόμενα

Κεφάλαιο 1	1
1.1 Εισαγωγικά	1
1.2 Στόχος	2
1.3 Μεθοδολογία.....	3
1.4 Δομή της εργασίας:	4
Κεφάλαιο 2	5
2.1 Ιδιωτικότητα	5
2.1.1 Ιδιωτικότητα και Παραβιάσεις.....	5
2.1.2 Ιδιωτικότητα και Τεχνολογία.....	7
2.1.3 Ιδιωτικότητα και Τυπικές Μέθοδοι	8
2.2 π -Calculus	10
2.2.1 Σημασιολογία.....	13
2.2.2 Παραδείγματα	14
2.3 π -Calculus with Groups	16
Κεφάλαιο 3	20
3.1 Γραμματική και Ισοδυναμίες	22
3.1.1 Παραδείγματα	24
3.2 Υποτύποι.....	25
3.2.1 Παραδείγματα	26
3.3 Τύποι.....	27
3.3.1 Σύστημα Τύπων	27
3.3.2 Έλεγχος τύπων.....	28
3.3.3 Πολιτική.....	31
3.3.4 Παραδείγματα:	33
Κεφάλαιο 4	36
4.1 Εισαγωγικά	36
4.2 Αρχεία.....	38
4.2.1 Αρχείο Ομάδων.....	38
4.2.2 Αρχείο Πολιτικών	40
4.2.3 Αρχείο Συστήματος	42
4.3 Εργαλεία	43
4.3.1 Εργαλείο Flex	43

4.3.2 Εργαλείο Bison	44
4.4 Υλοποίηση	45
4.4.1 Αντιστοίχιση Συμβόλων	45
4.4.2 Διαδικασία Δημιουργίας Δυναδικού Δέντρου Συστήματος	45
4.4.3 Διαδικασία προσομοίωσης	47
4.4.4 Διαδικασία Ελέγχου Τύπων	50
4.4.5 Στιγμιότυπα Οθόνης.....	51
Κεφάλαιο 5	57
5.1 Συμπεράσματα	57
5.2 Μελλοντικές Εργασίες.....	58
Βιβλιογραφία	59

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγικά	1
1.2 Στόχος	2
1.3 Μεθοδολογία	3
1.4 Δομή της Εργασίας	4

1.1 Εισαγωγικά

Το έγγραφο αυτό αποτελεί την αναφορά για την Ατομική Διπλωματική μου Εργασία. Στα πλαίσια αυτής μελέτησα Άλγεβρες Διεργασιών όπως την CCS και π -Calculus, με σκοπό την ανάπτυξη ενός προγράμματος το οποίο θα προσομοιώνει τους κανόνες μιας συγκεκριμένης Άλγεβρας Διεργασιών, την π GP-Calculus.

Η δημιουργία αυτής της Άλγεβρας Διεργασιών έγινε για σκοπούς διασφάλισης της ιδιωτικότητας. Αυτό επιτυγχάνεται με την χρήση πολιτικών και τύπων. Το σύστημα το οποίο θέλουμε να ελέγξουμε, το γράφουμε σε μορφή π GP-Calculus και δηλώνουμε την πολιτική του, δηλαδή αναφέρουμε τις ομάδες οι οποίες το απαρτίζουν καθώς και τα δικαιώματα που έχει η κάθε ομάδα και ονομάζουμε το σύνολο αυτών των δικαιωμάτων ως πολιτική. Στην συνέχεια, μέσω αυτών των πληροφοριών, αποφασίζουμε κατά πόσο οι ομάδες χρησιμοποιούν σωστά τα δεδομένα τα οποία τους δίνονται και δεν πέφτουν σε παραβιάσεις. Η ιδιωτικότητα είναι ένας όρος που, μέχρι σήμερα, μπορεί να σημαίνει κάτι

διαφορετικό για το κάθε άτομο. Αυτή η διαφοροποίηση οφείλεται στο υπόβαθρο του κάθε ανθρώπου και στο τι θεωρεί αυτός σημαντικό. Ακόμα υπάρχουν πολλές διαφορετικές σκοπιές να μελετήσει αυτόν τον όρο· φιλοσοφικές, νομικές, κοινωνικές και άλλες. Το μόνο σίγουρο είναι ότι οι νέες τεχνολογίες, οι οποίες βρίσκονται σε συνεχή ανάπτυξη στις μέρες μας, μπορούν να βοηθήσουν την διασφάλιση της, μπορούν όμως να θέσουν καινούργιες απειλές βρίσκοντας τρόπους παραβίασής της.

1.2 Στόχος

Την σήμερον ημέρα η επαφή με τον ηλεκτρονικό υπολογιστή για διεκπεραίωση εργασιών ή ακόμα και για διασκέδαση είναι πολύ κοινή και διαδεδομένη. Λόγω αυτής της ραγδαίας ανάπτυξης της τεχνολογίας, οι άνθρωποι με σκοπό να έχουν πρόσβαση σε διάφορες υπηρεσίες στο διαδίκτυο κυρίως, είναι πρόθυμοι να δώσουν προσωπικές του πληροφορίες δεδομένου ότι οι πληροφορίες αυτές δεν θα διαδοθούν πέρα από αυτήν την εταιρεία, οργανισμό ή ακόμα και άτομο. Εδώ είναι που εμφανίζεται το ερώτημα κατά πόσο είναι εφικτό να διαφυλάξουμε τις προσωπικές πληροφορίες του κάθε χρήστη δημιουργώντας ένα αυτοματοποιημένο τρόπο με τον οποίο να μπορούμε να ελέγχουμε τα συστήματα, με σκοπό τον προσδιορισμό των συστημάτων σε αυτά που διατηρούν την ιδιωτικότητα και σε αυτά που την απειλούν.

Γι' αυτό ένα από τα πιο σημαντικά προβλήματα σήμερα που υπάρχουν είναι η διασφάλιση της ιδιωτικότητας. Μέχρι στιγμής δεν υπάρχει τρόπος να πούμε με σιγουριά, μέσω ενός αυτοματοποιημένου συστήματος, αν η ιδιωτικότητα διασφαλίζεται μέσα σε ένα σύστημα.

Έχοντας αυτά υπόψη θέσαμε ως κύριο στόχο της εργασίας αυτής να δημιουργήσουμε ένα αυτοματοποιημένο σύστημα το οποίο να μπορεί να κάνει τους απαραίτητους υπολογισμούς έτσι ώστε να μπορεί να αποφασισθεί αν ένα σύστημα τηρεί τους κανόνες ιδιωτικότητας. Αυτό θα γίνει μέσω της ένταξης των απαιτήσεων σε ένα σύστημα τύπων και ακολούθως θα παίρνεται μια απόφαση μέσω του ελέγχου των τύπων αυτών. Το σημαντικό στοιχείο στο οποίο υπερτερεί το πGP -Calculus σε αντίθεση με άλλες άλγεβρες διεργασιών που αποσκοπούν στην διασφάλιση της ιδιωτικότητας, είναι το μοναδικό σύστημα τύπων το οποίο απαρτίζει την άλγεβρα αυτή. Μέσω αυτού του συστήματος, δηλώνεται το τι δικαιούται να κάνει και τι όχι η κάθε μια ομάδα που ανήκει στο σύστημα. Αυτές οι πληροφορίες είναι ο ορισμός της πολιτικής την οποία ακολουθεί το σύστημα. Για

παράδειγμα μπορούμε να μελετήσουμε το σύστημα ενός νοσοκομείου. Θέτουμε ότι οι ομάδες που αντιπροσωπεύουν το νοσοκομείο είναι οι γιατροί και οι νοσοκόμες, και τα δεδομένα που διακινούνται στο σύστημα του νοσοκομείου είναι οι πληροφορίες των ασθενών. Μια πολιτική θα ήταν οι νοσοκόμες να μην μπορούν να διαβάσουν και να γράψουν πάνω στις πληροφορίες των ασθενών αλλά μόνο να τις στείλουν, δηλαδή να τις μεταφέρουν εντός του συστήματος, και οι γιατροί θα μπορούν να επεξεργαστούν τις πληροφορίες, δηλαδή να γράψουν και να τις διαβάσουν αλλά να μην μπορούν να τις στείλουν. Άρα η διασφάλιση της ιδιωτικότητας γίνεται μέσω της διασφάλισης της σωστής λειτουργίας της πολιτικής.

1.3 Μεθοδολογία

Το σύστημα το οποίο αναπτύχθηκε για να εκτελεί τις εξής βασικές ενέργειες. Καταρχήν διαβάζει τα αρχεία τα οποία δίνονται ως είσοδος και κατασκευάζει δομές, τις οποίες θα χρησιμοποιήσει είτε για να γίνει έλεγχος τύπων, είτε για σκοπούς προσομοίωσης. Το σύστημα ακολουθεί το πρότυπο του π -calculus with groups και χρησιμοποιεί ένα σύστημα τύπων. Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την υλοποίηση είναι η C.

Για την δημιουργία του συστήματος χρησιμοποιήθηκαν τα εργαλεία Flex και Bison. Το Flex είναι ένας λεξικογραφικός αναλυτής και το Bison είναι ένας συντακτικός αναλυτής. Τα δύο αυτά εργαλεία συνεργάζονται με σκοπό να διαβαστούν τα αρχεία και να δημιουργηθούν οι απαραίτητες δομές.

Στο Flex δηλώνονται όλες οι κανονικές εκφράσεις οι οποίες θα πρέπει να αναγνωριστούν. Επίσης είναι δηλωμένες και οι δεσμευμένες λέξεις του συστήματος. Επιπλέον, αναγνωρίζονται και τα scores καθώς και οι παρενθέσεις. Κάθε φορά που αναγνωρίζεται μια κανονική έκφραση, αποστέλλεται στο Bison για να θέσει σε εφαρμογή ένα κανόνα γραμματικής. Κάθε φορά που ένας κανόνας αναγνωρίζεται πλήρως, εκτελείται ένα κομμάτι κώδικα ο οποίος αποθηκεύει πληροφορίες μέσα σε μια δομή οι οποίες κρατούν τα δεδομένα του κανόνα που αναγνωρίστηκε την δεδομένη στιγμή. Η δομή αυτή είναι ένα δυαδικό δέντρο στο οποίο κρατούμε το σύστημα έτσι ώστε να μπορούμε να το επεξεργαστούμε. Ο κάθε κόμβος έχει ένα τύπο ο οποίος υποδηλώνει το είδος των δεδομένων που κρατά καθώς και, ανάλογα με τον τύπο, πληροφορίες για το υπόδενδρο που ακολουθεί. Το σύστημα εκτελεί δύο κύριες δραστηριότητες μετά από το διάβασμα των

αρχείων, είτε θα εκτελέσει τις πράξεις μείωσης και θα δείξει την τροπή που θα πάρει το σύστημα και τις επικοινωνίες που θα γίνουν , είτε θα εκτελέσει έλεγχο τύπων για να δείξουμε κατά πόσο διασφαλίζεται η ιδιωτικότητα. Με σκοπό την εκτέλεση των ενεργειών χρειάζονται από τον χρήστη τρία αρχεία, το αρχείο του συστήματος, το αρχείο των ομάδων που απαρτίζουν το σύστημα καθώς και το αρχείο των πολιτικών. Για την πρώτη ενέργεια η οποία είναι η εκτέλεση των πράξεων μείωσης είναι αναγκαίο το αρχείο του συστήματος καθώς και το αρχείο των ομάδων, ενώ για την εκτέλεση του ελέγχου των τύπων χρειάζονται και τα τρία. Παρόλα αυτά, το σύστημα ζητά εξαρχής και τα τρία αρχεία.

1.4 Δομή της εργασίας:

Σε αυτήν την εργασία θα μελετήσουμε:

- Στο Κεφάλαιο 2 τις προηγούμενες εργασίες πάνω σε συγκεκριμένες άλγεβρες διεργασιών που συντέλεσαν στην ανάπτυξη της εργασίας αυτής.
- Στο Κεφάλαιο 3 την άλγεβρα διεργασιών που αναπτύχθηκε και πάνω στην οποία στηρίχθηκε το σύστημα που ανάπτυξα.
- Στο Κεφάλαιο 4 την υλοποίηση, από τα εργαλεία που χρησιμοποιήθηκαν μέχρι και την μεθοδολογία.
- Στο Κεφάλαιο 5 τα συμπεράσματα και τις μελλοντικές εργασίες που μπορούν να υπάρξουν.

Κεφάλαιο 2

Προηγούμενες Εργασίες

2.1 Ιδιωτικότητα	5
2.1.1 Ιδιωτικότητα και Παραβιάσεις	5
2.1.2 Ιδιωτικότητα και Τεχνολογία	7
2.1.3 Ιδιωτικότητα και Τυπικές Μέθοδοι	8
2.2 π -Calculus	10
2.2.1 Σημασιολογία	13
2.2.1 Παραδείγματα	14
2.3 π -Calculus with Groups	16

2.1 Ιδιωτικότητα

Πιο κάτω θα μελετήσουμε, σύμφωνα με την εργασία [1], τον όρο «ιδιωτικότητα» και τις πτυχές που έχει στην σημερινή κοινωνία της πληροφορίας.

2.1.1 Ιδιωτικότητα και Παραβιάσεις

Ο όρος «ιδιωτικότητα» έγινε γνωστός με την δημιουργία των φωτογραφικών τεχνολογιών και των τεχνολογιών εκτύπωσης. Αυτό συνέβηκε επειδή πλέον ο κόσμος είχε ένα μέσο μεταφοράς πληροφορίας με γρήγορο τρόπο. Τεράστια όμως σημασία απέκτησε με την γρήγορη και ραγδαία εξάπλωση του Διαδικτύου. Σήμερα, μέσω των διάφορων υπηρεσιών που προσφέρονται, των διάφορων κοινωνικών δικτύων που υπάρχουν, ακόμα και από πιο φρέσκες τεχνολογίες όπως το υπολογιστικό νέφος, είναι αναγκαίο να εμπιστευόμαστε πληροφορίες σε τρίτους, οι οποίες είναι εμπιστευτικές. Αυτό όμως συμβαίνει επειδή ο

κόσμος πλέον, είναι πρόθυμος να δώσει πληροφορίες που αφορούν τον ίδιο με σκοπό να του παρέχονται καλύτερες υπηρεσίες. Έτσι, πρέπει να διασφαλιστεί η ασφάλεια αυτών των πληροφοριών από εξωτερικές επιθέσεις, οι οποίες με τον χρόνο γίνονται περισσότερες και πιο αποδοτικές οι οποίες, σκόπιμα και μη, στοχεύουν να πάρουν προσωπικές πληροφορίες για άτομα τα οποία δεν τους εξουσιοδότησαν.

Από νομικής άποψης, σύμφωνα με τον Solove [2] υπάρχουν 4 κατηγορίες τέτοιων παραβιάσεων: η εισβολή, η συλλογή πληροφοριών, η επεξεργασία πληροφοριών και η διάδοση πληροφοριών.

Η εισβολή χωρίζεται σε 2 υποκατηγορίες, την φυσική εισβολή και την παρεμβολή αποφάσεων. *Φυσική εισβολή* διαπράττει ένα άτομο όταν εισβάλλει στον προσωπικό χώρο κάποιου άλλου ή στο ίδιο του το σώμα, δηλαδή όταν τον παρεμποδίζει να κάνει μια ενέργεια όπως να διασχίσει μια πόρτα. Η *παρεμβολή αποφάσεων* συμβαίνει όταν το άτομο που την κάνει επηρεάζει το θύμα της εισβολής αυτής στο να πάρει μια ελεύθερη απόφαση. Αυτό παρουσιάζεται σε διάφορες περιπτώσεις, όπως όταν ο θύτης κρατά προσωπικές πληροφορίες του θύματος τον εξαναγκάζει να πάρει αποφάσεις μέσω εκβιασμού αποκάλυψής τους.

Η συλλογή πληροφοριών χωρίζεται και αυτή σε δύο υποκατηγορίες, την *παρακολούθηση* και την *ανάκριση*. Ο τρόπος με τον οποίο παραβιάζεται η ιδιωτικότητα του ατόμου μέσω αυτών των μεθόδων δεν είναι τόσο η μέθοδος καθ' εαυτόν αλλά το πώς μπορούν να χρησιμοποιηθούν οι πληροφορίες που συλλέχτηκαν στην συνέχεια.

Η επεξεργασία δεδομένων διασπάται σε πολλές επιμέρους κατηγορίες. Παραβίαση της ιδιωτικότητας παρουσιάζεται όταν γίνεται *συνάθροιση δεδομένων*, όταν δηλαδή επεξεργαστούν δεδομένα τα οποία είναι προσβάλαμε από ένα άτομο, με τρόπο ο οποίος να παράγει δεδομένα τα οποία να μην είναι εξουσιοδοτημένος να έχει. Η *αναγνώριση* είναι η παραβίαση της ιδιωτικότητας που συμβαίνει όταν, μετά από επεξεργασία πληροφοριών, μπορεί κάποιος να συνδέσει ένα άτομο με ένα συμβάν ή μια κατάσταση, κάνοντας έτσι αυτήν την πληροφορία πιο διαθέσιμη από ότι θα έπρεπε να είναι. Επιπλέον υπάρχει η *ανασφάλεια* όπου πληροφορία είναι διαθέσιμη σε ομάδα ατόμων όπου δεν έπρεπε να είχαν δικαιώματα. Η *δευτερεύουσα χρήση* είναι η παραβίαση κατά την οποία πληροφορίες οι οποίες είναι διαθέσιμες, χρησιμοποιούνται για σκοπούς οι οποίοι δεν υπήρχε πρόθεση αρχικά. Τέλος ο *αποκλεισμός* υπάρχει όταν ένα άτομο δεν έχει την δυνατότητα να γνωρίζει ποιες πληροφορίες κρατιούνται, να τις δει και να τις επεξεργαστεί.

Η τελευταία κατηγορία είναι η διάδοση των δεδομένων. Αυτή με την σειρά της χωρίζεται σε υποκατηγορίες. Κατ' αρχήν υπάρχει η *παραβίαση της εμπιστευτικότητας* όπου ο εμπιστευόμενος κάτοχος πληροφοριών, ενός ή πολλών ατόμων, αποκαλύπτει τις πληροφορίες αυτές σε τρίτους το οποίο προκαλεί και ρήξη της εμπιστοσύνης του ατόμου προς τον κάτοχο των δεδομένων. Παρόμοια υπάρχει και η *αποκάλυψη* όπου ο κάτοχος των πληροφοριών μπορεί να μην είναι γνωστός στον χρήστη, σε αντίθεση με την παραβίαση της εμπιστευτικότητας, κάνει τις πληροφορίες αυτές προσβάσιμες σε άτομα τα οποία δεν έχουν την εξουσιοδότηση να τις δουν. Επιπλέον, η *έκθεση* είναι η αποκάλυψη πληροφοριών οι οποίες όμως είναι ντροπιαστικές για το άτομο στο οποίο αναφέρονται. Η *αλλοίωση* συμβαίνει όταν παρουσιάζονται ψευδής πληροφορίες για το υποκείμενο έχοντας ως αποτέλεσμα την λανθασμένη εικόνα προς το άτομό του, στρεβλώνοντας την εικόνα του καθώς και την γνώμη που έχουν τρίτα άτομα προς αυτόν. Η *αυξημένη πρόσβαση* είναι η παραβίαση κατά την οποία, πληροφορίες όπου εμπιστεύτηκαν σε ένα συγκεκριμένο άτομο, γίνονται πιο προσβάσιμες στον έξω κόσμο με πιθανό αποτέλεσμα να χρησιμοποιηθούν με τρόπο ο οποίος δεν συμφωνήθηκε μέσω της παραβίασης της *δευτερεύουσας χρήσης*. Τέλος, μια άλλη μορφή παραβίασης που μπορεί να κάνει ο κάτοχος δεδομένων, είναι να τα χρησιμοποιήσει με σκοπό τον *εκβιασμό* του ατόμου που αφορούν, εξαναγκάζοντάς τον να υποκύψει στις απαιτήσεις του.

2.1.2 Ιδιωτικότητα και Τεχνολογία

Η ανάπτυξη της τεχνολογίας έπαιξε σημαντικό ρόλο τόσο στην διασφάλιση, όσο και στην παραβίαση της ιδιωτικότητας. Απλές μηχανές οι οποίες δημιουργήθηκαν με σκοπό την ευκολία του ανθρώπου, όπως η φωτογραφική μηχανή, εύκολα μπορούν να καταχραστούν με τρόπο που να διαπράττουν παραβιάσεις ιδιωτικότητας άλλων ανθρώπων, όπως την παρακολούθηση.

Λόγω αυτής της ραγδαίας εξέλιξης έπρεπε να δημιουργηθούν νόμοι οι οποίοι να υποβλέπουν αυτές τις παρατυπίες και να τις τιμωρούν. Αυτοί οι νόμοι ξεκίνησαν από την δημιουργία των φωτογραφικών μηχανών και της τυπογραφίας και συνεχίζουν να αναπτύσσονται μέχρι και σήμερα. Παρακολούθησαν την ανάπτυξη της τεχνολογίας από την φωτοτυπία στις βιντεοκάμερες, στην μηχανές ανίχνευσης ακόμα και στην εναέρια παρακολούθηση. Σήμερα, με την τεχνολογία να δημιουργεί βάσεις δεδομένων τεράστιας

χωρητικότητας προσφέροντας τεράστιες ευκαιρίες για παραβιάσεις όπως την συνάθροιση δεδομένων, ανάγκασαν την δημιουργία επίτροπων με σκοπό την επίβλεψή τους.

Όμως παρ' όλες τις παραβιάσεις που μπορεί να δημιουργήσει, η τεχνολογία μπορεί και να βοηθήσει στην παρεμπόδισή τους. Τέθηκαν σε εφαρμογή διάφοροι κρυπτογραφικοί αλγόριθμοι με σκοπό την παρεμπόδιση μη εξουσιοδοτημένου προσωπικού στην είσοδο σε ιδιοκτησίες, ακόμα και σε συστήματα με σκοπό την ανάγνωση και επεξεργασία δεδομένων. Άλλες δημιουργίες της τεχνολογίας που προσπαθούν να βοηθήσουν την διαφύλαξη της ιδιωτικότητας είναι διάφορα συστήματα ανίχνευσης εισβολών ακόμα και διάφορα φίλτρα spam τα οποία προσπαθούν να διαφυλάξουν τον κάτοχό τους από ανεπιθύμητους εισβολείς.

Επιπλέον, εκτός από τις νέες τεχνολογίες, δημιουργήθηκαν και νέες μέθοδοι προστασίας. Με την χρήση της εξόρυξης δεδομένων, η οποία είναι εντός των επιτρεπτών πλαισίων της ιδιωτικότητας, μπορεί να ανιχνευτεί επικίνδυνη κίνηση στο διαδίκτυο ή ακόμα, μέσω των πληροφοριών που βρίσκονται σε διάφορες βάσεις δεδομένων στον κόσμο, να αποτραπεί εγκληματική ενέργεια στον κόσμο. Όσο για τα δεδομένα που βρίσκονται σε εξυπηρετητές ανά το διαδίκτυο, μπορούν πλέον ο κάθε χρήστης να βρει τα δικά του προσωπικά δεδομένα και να τα κάνει μονίμως απρόσιτα, μέσω μιας μεθόδου η οποία αναπτύχθηκε από τους Geambasu et al και ονομάστηκε *εξαφάνιση δεδομένων*[3].

Συνοψίζοντας από τα πιο πάνω, η τεχνολογία με την ραγδαία ανάπτυξη της, παρόλα τα προβλήματα που μπορεί να προκάλεσε, προσπαθεί να βρει τρόπους να εμποδίσει αυτήν την παραβίαση της ιδιωτικότητας. Έτσι, με αυτόν τον σκοπό δημιουργούνται νέες τεχνολογίες και νέα πεδία έρευνας στην περιοχή της επιστήμης τα οποία έχουν ως κύριο στόχο την πλήρη ασφάλεια των πολιτών από τέτοιες παραβιάσεις.

2.1.3 Ιδιωτικότητα και Τυπικές Μέθοδοι

Η ιδιωτικότητα, όπως αναφέραμε και πιο πάνω, είναι ένα φλέγον ερευνητικό θέμα. Έτσι η δημιουργία τυπικών μεθόδων για να δοθεί ένας φορμαλισμός στα συστήματα έχει πάρει μεγάλη προσοχή. Αυτό συνέβηκε διότι υπάρχει ανάγκη για διατύπωση θεμελιωδών αρχών με σκοπό την αποδεδειγμένη λύση για το πρόβλημα αυτό. Τυπικές μέθοδοι χαρακτηρίζονται μια ομάδα από συγκεκριμένες τεχνικές οι οποίες είναι βασισμένες στα

μαθηματικά και σκοπό έχουν τον προσδιορισμό καθώς και την ανάπτυξη και τον έλεγχο διαφόρων συστημάτων καθώς και υλικού υπολογιστών. Η χρήση τυπικών μεθόδων για τον σχεδιασμό υλικού και λογισμικού συμβαίνει λόγω της μεγάλης συμβολής που μπορεί να επιφέρουν οι μέθοδοι αυτοί στην αξιοπιστία και στην ευρωστία του σχεδιασμού. Οι τυπικές μέθοδοι είναι η εφαρμογή ενός μεγάλου εύρους θεωρητικών αρχών της πληροφορικής τυπικές γλώσσες, σε θεωρίες αυτομάτων ακόμα και σε συστήματα τύπων με σκοπό την επαλήθευση των προδιαγραφών λογισμικού και υλικού. Έτσι, οι τυπικές μέθοδοι, μπορούν να προσφέρουν πολλά προς αυτήν την κατεύθυνση, την δημιουργία δηλαδή, ενός αυτοματοποιημένου τρόπου για έλεγχο της ιδιωτικότητας. Έχουν γίνει πολλές προσπάθειες με χρήση διαφόρων μοντέλων για να προσομοιωθεί η συμπεριφορά ενός συστήματος όσον αφορά την ιδιωτικότητα. Έγινε χρήση μοντέλων όπως λογικές, μηχανές καταστάσεων και άλγεβρες διεργασιών.

Η χρήση τυπικών μεθόδων για μοντελοποίηση είναι μια συνηθισμένη μέθοδος για να περιγράψουμε την σχέση που έχει ένα σύστημα με το περιβάλλον του καθώς και τις αλληλεπίδρασης που έχει με αυτό. Για την δημιουργία, όμως του περιβάλλοντος, χρειάζονται να γίνουν κάποιες υποθέσεις, πράγμα που δεν είναι τόσο εύκολο να γίνει στην περίπτωση της ιδιωτικότητας. Είναι πιο εύκολο να κάνεις υποθέσεις για το πότε ένας εξυπηρετητής θα τεθεί εκτός λειτουργίας, παρά πότε θα παραβιαστεί μια αρχή της ιδιωτικότητας σε ένα σύστημα. Επίσης, στην ιδιωτικότητα έχουμε τρεις ξεχωριστές ομάδες οντοτήτων, το σύστημα το οποίο κρατά τα δεδομένα, τον αντίπαλο ο οποίος είναι μέρος του περιβάλλοντος και το άτομο τον οποίο αφορούν τα δεδομένα. Στο μοντέλο που θα πρέπει να δημιουργηθεί δεν πρέπει να μοντελοποιηθούν μόνο οι σχέσεις μεταξύ περιβάλλοντος και συστήματος, αλλά και οι σχέσεις μεταξύ αυτών των τριών οντοτήτων.

Η χρήση λογικών είναι επίσης διαδεδομένος τρόπος μελέτης της ιδιωτικότητας. Το πρόβλημα όμως που υπάρχει είναι ότι δεν μπορεί να γίνει διάκριση μεταξύ του επιπέδου εμπιστευτικότητας που υπάρχει μεταξύ των δεδομένων, από, για παράδειγμα, πλήρως εμπιστευτικά έως ανοικτά για όλους. Γι' αυτό υπάρχει ανάγκη για μελέτες της λογικής για ανάπτυξης νέων λογικών οι οποίων να μπορούν να καθρεφτίσουν αυτή την διάκριση.

Όσον αφορά τις τυπικές γλώσσες πολιτικών είναι και αυτές μια μέθοδος ευρεία διαδεδομένη για τυποποίηση της ιδιωτικότητας. Οι κάτοχοι των δεδομένων γράφουν τις πολιτικές τους σε μια κωδικοποιημένη μορφή η οποία γίνονται προσπάθειες να γίνει αναγνώσιμη από μηχανές. Το πρόβλημα όμως με αυτές τις γλώσσες είναι, ότι μέχρι

στιγμής, υπάρχει έλλειψη στη σημασιολογία κάνοντας δύσκολη τη έκφραση. Έτσι έχουν γίνει πολλές διαφορετικές προσπάθειες για την δημιουργία τυπικών γλωσσών με πολιτικές που να ξεπερνούν τα πιο πάνω προβλήματα. Γι' αυτόν τον σκοπό απαιτούνται καινούριες προσεγγίσεις, αφού παρατηρούνται διαφορετικά μοντέλα και υποθέσεις. Έτσι, με αυτόν τον στόχο, έχουν αρχίσει να υλοποιούνται τρόποι σε διάφορες ερευνητικές προτάσεις. Στο υπόλοιπο του κεφαλαίου θα αναλυθούν οι άλγεβρες διεργασιών που συντέλεσαν την δημιουργία του πGP-Calculus όπου και βασίστηκε το σύστημα που υλοποιήθηκε.

2.2 π-Calculus

Το π-Calculus είναι μια άλγεβρα διεργασιών [4] η οποία βασίζεται στα *ονόματα* τα οποία είναι απεριόριστα και ανήκουν σε ένα σύνολο, έστω N . Τα ονόματα είναι κανάλια τα οποία μπορούν να χρησιμοποιηθούν για διάβασμα δεδομένων, δηλαδή είσοδο, για γράψιμο δεδομένων, δηλαδή έξοδο δεδομένων, ή ως αντικείμενο επικοινωνίας δηλαδή να σταλθούν μεταξύ οντοτήτων. Το μόνο είδος οντότητας που υπάρχει στο π-Calculus είναι η διεργασία. Η κάθε διεργασία είναι ένα σύνολο από ονόματα και η γραμματική που τις αποτελεί είναι η εξής:

$$P ::= x(y).P \mid \bar{x}(y).P \mid P|Q \mid !P \mid (vx)P \mid P+Q \mid 0$$

Θα συνεχίσουμε με την εξήγηση της πιο πάνω γραμματικής. Αρχικά έχουμε τον τελεστή $x(y).P$. Κατά αυτόν τον τελεστή, το όνομα y είναι δεμένο κάτω από το όνομα x . Έτσι γίνεται είσοδος του κάποιου ονόματος στο όνομα y στο συνδεδεμένο όνομα x . Όταν εκτελεστεί η είσοδος η εκτέλεση συνεχίζει σύμφωνα με τη διεργασία P . Μετά έχουμε τον τελεστή $\bar{x}(y).P$. Κατά αυτόν τον τελεστή, γίνεται αποστολή του ονόματος y μέσω του συνδεδεμένου ονόματος x . Όταν εκτελεστεί η έξοδος η εκτέλεση συνεχίζει σύμφωνα με την διεργασία P . Ο τελεστής $P|Q$, θέτει δύο διεργασίες, εδώ τις P και Q , να τρέξουν παράλληλα. Επιπλέον, ο τελεστής $!P$ αντιγράφει την διεργασία P , όσες φορές θέλει, την κάθε αντιγραφή παράλληλα με τις άλλες. Δηλαδή

$$!P \equiv P \mid P \mid P \mid \dots$$

Ο τελεστής $(vx)P$ περιορίζει ένα όνομα, εδώ το x , το οποίο μπορεί να δράσει μόνο εντός του P και δεν μπορεί να επικοινωνήσει με άλλες διεργασίες έξω από το P . Τέλος, τελευταίος

τελεστής είναι το 0. Το 0 σημαδεύει το τέλος της διεργασίας στην οποία ανήκει. Τις περισσότερες φορές παραλείπεται.

Επιπλέον, εκτός από τους τελεστές της γραμματικής, υπάρχει και ο δυαδικός τελεστής +. Στο π -Calculus ο τελεστής + αφορά μόνο διεργασίες. Δηλώνει ότι μπορεί να συμβεί μια, οποιαδήποτε, από τις 2. Δηλαδή, έστω $P + Q$, μπορεί να εκτελεστεί μόνο το P ή μόνο το Q .

Στον π -Calculus, υπάρχουν δύο σύνολα τα οποία ορίζουμε. Ορίζουμε το σύνολο $fn(P)$ ως το σύνολο στο οποίο ανήκουν όλα τα ονόματα τα οποία δεν είναι δεσμευμένα μέσα στην διεργασία P . Επιπλέον, ορίζουμε το σύνολο $bn(P)$ ως το σύνολο το οποίο περιέχει όλα τα ονόματα τα οποία είναι δεσμευμένα μέσα στην διεργασία P .

Για καλύτερη επεξήγηση των δύο συνόλων, έστω τα πιο κάτω παραδείγματα:

- $P \equiv x(y)$

Το σύνολο των δεσμευμένων ονομάτων είναι : $bn(P) = [3]$

Το σύνολο των ελεύθερων ονομάτων είναι: $fn(P) = [3]$

- $P \equiv \bar{x}(y)$

- Το σύνολο των δεσμευμένων ονομάτων είναι : $bn(P) = \emptyset$

- Το σύνολο των ελεύθερων ονομάτων είναι: $fn(P) = \{x,y\}$

Εκτός από τους πιο πάνω τελεστές, ισχύουν και κάποιες ισοδυναμίες, οι οποίες είναι πολύ χρήσιμες.

Κατ' αρχήν ισχύει ο κανόνας της ανάκλασης, δηλαδή μια διεργασία είναι ισοδύναμη με τον εαυτό της, γι' αυτό ισχύει ότι για μια διεργασία P :

$$P \equiv P$$

Ισχύει ο κανόνας της συμμετρίας. Δηλαδή, αν μια διεργασία P είναι ισοδύναμη με μια διεργασία Q , τότε και η διεργασία Q είναι ισοδύναμη με την διεργασία P .

$$P \equiv Q \Rightarrow Q \equiv P$$

Επιπρόσθετα ισχύει ο κανόνας μεταβατικότητας. Αν, δηλαδή, μια διεργασία P είναι ισοδύναμη με μια διεργασία Q και η διεργασία Q είναι ισοδύναμη με μια διεργασία R , τότε η διεργασία P είναι ισοδύναμη με την διεργασία R .

$$P \equiv Q, Q \equiv R \Rightarrow P \equiv R$$

Επιπλέον, υπάρχουν και κάποιες ισοδυναμίες οι οποίες αφορούν αποκλειστικά τον τελεστή «!». Αφού από το !P μπορείς να δημιουργήσεις άπειρα αντίγραφα της διεργασίας αυτής, τα οποία τρέχουν παράλληλα, τότε μπορείς να δημιουργήσεις ένα και να κρατήσεις το !P σε παράλληλη σύνδεση μαζί του. Έτσι λέμε ότι ισχύει το:

$$!P \equiv P \mid !P$$

Ακόμα, αν γνωρίζουμε ότι μια διεργασία P είναι ισοδύναμη με μια άλλη διεργασία Q, τότε το !P είναι ισοδύναμο με το !Q, παράγοντας την εξής ισοδυναμία:

$$P \equiv Q \Rightarrow !P \equiv !Q$$

Υπάρχουν και κανόνες, οι οποίοι αφορούν τον τελεστή «|». Κατ' αρχήν κατά τον τελεστή «|» ισχύει η αντιμεταθετική ιδιότητα. Αυτό συμβολίζεται ως εξής:

$$P \mid Q \equiv Q \mid P$$

Επίσης ορίζουμε τον κανόνα εισαγωγής παράλληλης σύνδεσης ως ακολούθως. Αν μια διεργασία P είναι ισοδύναμη με μια διεργασία Q, τότε αν συνδέσεις την P με την R μέσω παράλληλης σύνδεσης, το αποτέλεσμα είναι ισοδύναμο με την παράλληλη σύνδεση της Q σε παράλληλη σύνδεση με την ίδια διεργασία R.

$$P \equiv Q \Rightarrow P \mid R \equiv Q \mid R$$

Κατά τον τελεστή «|» ισχύει και η προσεταιριστική ιδιότητα.

$$(P \mid Q) \mid R \equiv P \mid (Q \mid R)$$

Ακολουθούν οι ισοδυναμίες για τον τελεστή (vx). Υπάρχει η ακόλουθη ισοδυναμία:

$$(vx)0 \equiv 0$$

Κατά αυτήν, το x είναι περιορισμένο μόνο μέσα στην διεργασία 0, η οποία είναι τερματική διεργασία και δεν εκτελεί τίποτα. Άρα η απαλοιφή του δεν επηρεάζει την φυσική ροή της διεργασίας.

Μια δεύτερη ισοδυναμία δηλώνει ότι:

$$x \neq y \Rightarrow (\forall x)(\forall y)P \equiv (\forall y)(\forall x)P$$

Το $(\forall x)(\forall y)P$ δηλώνει ότι τα x και y είναι περιορισμένα μέσα στην διεργασία P . Αν αλλάξει η σειρά τους, δεν εμποδίζει την κανονική λειτουργία της διεργασίας μιας και δηλώνει ακριβώς το ίδιο. Πρέπει όμως στο x να είναι διαφορετικό από το y διότι δεν θα ξέρουμε μετά, στις εμφανίσεις μέσα στο P για πιο όνομα αναφέρεται η εμφάνιση.

Τέλος μπορούμε να εξάγουμε μια διεργασία, έστω P , έξω από το πεδίο ενός $(\forall x)$, αν το x δεν ανήκει στο σύνολο των ελεύθερων ονομάτων της διεργασίας αυτής. Αυτό συμβολίζεται ως εξής:

$$\text{Αν } x \notin \text{fn}(P) \text{ τότε } (\forall x)(P|Q) \equiv P \mid (\forall x)Q$$

2.2.1 Σημασιολογία

Για την σωστή λειτουργία αυτής της Άλγεβρας Διεργασιών, είναι αναγκαίο να δημιουργηθούν κανόνες, οι οποίοι να υποδεικνύουν τις μεταβάσεις των διεργασιών. Αυτοί οι κανόνες ονομάζονται κανόνες μετάβασης.

Αρχικά έχουμε τον βασικό κανόνα μετάβασης:

$$\frac{P \xrightarrow{a(x)} P' \quad Q \xrightarrow{\bar{a}(x)} Q'}{} \quad \text{BASIC}$$

Αυτός ο κανόνας δηλώνει την επικοινωνία μεταξύ δύο διεργασιών. Αν μια διεργασία κάνει είσοδο στο όνομα a και μια διεργασία κάνει έξοδο στο ίδιο όνομα, τότε γίνεται η επικοινωνία και οι δύο διεργασίες μεταβαίνουν σε μια καινούργια κατάσταση.

Δεύτερον, είναι ο κανόνας PAR:

$$\frac{P \rightarrow P'}{P|Q \rightarrow P'|Q} \quad \text{PAR}$$

Αυτός ο κανόνας δηλώνει ότι, αν μπορεί μια διεργασία να μεταβεί από την κατάσταση P στην κατάσταση P' , τότε η ίδια διεργασία αν βρεθεί σε παράλληλη σύνδεση με μια διεργασία Q , να κάνει την ίδια μετάβαση.

Μετά, είναι ο κανόνας RES:

$$\frac{P \rightarrow P'}{(vx)P \rightarrow (vx)P'}$$

Αν μπορεί μια διεργασία να μεταβεί από την κατάσταση P στην κατάσταση P' από μόνη της, τότε μπορεί να το πράξει έστω και αν ένα όνομα είναι περιορισμένο μόνο σε αυτή.

Υπάρχει και ο κανόνας STRUCT ο οποίος δηλώνεται ως εξής:

$$\frac{Q \equiv P \quad P \rightarrow P' \quad P' \equiv Q'}{Q \rightarrow Q'}$$

Αν μια διεργασία Q είναι ισοδύναμη με μια διεργασία P, μια διεργασία P' είναι ισοδύναμη με μια διεργασία Q' και η διεργασία P μπορεί να μετατραπεί σε διεργασία P', τότε και η διεργασία Q μπορεί να μετατραπεί σε διεργασία Q'.

Τέλος είναι ο κανόνας «+»:

$$\frac{P \rightarrow P'}{P + Q \rightarrow P'} \quad \frac{Q \rightarrow Q'}{P + Q \rightarrow Q'}$$

Αν μια διεργασία P μεταβαίνει σε μια κατάσταση P', τότε το P+Q μπορεί να μεταβεί στην κατάσταση P'. Συμμετρικά, ισχύει και για την διεργασία Q'.

2.2.2 Παραδείγματα

Παράδειγμα 1

$$\bar{x}(y) \mid x(u).u(v)$$

Το όνομα x αποστέλλει το όνομα y μέσω της πρώτης διεργασίας το οποίο παραλαμβάνεται από την δεύτερη διεργασία. Το αποτέλεσμα είναι:

$$0 \mid y(v)$$

Όταν γίνεται η επικοινωνία, η πρώτη διεργασία έχει τελειώσει ενώ η δεύτερη έχει ακόμα μια αποστολή να κάνει.

Παράδειγμα 2

$$\bar{x}(y) \mid x(u).u(v) \mid \bar{x}(a)$$

Υπάρχουν 2 πιθανοί τρόποι επικοινωνίας μέσα σε αυτό το παράδειγμα. Είτε θα επικοινωνήσει η πρώτη διεργασία με την δεύτερη, είτε θα επικοινωνήσει η τρίτη διεργασία με την δεύτερη.

$$0 \mid y(v) \mid \bar{x}(a) \text{ ή } \bar{x}(y) \mid a(v) \mid 0$$

Παράδειγμα 3

$$!(x(y)) \mid x(u).u(v)$$

Το «!» θα αντιγράψει το περιεχόμενό του για να μπορέσει να γίνει η επικοινωνία.

$$\bar{x}(y) \mid !(x(y)) \mid x(u).u(v)$$

Τώρα μπορεί να γίνει η επικοινωνία της πρώτης διεργασίας με την τελευταία δίνοντας το ακόλουθο αποτέλεσμα:

$$!(x(y)) \mid y(v)$$

Παράδειγμα 4

$$(vy)(\bar{y}(x)) \mid y(a).a(b) + x(a) \mid \bar{y}(z)$$

Σε αυτό το παράδειγμα μόνο μια εφικτή επικοινωνία υπάρχει. Μπορεί να επικοινωνήσει μόνο η δεύτερη διεργασία με την τρίτη. Αυτό συμβαίνει διότι το y το οποίο είναι περιορισμένο, δεν μπορεί να αποσταλεί έξω από το (vy) και από το γεγονός ότι το y είναι ελεύθερο όνομα μέσα στην δεύτερη διεργασία. Αυτό θα συντελέσει στο εξής αποτέλεσμα:

$$(vy)(\bar{y}(x)) \mid z(b)$$

Παράδειγμα 5

$$\bar{x}(y) \mid x(u).u(v) + z(b).b(c) \mid \bar{z}(a)$$

Σε αυτό το παράδειγμα υπάρχουν 2 τρόποι επικοινωνίας. Είτε θα επικοινωνήσει η πρώτη διεργασία με το αριστερό σκέλος της δεύτερης, είτε θα επικοινωνήσει η τρίτη διεργασία με το δεύτερο σκέλος της δεύτερης διεργασίας. Δηλαδή τα πιθανά αποτελέσματα είναι τα εξής:

$$0 \mid y(v) \mid \bar{z}(a) \text{ ή } \bar{x}(y) \mid a(c) \mid 0$$

2.3 π-Calculus with Groups

Κατά το π-Calculus, το $(\nu x)P$ δημιουργούσε ένα νέο όνομα το οποίο είναι για χρήση μόνο εντός του πεδίου του P . Άρα μπορούμε να πούμε, για χάρη της ιδιωτικότητας, ότι προσομοίωνε ένα μυστικό το οποίο θα έμενε εντός ενός συστήματος. Όμως, στο π-Calculus υπήρχε τρόπος αποστολής του μηνύματος και εκτός αυτού του πεδίου, αν πληρούνταν κάποιες προϋποθέσεις. Ας θέσουμε το εξής παράδειγμα:

$$A \mid (\nu x)B$$

Έστω οι εξής διεργασίες A και B , τέτοιες ώστε το B να κρατά ένα «μυστικό» το οποίο δεν πρέπει να μάθει η διεργασία A . Έστω ότι το B έχει ένα ελεύθερο όνομα αποστολής του x , το a . Έτσι μπορούμε να ορίζουμε το B ως:

$$B \equiv \bar{a}(x).B'$$

Έστω ότι το A γνωρίζει το όνομα a και το χρησιμοποιεί για να παραλάβει ονόματα. Άρα το A μπορεί να οριστεί ως εξής:

$$A \equiv a(b).A'$$

Άρα, το αρχικό μας παράδειγμα μπορεί να γραφτεί ως εξής:

$$a(b).A' \mid (\nu x)\bar{a}(x).B'$$

Επιπλέον, έστω ότι στο A δεν εμφανίζεται ως ελεύθερο όνομα το x . Άρα μπορεί το πεδίο του (νx) να επεκταθεί και να περιλάβει την διεργασία A , δίνοντας το εξής αποτέλεσμα:

$$(\nu x)(a(b).A' \mid \bar{a}(x).B')$$

Με αυτόν τον τρόπο μπορούν να εφαρμοστούν οι κανόνες μετάβασης και να γίνει η επικοινωνία, περνώντας έτσι το μυστικό στην διεργασία A .

$$(\nu x)(A' \{x/b\} \mid B')$$

Έτσι, εμφανίστηκε η ανάγκη για μια πιο περιοριστική γλώσσα ή οποία να μπορεί να εφαρμοστεί για τη διατήρηση της ιδιωτικότητας. Γι' αυτόν τον λόγο προστέθηκαν στο π-Calculus οι ομάδες, δημιουργώντας το π-Calculus with Groups [5]. Στον κανονικό π-Calculus, όλες οι ομάδες είναι καθολικές, επιτρέποντας έτσι την παρατυπία που έγινε στο πιο πάνω παράδειγμα.

$$P ::= x(y:T).P \mid \bar{x}(y).P \mid P|Q \mid !P \mid (vx:T)P \mid (\nu G)P \mid 0$$

$$T ::= G[T_1, \dots, T_n]$$

Τα ονόματα τα οποία είναι δεσμευμένα, όπως το y στο $x(y)$, πλέον έχουν και ένα τύπο για να φαίνεται σε πια ομάδα ανήκουν. Ο τύπος εξειδικεύει, για το κάθε όνομα, την ομάδα που ανήκει καθώς και το τι είδους τιμές μπορούν να ανταλλαχθούν μέσω αυτού.

Οι τελεστές της γραμματικής που είδαμε στο προηγούμενο υποκεφάλαιο, ισχύουν και στην Άλγεβρα Διεργασιών π-Calculus with Groups. Όμως, λόγω του ότι επεκτάθηκε η γραμματική, υπάρχουν καινούργιοι τελεστές καθώς και καινούργιες ισοδυναμίες και κανόνες μετάβασης.

Αρχικά θα μελετήσουμε τα νέα στοιχεία της γραμματικής. Το πρώτο νέο στοιχείο είναι ο τελεστής $(\nu G)P$. Σύμφωνα με αυτόν τον τελεστή δημιουργείται μια δυναμική ομάδα η οποία δρα μόνο μέσα στο πεδίο της P . Το δεύτερο νέο στοιχείο είναι η έννοια των τύπων T ο οποίος συντελεί στην εισαγωγή του τύπου στα ονόματα τα οποία είναι δεσμευμένα.

Με την εισαγωγή του επιπλέον τελεστή, έχουμε και τις εξής επιπρόσθετες ισοδυναμίες. Η πρώτη ισοδυναμία λέει ότι εάν μια διεργασία P είναι ισοδύναμη με μια διεργασία Q , τότε το P μέσα στο πεδίο ενός G είναι ισοδύναμο με τη διεργασία Q μέσα στο πεδίο του ιδίου G .

$$P \equiv Q \Rightarrow (\nu G)P \equiv (\nu G)Q$$

Η επόμενη ισοδυναμία συμβολίζεται ως εξής:

$$(\nu G_1)(\nu G_2)P \equiv (\nu G_2)(\nu G_1)P$$

Κατά αυτή την ισοδυναμία το $(\nu G_1)(\nu G_2)P$ δηλώνει ότι η διεργασία P είναι μέσα στο πεδίο των δύο ομάδων. Αν μεταθέσουμε τις δύο αυτές ομάδες, δεν επηρεάζεται η λειτουργία της P .

Η επόμενη ισοδυναμία δηλώνει ότι αν μια ομάδα G δεν ανήκει μέσα σε ένα τύπο T , τότε μπορεί να συμπεριληφθεί εντός ενός $(vx:T)$. Δηλαδή συμβολικά αναπαρίσταται ως εξής:

$$G \neq fg(T) \Rightarrow (\nu G)(\nu x:T)P \equiv (\nu x:T)(\nu G)P$$

Επιπλέον, αν μια ομάδα G δεν είναι ελεύθερη μέσα σε μια διεργασία P τότε η διεργασία αυτή μπορεί να βγει εκτός του πεδίου του (νG) . Συμβολικά αυτό γράφεται ως:

$$G \neq \text{fg}(P) \Rightarrow (\nu G)(P \mid Q) \equiv P \mid (\nu G)Q$$

Η σημασιολογία του π -Calculus with Groups είναι η ίδια με την σημασιολογία του π -Calculus. Η προσθήκη των τύπων στο σύστημα δεν αποφέρουν αλλαγές στην σημασιολογία. Η προσθήκη όμως των ομάδων δημιουργεί έναν επιπλέον κανόνα μετάβασης:

Κανόνας GRes:

$$\frac{P \rightarrow Q}{(\nu G)P \rightarrow (\nu G)Q}$$

ο οποίος δηλώνει ότι αν μια διεργασία από την κατάσταση P μπορεί να μεταβεί στην κατάσταση Q , τότε αν περικλύσουμε την διεργασία κάτω από τον τελεστή (νx) τότε θα εξακολουθεί να έχει αυτήν την δυνατότητα μετάβασης.

Έχοντας αναφέρει τα πιο πάνω φτάνουμε στο συμπέρασμα ότι οι τύποι βοηθούν στην αποτροπή της παραβίασης της συλλογής πληροφοριών. Με το να ορίσουμε τύπους μέσα στα ονόματα εμποδίζουμε επικοινωνίες οι οποίες θα παρουσίαζαν προβλήματα ιδιωτικότητας. Η χρήση του τελεστή $(\nu x)P$ υπάρχει για να δημιουργεί ένα «μυστικό» το οποίο θα παρέμενε εντός της εμβέλειας του. Στο αρχικό παράδειγμα που δώσαμε, είδαμε ότι με κάποιες προϋποθέσεις υπάρχει ο κίνδυνος αυτό το μυστικό να διαρρεύσει προς διεργασίες οι οποίες δεν θέλαμε να έχουν πρόσβαση σε αυτό. Έτσι, έχοντας αυτό υπόψη έγιναν οι απαραίτητες αλλαγές για να εξυπηρετήσουν αυτό τον σκοπό, προσθέτοντας έτσι τα νέα στοιχεία της γραμματικής, τις ομάδες και τους τύπους.

Στη συνέχεια του υποκεφαλαίου θα διορθώσουμε το παράδειγμα που δώσαμε στην αρχή του κεφαλαίου με χρήση του π -Calculus with Groups για να δούμε πως αποτρέπεται η συλλογή πληροφοριών.

Όπως είχαμε αναφέρει στο παράδειγμα

$$A \mid (\nu x)B$$

αν το A και B ήταν ως ακολούθως

$$a(b).A' \mid (\nu x)\bar{a}(x).B'$$

τότε η επικοινωνία γινόταν παρόλο που αυτό δεν ήταν επιθυμητό. Αν προσθέσουμε τις ομάδες, μπορούμε να μεταποιήσουμε το αρχικό μας παράδειγμα ως εξής:

$$A \mid (\nu G)(\nu x: G[T])B$$

Αυτό θα οδηγούσε στο εξής αποτέλεσμα, αν εφαρμόζαμε τις ίδιες αλλαγές όπως πριν:

$$(\nu G')a(b: G'[T']).A' \mid (\nu G)(\nu x: G[T])\bar{a}(x).B'$$

Έχοντας κάνει χρήση αυτών των ομάδων, η επικοινωνία θα γινόταν αν και μόνο αν η ομάδα G και ο τύπος T ήταν γνωστά και στην διεργασία A έτσι ώστε τα $G \equiv G'$ και $T \equiv T'$, κάτι το οποίο θα είναι πολύ δύσκολο να γίνει μιας και η ομάδα G αποτελεί «μυστικό» της διεργασίας B εμποδίζοντας έτσι την επικοινωνία.

Το π -Calculus with Groups είναι καλό για την μοντελοποίηση καταστάσεων, αλλά δεν μπορεί να παρουσιάσει ακριβώς τι συμβαίνει μεταξύ των δεδομένων κατά την επικοινωνία σε υπολογιστικά συστήματα. Έτσι δεν μπορεί να αντικατοπτρίσει με ακρίβεια ένα υπολογιστικό σύστημα άρα και να βρεθεί τρόπος να μπορεί να πει κάποιος με σιγουριά ότι υπάρχει τρόπος δημιουργίας ενός αυτοματοποιημένου τρόπου εύρεσης του κατά πόσο η ιδιωτικότητα διαφυλάσσεται. Με άλλα λόγια, το π -Calculus with groups, επιτρέπει την ανάγκη της συλλογής πληροφοριών αλλά εμάς μας ενδιαφέρουν και άλλες ανάγκες όπως την ανάλυση και την διάδοση πληροφοριών. Έτσι, αυτές οι ανάγκες, έδωσαν το έναυσμα στην δημιουργία μιας πιο εξειδικευμένης άλγεβρας διεργασιών, την πGP -Calculus, με στόχο να συμπληρώσουν αυτά τα κενά.

Κεφάλαιο 3

Ένα Σύστημα Τύπων για Ιδιωτικότητα

3.1 Γραμματική και Ισοδυναμίες	22
3.1.1 Παραδείγματα	24
3.2 Υποτύποι	25
3.2.1 Παραδείγματα	26
3.3 Σύστημα Τύπων	27
3.3.1 Σύστημα Τύπων	27
3.3.2 Έλεγχος Τύπων	28
3.3.3 Πολιτική	31
3.3.4 Παραδείγματα	33

Όπως ειπώθηκε στο προηγούμενο κεφάλαιο, οι άλγεβρες διεργασιών π -Calculus και π -Calculus with Groups κατάφεραν να αντικατοπτρίσουν και να δώσουν λύση σε ένα μέρος του προβλήματος της ιδιωτικότητας. Συντέλεσαν στον έλεγχο για την συλλογή των πληροφοριών αλλά εμάς μας ενδιαφέρει περισσότερο άλλοι παράγοντες παραβίασης. Γι' αυτόν το σκοπό αναπτύχθηκε ένα σύστημα τύπων το οποίο συντέλεσε στην δημιουργία του π GP-Calculus. Η δημιουργία του συστήματος τύπων και γενικά του π GP-Calculus έγινε από τους Α. Φιλίππου και Δ. Κουζαπά [6]. Το π GP-Calculus κάνει χρήση του π -Calculus with Groups με ένα ανεπτυγμένο σύστημα τύπων. Τα ονόματα πλέον αναφέρονται ως κανάλια επικοινωνίας.

Αρχικά θα δούμε ένα παράδειγμα για να δούμε πως χρησιμοποιείται το π -Calculus with Groups με σκοπό την παρεμπόδιση της παραβίασης που αφορά το κατηγορία της συλλογής πληροφοριών. Ας θεωρήσουμε τις εξής διεργασίες:

$$\begin{aligned}
DBadmin &= \bar{a}(c).0 \\
Nurse &= a(x).\bar{b}(x).0 \\
Doctor &= b(x).x(y).\bar{x}(data).0
\end{aligned}$$

Η διεργασία του διαχειριστή της βάσης DBadmin αποστέλλει μια αναφορά c για τα δεδομένα ενός ασθενή στην διεργασία γιατρού Doctor μέσω της διεργασίας Nurse. Το κανάλι c στέλλεται στην νοσοκόμα μέσω του καναλιού a και αυτό προωθείται στον γιατρό μέσω του καναλιού b . Ο γιατρός στην συνέχεια χρησιμοποιεί το κανάλι για να γράψει και να διαβάσει τα δεδομένα του ασθενή. Για την σωστή λειτουργία του πιο πάνω παραδείγματος πρέπει να γίνει κάτω από μια νέα ομάδα νοσοκομείου Hosp, και με την τυποποίηση του καναλιού c έτσι ώστε να διασφαλιστεί ότι εξωτερικές ομάδες δεν θα μπορέσουν να αποκτήσουν πρόσβαση στα δεδομένα αυτά. Έτσι το κανάλι c θα ανήκει σε μια ομάδα G έτσι ώστε να μην διαρρεύσει έξω από το σύστημα επειδή οι ομάδες δεν είναι τιμές που μπορούν να επικοινωνήσουν και η ομάδα G είναι γνωστή μόνο από τις 3 διεργασίες. Έτσι, το γενικό σύστημα γράφεται ως εξής:

$$(v \text{ Hosp})(((vc:\text{Hosp}[])DBadmin)|Nurse|Doctor)$$

Στην συνέχεια, θα προχωρήσουμε στην παραβίαση της επεξεργασίας πληροφοριών. Αυτή η παραβίαση αφορά τα δικαιώματα που έχουν οι διεργασίες. Στο παράδειγμα που είδαμε, είδαμε ότι οι πληροφορίες δεν θα σταλούν σε οντότητες έξω από το σύστημα ορθά, αλλά δεν είδαμε να υπάρχει τρόπος που να καθορίζει την παρεμπόδιση της λάθος χρήσης των πληροφοριών αυτών από τις οντότητες εντός του συστήματος. Για παράδειγμα η νοσοκόμα δεν θα πρέπει να μπορεί να διαβάσει ή να γράψει πάνω στα δεδομένα του ασθενή, πράγμα το οποίο δεν μπορεί να παρεμποδιστεί από το π -Calculus with Groups όπως είναι. Εδώ είναι που τίθεται σε εφαρμογή το σύστημα των τύπων. Το σύστημα των τύπων είναι αυτό το οποίο ελέγχει τις δυνατότητες γραφής, ανάγνωσης καθώς και αποστολής. Στο συγκεκριμένο παράδειγμα για να ικανοποιούνται οι απαιτήσεις αυτές δηλώνουμε τους τύπους ως εξής:

$$\begin{aligned}
T_{data} &= \text{Hosp}[\text{MedicalData}]^- \\
T_c &= \text{Hosp}[T_{data}]^{rw} \\
T_a &= \text{Hosp}[\text{Hosp}[T_{data}]]^{rw} \\
T_b &= \text{Hosp}[\text{Hosp}[T_{data}]]^{rw}{}^{rw}
\end{aligned}$$

Τα κανάλια a , b και c έχουν τους αντίστοιχους τύπους T_a , T_b και T_c . Το $MedicalData$ είναι ένας βασικός τύπος ο οποίος δεν έχει δικαιώματα ούτε γραφής, ούτε ανάγνωσης. Το κανάλι c μπορεί να χρησιμοποιηθεί για να γράψιμο και διάβασμα $MedicalData$. Το κανάλι a χρησιμοποιείται για αποστολή πληροφοριών στην νοσοκόμα χωρίς να δίνει δικαιώματα στην νοσοκόμα να επεξεργαστεί τις πληροφορίες. Το κανάλι b δίνει δικαιώματα στον γιατρό να επεξεργαστεί πληροφορίες για γράψιμο και διάβασμα. Όμως παρουσιάζεται το πρόβλημα ότι η νοσοκόμα μπορεί να χρησιμοποιήσει τις ιδιότητες του καναλιού b και έτσι να μπορεί να επεξεργαστεί τα δεδομένα του καναλιού c . Για αποφυγή αυτού του προβλήματος θα πρέπει να επαναπροσδιοριστούν οι διεργασίες ως εξής:

$$DBadmin = (vb : T_b) \overline{tonurse}(b). \overline{todoc}(b). \bar{a}(c). 0$$

$$Nurse = tonurse(z). a(x). \bar{z}(x). 0$$

$$Doctor = todoc(z). z(x). x(y). \bar{x}(data). 0$$

Το κανάλι $todoc$ έχει τύπο $Hosp[T_b]^{rw}$, το κανάλι $tonurse$ έχει τύπο $Hosp[T_b']^{rw}$ όπου $T_b' = Hosp[Hosp[T_{data}]^{rw}]^w$. Έτσι η νοσοκόμα δεν έχει ικανότητες ανάγνωσης στο κανάλι b .

Όσον αφορά την διάδοση πληροφοριών πρέπει να ορίσουμε αν μπορεί ο κάτοχος των δεδομένων να διαδώσει ή όχι τις πληροφορίες. Στο παράδειγμα μας απαιτούμε από την νοσοκόμα να αποστείλει τις πληροφορίες στον γιατρό, αλλά απαγορεύουμε από αυτόν να τις στείλει παρακάτω. Αυτό μπορεί να εκφραστεί ως:

$$T_{data} = Hosp[MedicalData]^{-*}$$

$$T_c = Hosp[T_{data}]^{rw*}$$

$$T_a = Hosp[Hosp[T_{data}]^{-*}]^{rw0}$$

$$T_b = Hosp[Hosp[T_{data}]^{rw0}]^{rw0}$$

$$T_b' = Hosp[Hosp[T_{data}]^{rw0}]^{w0}$$

Η σήμανση $*$ σημαίνει ότι οι πληροφορίες μπορούν να αποσταλούν, ενώ το 0 δηλώνει την απουσία αυτού του δικαιώματος. Στο παράδειγμα βλέπουμε ότι το κανάλι a μπορεί να επικοινωνήσει με τιμές που μπορούν να σταλούν ενώ το b όχι. Επιπλέον και τα δύο κανάλια δεν μπορούν να σταλούν σε άλλες οντότητες.

3.1 Γραμματική και Ισοδυναμίες

Σε αυτό το σημείο θα παρουσιάσω και θα εξηγήσω την γραμματική η οποία απαρτίζει αυτή την νέα Άλγεβρα Διεργασιών.

Η γραμματική του νέου αυτού συστήματος με τύπους είναι η εξής:

$$S ::= (vG)P \mid (vG)S \mid S_1 \mid S_2 \mid 0$$

$$P ::= x(y:T).P \mid \bar{x}(y).P \mid P \mid Q \mid !P \mid (vx:T)P \mid 0$$

$$T ::= G[T]^{p\lambda} \mid BT$$

$$p ::= e \mid r \mid w \mid rw$$

$$\lambda ::= * \mid 0$$

Οι τρεις τελευταίοι τελεστές είναι αυτοί που απαρτίζουν την γραμματική των τύπων. Κάθε κανάλι, έχει ένα συγκεκριμένο τύπο που πρέπει να δηλωθεί για να γνωρίζουμε το είδος των δεδομένων που μπορεί να διαβάσει, να γράψει ή να στείλει.

Σε αυτό το σημείο θα εξηγήσω τους τελεστές οι οποίοι αποτελούν το βασικότερο κομμάτι της γραμματικής αυτής. Καταρχήν έχουμε το στοιχείο $T ::= BT$. Τα BT είναι βασικοί τύποι οι οποίοι είναι ορισμένοι από τον δημιουργό του συστήματος και διαφέρουν ανάλογα με το σύστημα. Δηλώνονται βάσει της πολιτικής που θέλει να διατηρήσει το σύστημα. Αποτελούν τερματική περίπτωση της γραμματικής των τύπων. Κάθε κανάλι πρέπει να περιέχει ένα βασικό τύπο. Στη συνέχεια έχουμε τη δομή $T ::= G[T]^{p\lambda}$. Αυτή η δομή, περικλείει ένα τύπο μέσα σε μια ομάδα, δίνοντας του δικαιώματα σύμφωνα με τα p και λ . Στη συνέχεια έχουμε τον τελεστή $p ::= e \mid r \mid w \mid rw$. Το p είναι ο τελεστής ο οποίος δίνει δικαιώματα στον τύπο. Αν το p είναι ίσο με e τότε το κανάλι δεν έχει δικαίωμα ούτε να γράψει, ούτε να διαβάσει. Αν έχει δικαίωμα r ή w έχει το δικαίωμα να γράψει ή να διαβάσει αντίστοιχα. Τέλος, αν έχει το δικαίωμα rw τότε δικαιούται και να διαβάσει και να γράψει πάνω στο κανάλι. Ο τελεστής $\lambda ::= * \mid 0$ είναι αυτός ο οποίος αφορά το δικαίωμα αποστολής δεδομένων. Αν το κανάλι έχει το δικαίωμα $*$ τότε μπορούν να σταλούν δεδομένα του συγκεκριμένου τύπου μέσω του καναλιού άπειρες φορές, ενώ αν έχει το δικαίωμα 0 τότε δεν μπορούν να σταλούν μέσω του καναλιού.

Οι υπόλοιποι τελεστές καθώς και οι ισότητες που ισχύουν είναι οι ίδιες με του π -Calculus with Groups.

3.1.1 Παραδείγματα

Παράδειγμα 1 – Επεξήγηση δικαιωμάτων:

Ένα κανάλι με τύπο $T = G[\text{data}]^r$ είναι ένα κανάλι που ανήκει στην ομάδα G και δικαιούται να παραλάβει δεδομένα τύπου data (r) και να σταλεί ως αντικείμενο επικοινωνίας άπειρες φορές.

Παράδειγμα 2

Έστω η διεργασία P όπου:

$$P = (\nu G)(x(y:T).\bar{a}(y))$$

Η διεργασία P παραλαμβάνει δεδομένα τύπου T στο κανάλι x το οποίο έχει τύπο $G[T]^r$ και τα στέλλει με την χρήση του καναλιού a το οποίο έχει τύπο $G[T]^w$. Η διεργασία P λειτουργεί για αποστολή δεδομένων.

Παράδειγμα 3

Έστω η διεργασία P όπου:

$$P = (\nu G)(x(y:G[T]).y(z))$$

Η διεργασία P παραλαμβάνει κανάλι τύπου $G[T]$ στο κανάλι x το οποίο έχει τύπο $G[T]^r$ και το χρησιμοποιεί για να διαβάσει ένα κανάλι z τύπου T . Η διεργασία P λειτουργεί για διάβασμα δεδομένων.

Παράδειγμα 4

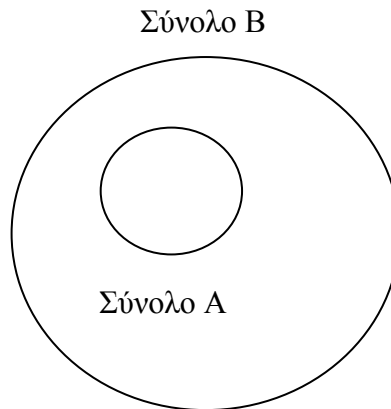
Έστω η διεργασία P όπου:

$$P = (\nu G)(x(y:G[T]).\bar{y}(z:T))$$

Η διεργασία P παραλαμβάνει κανάλι τύπου $G[T]$ στο κανάλι x το οποίο έχει τύπο $G[T]^r$ και το χρησιμοποιεί για να γράψει δεδομένα z τύπου T . Η διεργασία P λειτουργεί για γράψιμο δεδομένων.

3.2 Υποτύποι

Κατά την θεωρία συνόλων, ένα σύνολο A θεωρείται υποσύνολο κάποιου άλλου συνόλου B , αν όλα του τα στοιχεία του A συμπεριλαμβάνονται μέσα στο B . Παρομοίως μπορούμε να πούμε και για τους τύπους. Στην γλώσσα προγραμματισμού C για παράδειγμα, ο τύπος `integer`, ο οποίος περιέχει μόνο ακραίους αριθμούς, μπορούμε να πούμε ότι είναι υποτύπος του τύπου `float` ο οποίος περιέχει όλους τους φυσικούς αριθμούς.



Σε αυτό το σύστημα τύπων, ένας τύπος T_1 θεωρείται υποτύπος ενός τύπου T_2 , αν ο τύπος T_2 έχει όλα τα δικαιώματα τα οποία έχει ο T_1 .

Έστω ο τύπος $T_1 = G[\text{data}]^{r0}$ και $T_2 = G[\text{data}]^{rw*}$. Ο τύπος T_1 ανήκει στην ομάδα G και έχει δικαίωμα να παραλάβει δεδομένα τύπου `data`. Ο τύπος T_2 ανήκει στην ίδια ομάδα G και έχει δικαίωμα να παραλάβει και να αποστείλει δεδομένα τύπου `data`, καθώς και να σταλεί ως αντικείμενο επικοινωνίας άπειρες φορές. Άρα βλέπουμε ότι ο τύπος T_1 είναι υποτύπος του τύπου T_2 .

Συμβολικά, όσον αφορά τα δικαιώματα υπάρχει ο συμβολισμός $\sqsubseteq_p$ και $\sqsubseteq_\lambda$.

Το $\sqsubseteq_p$ ορίζεται ως εξής:

- $rw \sqsubseteq_p w$
- $rw \sqsubseteq_p r$
- $r, w, rw \sqsubseteq_p e$

Το Ξ_λ ορίζεται ως εξής:

$$- \quad 0 \Xi_\lambda *$$

Για να δείξουμε τη σχέση μεταξύ τύπων, χρησιμοποιούμε τον συμβολισμό $\leq$, δηλαδή αν ο τύπος T_1 είναι υποτύπος του τύπου T_2 γράφουμε $T_1 \leq T_2$.

Ο συμβολισμός $\leq$ ορίζεται ως εξής:

1. $BT \leq BT$
2. $G[T_1]^{p\lambda_1} \leq G[T_2]^{-\lambda_2}$ αν $T_1 \leq T_2$, $T_2 \leq T_1$ και $\lambda_1 \Xi_\lambda \lambda_2$
3. $G[T_1]^{p\lambda_1} \leq G[T_2]^{r\lambda_2}$ αν $T_2 \leq T_1$, $p \Xi_p r$ και $\lambda_1 \Xi_\lambda \lambda_2$
4. $G[T_1]^{p\lambda_1} \leq G[T_2]^{w\lambda_2}$ αν $T_2 \leq T_1$, $p \Xi_p w$ και $\lambda_1 \Xi_\lambda \lambda_2$
5. $G[T_1]^{rw\lambda_1} \leq G[T_2]^{rw\lambda_2}$ αν $T_1 \leq T_2$, $T_2 \leq T_1$ και $\lambda_1 \Xi_\lambda \lambda_2$

Μαζί με αυτούς τους συμβολισμούς ισχύουν όλοι όσοι ισχύουν στον π -Calculus with Groups.

3.2.1 Παραδείγματα

Παράδειγμα 1

$$G[BT]^{r*} \leq G[BT]^{e*}$$

Σε αυτό το παράδειγμα ισχύει διότι:

- $BT \leq BT$
- $r \Xi_p e$
- $* \Xi_\lambda *$

Παράδειγμα 2

$$G_1[G_2[BT]^{rw0}]^{w*} \leq G_1[G_2[BT]^{r*}]^{e*}$$

Αυτό το παράδειγμα είναι ορθό διότι:

- $G_2[BT]^{rw0} \leq G_2[BT]^{r*}$ διότι
 - $BT \leq BT$
 - $rw \sqsubseteq_p r$
 - $0 \sqsubseteq_\lambda *$
- $w \sqsubseteq_p e$
- $* \sqsubseteq_\lambda *$

Οι υποτύποι είναι σημαντικοί στην εκτέλεση της εργασίας μας. Έστω μια διεργασία θέλει να εκτελέσει μια εργασία στην οποία θα χρειαστεί ένα κανάλι τύπου T_α και της δοθεί ένα κανάλι τύπου T_b όπου όμως το T_b είναι υποτύπος του T_α τότε μπορεί να την εκτελέσει. Άρα οι υποτύποι συμβάλουν στην αναγνώριση των καναλιών τα οποία είναι ικανά να εξυπηρετήσουν τις εργασίες που πρέπει να γίνουν. Για παράδειγμα αν μια διεργασία P χρειάζεται ένα κανάλι με τύπο $T = G[BT]a^{e*}$ και της δοθεί ένα κανάλι με τύπο $T_1 = G[BT]^{r*}$ και αφού $G[BT]^{r*} \leq G[BT]^{e*}$ τότε η εκτέλεση της διεργασίας θα μπορεί να γίνει σωστά.

3.3 Τύποι

Σε αυτό το υποκεφάλαιο θα αναλύσουμε το σύστημα των τύπων, τους κανόνες μετάβασης καθώς και τα βήματα για τον έλεγχο των τύπων. Αρχικά θα αναλυθεί το σύστημα τύπων και θα εξηγηθεί ο πλήρης συμβολισμός και στην συνέχεια θα εξηγηθεί η μεθοδολογία που ακολουθείται για τον σωστό έλεγχο τύπων

3.3.1 Σύστημα Τύπων

Για να ορίσουμε το σύστημα τύπων, αρχικά θα ορίσουμε την έννοια του περιβάλλοντος. Ορίζουμε το περιβάλλον Γ ως το περιβάλλον στο οποίο θα γίνει ο έλεγχος των τύπων. Η γραμματική που ορίζει το Γ είναι η ακόλουθη:

$$\Gamma ::= \emptyset \mid \Gamma \cdot x:T \mid \Gamma \cdot G$$

Επίσης ορίζουμε ως $\text{dom}(\Gamma)$ το πεδίο ορισμού του περιβάλλοντος Γ το οποίο περιέχει όλες τις ομάδες καθώς και όλα τα κανάλια τα οποία ανήκουν στο περιβάλλον Γ . Ορίζουμε τρεις κανόνες τύπων. Η πρώτη ορίζεται ως $\Gamma \vdash x \triangleright T$ και δηλώνει ότι το x δηλωμένο κάτω από το περιβάλλον Γ και έχει τύπο T . Η δεύτερη ορίζεται ως $\Gamma \vdash P \triangleright \Gamma'$ και δηλώνει ότι η διεργασία P είναι ορθή κάτω από το περιβάλλον Γ και ότι παράγει το περιβάλλον Γ' . Η τρίτη και τελευταία απόφαση τύπων ορίζει ως $\Gamma \vdash S \triangleright \Delta$ και δηλώνει ότι το σύστημα S είναι ορθό κατά το περιβάλλον Γ και παράγει την διεπιφάνεια Δ . Μια διεπιφάνεια Δ κρατά όλες

τις ομάδες που συμμετέχουν στο S καθώς και όλα τα κανάλια μαζί με τον τρόπο χρήσης τους. Άρα μπορούμε να την ορίσουμε αυστηρά ως:

$$\Delta ::= \mathcal{E} \mid \langle G_1 \cdot \dots \cdot G_n, \Gamma \rangle \cdot \Delta$$

Στη συνέχεια θα ορίσουμε τον τελεστή $\oplus$ ο οποίος είναι δυαδικός τελεστής που αφορά μεταβλητές δικαιωμάτων λ και p , όπου:

- $\lambda_1 \oplus \lambda_2 = 0$ αν $\lambda_1 = 0$ ή $\lambda_2 = 0$
- $\lambda_1 \oplus \lambda_2 = *$ αν $\lambda_1 = *$ και $\lambda_2 = *$
- $p_1 \oplus p_2 = p_1 p_2$

Δηλαδή, για παράδειγμα, αν $p_1 = r$, $p_2 = e$ και $p_3 = rw$ τότε $p_1 \oplus p_2 = r$, $p_1 \oplus p_3 = rw$.

Επίσης υπάρχει ο δυαδικός τελεστής $\uplus$ ο οποίος αφορά περιβάλλοντα. Έστω τα περιβάλλοντα Γ_1 και Γ_2 . Το $\Gamma_1 \uplus \Gamma_2$, επιστρέφει την ένωση των 2 περιβαλλόντων και ορίζεται ως:

$$\Gamma_1 \uplus \Gamma_2 = \Gamma_1 \setminus \Gamma_2 \cdot \Gamma_2 \setminus \Gamma_1 \cdot \{x: T_1 \oplus T_2 \mid x: T_1 \in \Gamma_1, x: T_2 \in \Gamma_2\}$$

Κατά αυτόν τον ορισμό, ένα περιβάλλον $\Gamma_3 = \Gamma_1 \uplus \Gamma_2$, περιέχει τα κανάλια που είναι μοναδικά στο Γ_1 , τα κανάλια που είναι μοναδικά στο Γ_2 και για τα υπόλοιπα, τα οποία είναι κοινά, περιέχει το αποτέλεσμα που απορρέει όταν εφαρμοστεί επάνω τους ο τελεστής $\oplus$. Στη συνέχεια να δούμε και θα αναλύσουμε τους κανόνες μετάβασης.

3.3.2 Έλεγχος τύπων

Σε αυτό το υποκεφάλαιο θα δώσω τους κανόνες που ισχύουν καθώς και κάποια παραδείγματα. Θα αναφερόμαστε στους κανόνες με τα ονόματα τα οποία δόθηκαν στο [6].

- Κανόνας Name:

$$\frac{x \notin \text{dom}(\Gamma) \quad fg(T) \subseteq \text{dom}(\Gamma)}{\Gamma \cdot x : T \vdash x \triangleright T}$$

Αν το x δεν ανήκει στο πεδίο ορισμού του περιβάλλοντος Γ και το σύνολο των ελεύθερων Ομάδων του τύπου T είναι υποσύνολο του πεδίου ορισμού του

περιβάλλοντος Γ , τότε μπορούμε να πούμε ότι προσθέτοντας το x με τύπο T , τότε το x με τύπο T είναι ορθό κάτω από το νέο αυτό περιβάλλον.

- Κανόνας SubN:

$$\frac{\Gamma \vdash x \triangleright T', T' \leq T}{\Pi, \Gamma \vdash x \triangleright T}$$

Αν κάτω από το περιβάλλον Γ το x έχει τύπο T' και το T' αποτελεί υποτύπο του T , τότε κάτω από το περιβάλλον Γ το x μπορεί να έχει τύπο T .

- Κανόνας Nil:

$$\Gamma \vdash 0 \triangleright \emptyset$$

Το 0 χωρίς τύπο είναι ορθό κάτω από το περιβάλλον Γ .

- Κανόνας Rep:

$$\frac{\Gamma_1 \vdash P \triangleright \Gamma_2}{\Gamma_1 \vdash !P \triangleright \Gamma_2}$$

Αν η διεργασία P είναι ορθή κάτω από το περιβάλλον Γ_1 και παράγει το περιβάλλον Γ_2 τότε και η διεργασία $!P$ είναι ορθή κάτω από το περιβάλλον Γ_1 και παράγει το περιβάλλον Γ_2 .

- Κανόνας In:

$$\frac{\begin{array}{c} \Gamma \cdot y:T \vdash P \triangleright \Gamma' \cdot y:T' \\ \Gamma \vdash x \triangleright G[T]^{r0} \end{array}}{\Gamma \cdot y:T \vdash x(y:T).P \triangleright (\Gamma' \cdot y:T' \uplus \{x:G[T']^{r0}\})}$$

Αν η διεργασία P είναι ορθή κάτω από το περιβάλλον $\Gamma \cdot y:T$ και παράγει το περιβάλλον $\Gamma' \cdot y:T'$ και το κανάλι x μπορεί να χρησιμοποιηθεί για διάβασμα κάτω από το περιβάλλον Γ με τύπο $G[T]^{r0}$, τότε το $x(y:T).P$ είναι ορθό κάτω από το περιβάλλον $\Gamma \cdot y:T$ και παράγει το περιβάλλον $\Gamma' \cdot y:T' \uplus \{x:G[T']^{r0}\}$.

- Κανόνας Out:

$$\frac{\begin{array}{c} \Gamma \vdash P \triangleright \Gamma' \cdot y: G_y[T']^{p'\lambda} \\ \Gamma \vdash x \triangleright G_x[G_y[T]^{p*}]^{w0} \end{array}}{\Gamma \vdash \bar{x}(y).P \triangleright \Gamma' \cdot y: G_y[T']^{p'} \uplus \{x: G_x[G_y[T]^{p*}]^{w0}\}}$$

Αν η διεργασία P είναι ορθή κάτω από το περιβάλλον $\Gamma \cdot y: T$ και παράγει το περιβάλλον $\Gamma' \cdot y: G_y[T']^{p'\lambda}$ και το κανάλι x μπορεί να χρησιμοποιηθεί για γράψιμο κάτω από το περιβάλλον Γ με τύπο $G_x[G_y[T]^{p*}]^{w0}$, τότε το $\bar{x}(y).P$ είναι ορθό κάτω από το περιβάλλον Γ και παράγει το περιβάλλον $\Gamma' \cdot y: G_y[T']^{p'} \uplus \{x: G_x[G_y[T]^{p*}]^{w0}\}$.

- Κανόνας ParP:

$$\frac{\Gamma \vdash P_1 \triangleright \Gamma'_1 \quad \Gamma \vdash P_2 \triangleright \Gamma'_2}{\Gamma \vdash P_1|P_2 \triangleright \Gamma'_1 \uplus \Gamma'_2}$$

Αν η διεργασία P_1 είναι ορθή κάτω από το περιβάλλον Γ και παράγει το Γ'_1 , και αν η διεργασία P_2 είναι ορθή κάτω από το περιβάλλον Γ και παράγει το Γ'_2 , τότε το $P_1|P_2$ είναι ορθό κάτω από το περιβάλλον Γ και παράγουν το νέο περιβάλλον $\Gamma'_1 \uplus \Gamma'_2$.

- Κανόνας ResN:

$$\frac{\Gamma \cdot x: T \vdash P \triangleright \Gamma'}{\Gamma \vdash (\nu x: T)P \triangleright \Gamma'}$$

Αν η διεργασία P είναι ορθή κάτω από το περιβάλλον $\Gamma \cdot x: T$ και παράγει το περιβάλλον Γ' , τότε η διεργασία $(\nu x: T)P$ είναι ορθή κάτω από το Γ και παράγει το Γ' .

- Κανόνας ResGP:

$$\frac{\Gamma \vdash P \triangleright \Gamma'}{\Gamma \vdash (\nu G)P \triangleright \langle G, \Gamma' \rangle}$$

Αν η διεργασία P είναι ορθή κάτω από το περιβάλλον Γ και παράγει το περιβάλλον Γ' , τότε το σύστημα $(\nu G)P$ είναι ορθό κάτω από το περιβάλλον Γ και παράγει την διεπιφάνεια $\langle G, \Gamma' \rangle$.

- Κανόνας ResGS:

$$\frac{\Gamma \vdash S \triangleright \langle \widetilde{G}_i, \Gamma_i \rangle_{i \in I}}{\Gamma \vdash (\nu G)S \triangleright \langle G \cdot \widetilde{G}_i, \Gamma_i \rangle_{i \in I}}$$

Αν το σύστημα S είναι ορθό κάτω από το περιβάλλον Γ και παράγει την διεπιφάνεια $\langle \widetilde{G}_i, \Gamma_i \rangle_{i \in I}$, τότε το $(\nu G)S$ είναι ορθό κάτω από το περιβάλλον Γ και παράγει την διεπιφάνεια $\langle G \cdot \widetilde{G}_i, \Gamma_i \rangle_{i \in I}$.

- Κανόνας ParS:

$$\frac{\Gamma \vdash S_1 \triangleright \Delta_1 \quad \Gamma \vdash S_2 \triangleright \Delta_2}{\Gamma \vdash S_1 | S_2 \triangleright \Delta_1 \cdot \Delta_2}$$

Αν το σύστημα S_1 είναι ορθό κάτω από το περιβάλλον Γ και παράγει την διεπιφάνεια Δ_1 και το σύστημα S_2 είναι ορθό κάτω από το περιβάλλον Γ και παράγει την διεπιφάνεια Δ_2 τότε το $S_1 | S_2$ είναι ορθό κάτω από το περιβάλλον Γ και παράγει την διεπιφάνεια $\Delta_1 \cdot \Delta_2$.

Μόλις δημιουργηθεί το Δ τότε πρέπει να γίνει ο έλεγχος μεταξύ του Δ και της πολιτικής. Το Δ αποτελείται από μια επαναλαμβανόμενη, μια ή πολλές φορές, ακολουθία $\langle G_1, \dots, G_n, x_1, \dots, x_m \rangle$ όπου x συμβολίζουμε τα κανάλια και G τις ομάδες που ανήκουν τα κανάλια που ακολουθούν, μέχρι τις ομάδες της επόμενης επανάληψης. Κάθε κανάλι κρατά και τη χρήση του που έχει στο σύστημα, δηλαδή αν εκτελεί διάβασμα, γράψιμο ή αποστολή καθώς και τον τύπο του. Παίρνουμε ένα – ένα τα κανάλια και βλέπουμε τη χρήση που έχει στο σύστημα. Μετά ελέγχουμε αν τουλάχιστον μια από τις ομάδες στις οποίες ανήκει ικανοποιεί, και άρα επιτρέπει, την χρήση που γίνεται από το κανάλι. Αν κανένα κανάλι δεν παρουσιάσει παραβίαση της πολιτικής, τότε μπορούμε να πούμε ότι το σύστημα δεν παραβιάζει την πολιτική που δόθηκε.

3.3.3 Πολιτική

Η πολιτική στο πGP-Calculus είναι ένας τυποποιημένος τρόπος να δείξουμε την ιεραρχία που υπάρχει μεταξύ των ομάδων, καθώς και τα δικαιώματα που έχει η κάθε ομάδα πάνω σε κάθε βασικό τύπο. Η πολιτική $\wp = \langle H, R \rangle$ αποτελείται από το H που είναι η ιεραρχία των ομάδων και το R που είναι το σύνολο των δικαιωμάτων των ομάδων. Το $H = ((V, E), A)$ όπου το V είναι το σύνολο των ομάδων στο σύστημα και το E είναι το σύνολο όπου το $(G_1, G_2) \in E$ αν και μόνο αν το G_2 είναι υπο-ομάδα της ομάδας G_1 . Το A είναι ένα υποσύνολο του συνόλου V και δηλώνει τις ομάδες αποδοχής, δηλαδή τις ομάδες όπου μπορεί να σταματήσει η συμμετοχή των ομάδων. Για το παράδειγμα του νοσοκομείου θα μπορούσαμε να ορίζουμε την εξής ιεραρχία H :

$$H = \{(Hospital, Patient), (Hospital, Doctor), (Doctor, Surgeon), (Doctor, GP)\}$$

$$A = \{Patient, Doctor, Surgeon, GP\}$$

Το πιο κάτω σύστημα ικανοποιεί την πιο πάνω ιεραρχία:

$$Hosp = (v Hospital) \left((v Doctor) D_1 \right) \left| \left((v Doctor) (v Surgeon) D_2 \right) \right| (v Patient) P$$

επειδή οι ομάδες μέχρι κάθε μια διεργασία έχουν σύνδεση μεταξύ τους, δηλαδή για το D_1 το Hospital έχει σύνδεση με το Doctor και το Doctor που είναι η τελευταία ομάδα ανήκει στο A .

Το R παρουσιάζει τα δικαιώματα που επιτρέπει η πολιτική για κάθε ομάδα πάνω σε κάθε τύπο. Δηλαδή το $R = BT \times G \rightarrow 2^{Per}$ όπου BT είναι το σύνολο των βασικών τύπων, G το σύνολο των ομάδων και $Per = \{read, write, forward, exclude, nondisclose\}$. Το $read$ δίνει σε μια ομάδα το δικαίωμα να διαβάσει τα δεδομένα που ανήκουν στον βασικό τύπο, ενώ το $write$ να γράψει πάνω σε δεδομένα αυτού του τύπου. Το $exclude$ δεν δίνει κανένα δικαίωμα και το $nondisclose$ δηλώνει ότι δεν μπορούν να βγουν πληροφορίες έξω από την ομάδα αυτή.

Για να ορίσουμε τον όρο της συμβατότητας της πολιτικής $\wp = \langle H, R \rangle$ με την διεπιφάνεια $\Delta = \langle \tilde{G}_i, \Gamma_i \rangle_{i \in I}$. Η πολιτική είναι συμβατή με την διεπιφάνεια, και συμβολίζεται ως $\wp \preceq \Delta$ αν ισχύουν τα ακόλουθα:

- Σε κάθε ακολουθία ομάδων $\tilde{G}_i$, $i \in I$ πρέπει η πρώτη ομάδα να είναι ομάδα ρίζα και η τελευταία να είναι ομάδα αποδοχής.
- Για όλα τα κανάλια $x:G[BT]^{p\lambda}$ πρέπει να ισχύουν τα εξής:

- Αν στα δικαιώματα της ομάδας G στον τύπο BT ανήκει το nondisclose τότε $G' \sim G$
- Αν το $\lambda = *$ τότε πρέπει μέσα στα δικαιώματα κάποιου G μέσα στην ακολουθία των ομάδων όπου να έχει το δικαίωμα forward
- Αν το $r \in p$ τότε πρέπει μέσα στα δικαιώματα κάποιου G μέσα στην ακολουθία των ομάδων όπου να έχει το δικαίωμα read
- Αν το $w \in p$ τότε πρέπει μέσα στα δικαιώματα κάποιου G μέσα στην ακολουθία των ομάδων όπου να έχει το δικαίωμα write

Άρα μπορούμε να δηλώσουμε το εξής θεώρημα:

Αν ένα σύστημα S είναι ορθό κάτω από ένα περιβάλλον Γ και παράγει μια διεπιφάνεια Δ η οποία είναι συμβατή με την πολιτική τότε και το σύστημα S ικανοποιεί την πολιτική.

3.3.4 Παραδείγματα:

Έστω ο τύπος δεδομένων Data ο οποίος ορίζεται στην πολιτική ως εξής:

```
Data {
    G1: r*
    G2: rw0
}
```

δηλαδή το A μπορεί να διαβάσει δεδομένα τύπου Data και να τα στείλει, ενώ το B μόνο να τα διαβάσει και να γράψει πάνω τους.

Έστω το σύστημα System το οποίο ορίζεται ως :

$$System = (\nu G_1)A | (\nu G_2) B$$

$$A = x(y).y(a).\bar{b}(y)$$

$$B = b(z).z(c).\bar{z}(d)$$

Ας αναλύσουμε το A. Ο τύπος του γ είναι $G[Data]^{rw*}$, ο τύπος του x είναι $G[G[Data]^{rw*}]^r$, ο τύπος του α είναι Data και του b είναι $G[G[Data]^{rw*}]^w$. Άρα το A μπορεί διαβάσει δεδομένα τύπου Data και να προωθήσει δεδομένα τύπου Data μέσω του καναλιού γ. Άρα το A ικανοποιεί την πολιτική, άρα είναι ορθό.

Προχωρούμε στο B. Ο τύπος του b είναι $G[G[Data]^{rw*}]^r$, ο τύπος του z είναι $G[Data]^{rw}$ και ο τύπος των c,d είναι Data. Άρα το B ικανοποιεί την πολιτική, άρα είναι ορθό.

Έστω ότι χρησιμοποιώντας την ίδια πολιτική και σύστημα, το A και B ορίζοντας ως εξής:

$$A = x(y). \bar{y}(a)$$

$$B = \bar{x}(z). a(z)$$

Σε αυτό το παράδειγμα, για το A, ο τύπος του x είναι $G[G[Data]]^r$, ο τύπος του y είναι $G[Data]^w$ και του a είναι Data. Εδώ γίνεται εγγραφή μέσω του καναλιού y το οποίο δεν επιτρέπεται από την πολιτική. Για το B, ο τύπος του x είναι $G[G[Data]]^w$, του z είναι $G[Data]$ και του a είναι $G[G[Data]]^r$. Εδώ γίνεται αποστολή του καναλιού z πράγμα που δεν είναι σύμφωνο με την πολιτική, άρα και αποτυγχάνει ο έλεγχος των τύπων.

Έλεγχος τύπων με χρήση κανόνων:

Έστω το σύστημα με το λάθος στους τύπους που μόλις είδαμε.

$$\Gamma \vdash (vG_1)A \mid (vG_2)B \triangleright \Delta, \text{όπου } \Delta = \Delta_1 \cdot \Delta_2, \text{από κανόνα } ParS$$

$$\Gamma \vdash (vG_1)x(y: G[Data]^{w0}). \bar{y}(a: Data) \triangleright \Delta_1, \text{όπου } \Delta_1 = \langle G_1, \Gamma_1 \rangle, \text{από κανόνα } ResGP$$

$$\Gamma \vdash x(y: G[Data]^{w0}). \bar{y}(a: Data) \triangleright \Gamma_1, \text{όπου}$$

$$\Gamma_1 = \Gamma_2 \cdot y: T' \uplus \{x: G[T']^{r0}\}, \text{από κανόνα } In$$

$$\Gamma \vdash \bar{y}(a: Data) \triangleright \Gamma_2 \cdot y: T', \text{όπου } \Gamma_2 = \Gamma_3 \cdot a: T_a \uplus \{x: G[Data]^{w0}\}, \text{από κανόνα } Out$$

$$\Gamma \vdash 0 \triangleright \Gamma_3, \text{όπου } \Gamma_3 = \emptyset, \text{από κανόνα } Nil$$

$$\Gamma \vdash (vG_2)\bar{x}(z: G[Data]^{e*}). a(z: G[Data]^{e*}) \triangleright \Delta_2, \text{όπου } \Delta_2 = \langle G_2, \Gamma_4 \rangle, \text{από κανόνα } ResGP$$

$$\Gamma \vdash \bar{x}(z: G[Data]^{e*}). a(z: G[Data]^{e*}) \triangleright \Gamma_4 \cdot \bar{x}: T'', \text{όπου}$$

$$\Gamma_4 = \Gamma_5 \cdot z: T_z \uplus \{x: G[T_z]^{w0}\}, \text{από κανόνα } Out$$

$$\Gamma \vdash a(z: G[Data]^{e*}) \triangleright \Gamma_5, \text{όπου } \Gamma_5 = \Gamma_6 \cdot z: T_z \uplus \{a: G[T_z]^{r0}\}, \text{από κανόνα } In$$

$$\Gamma \vdash 0 \triangleright \Gamma_6, \text{όπου } \Gamma_6 = \emptyset, \text{από κανόνα } Nil$$

$$\Rightarrow \Gamma_2 = \{y: G[Data]^{w0}\}$$

$$\Rightarrow \Gamma_1 = \{y: G[Data]^{w0}, x: G[G[Data]^{w0}]^{r0}\}$$

$$\Rightarrow \Delta_1 = \langle G_1, \{y: G[Data]^{w0}, x: G[G[Data]^{w0}]^{r0}\} \rangle$$

$$\Rightarrow \Gamma_3 = \{z: G[Data]^{e*}, a: G[G[Data]^{e*}]^{r0}\}$$

$$\Rightarrow \Gamma_4 = \{x: G[G[Data]^{e*}]^{w0}, z: G[Data]^{e*}, a: G[G[Data]^{e*}]^{r0}\}$$

$$\Rightarrow \Delta_2 = \langle G_2, \{x: G[G[Data]^{e*}]^{w0}, z: G[Data]^{e*}, a: G[G[Data]^{e*}]^{r0}\} \rangle$$

$$\Rightarrow \Delta = \langle G_1, \{y: G[Data]^{w0}, x: G[G[Data]^{w0}]^{r0}\}, \\ G_2, \{x: G[G[Data]^{e*}]^{w0}, z: G[Data]^{e*}, a: G[G[Data]^{e*}]^{r0}\} \rangle$$

Εδώ παρατηρούμε ότι το κανάλι y κάνει εγγραφή κάτω από την ομάδα G_1 το οποίο δεν επιτρέπεται, καθώς και το κανάλι z κάνει αποστολή κάτω από την ομάδα G_2 που δεν επιτρέπεται ούτε αυτό. Άρα ο έλεγχος τύπων βρίσκει ότι το σύστημα είναι ασύμφωνο με την πολιτική.

Κεφάλαιο 4

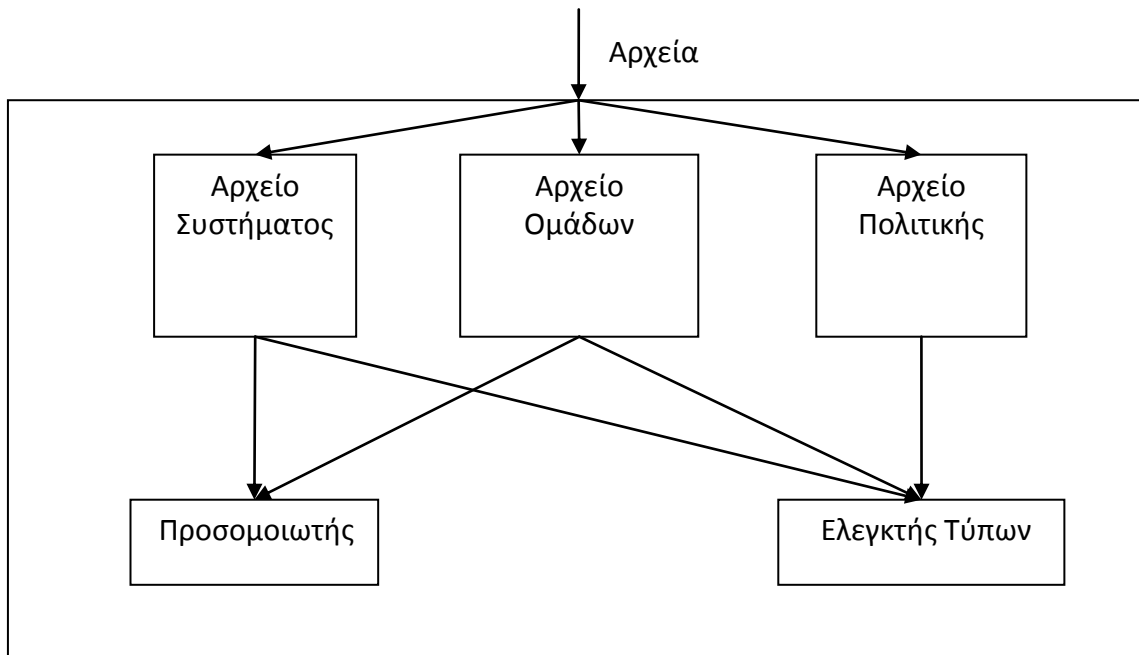
Υλοποίηση Συστήματος

4.1 Εισαγωγικά	36
4.2 Αρχεία	38
4.2.1 Αρχείο Ομάδων	38
4.2.2 Αρχείο Πολιτικών	40
4.2.3 Αρχείο Συστήματος	42
4.3 Εργαλεία	43
4.3.1 Εργαλείο Flex	43
4.3.2 Εργαλείο Bison	44
4.4 Υλοποίηση	45
4.4.1 Αντιστοίχιση Συμβόλων	45
4.4.2 Διαδικασία Δημιουργίας Δυναμικού Δέντρου Συστήματος	45
4.4.3 Διαδικασία Προσομοίωσης	47
4.4.4 Διαδικασία Ελέγχου Τύπων	50
4.4.5 Στιγμιότυπα Οθόνης	51

4.1 Εισαγωγικά

Το σύστημα που έχουμε υλοποιήσει χωρίζεται σε δύο μέρη, τον προσομοιωτή και τον ελεγκτή τύπων. Το σύστημα μετά από τις απαραίτητες εισόδους που θα πάρει θα μπορεί να εκτελέσει μια από τις δύο ενέργειες. Ο προσομοιωτής παίρνει το πρόγραμμα και το τρέχει, βρίσκοντας και εκτελώντας τους απαραίτητους κανόνες μετάβασης με βάσει τους

κανόνες του πGP-calculus. Ο ελεγκτής τύπων παίρνει το πρόγραμμα αλλά αντί να το τρέξει κάνει έλεγχο στους τύπους αν είναι σωστοί και αν συνάδουν με την πολιτική που δίνεται. Δηλαδή σε μια σχηματική αναπαράσταση, το σύστημα θα φαινόταν ως εξής:



Οι απαιτήσεις του συστήματος είναι, με την κατάλληλη είσοδο των ζητούμενων αρχείων, να μπορεί να εκτελέσει είτε προσομοίωση της εκτέλεσης των αρχείων, είτε να κάνει έλεγχο τύπων. Η προσομοίωση θα πρέπει να λειτουργεί με δύο τρόπους ανάλογα με το ποιος εκτελεί τους κανόνες μετάβασης. Θα γίνονται είτε με επιλογή του χρήστη, είτε με τυχαία επιλογή από το πρόγραμμα. Η επιλογή αυτή θα καθορίζεται στην αρχή της εκτέλεσης του συστήματος. Κατά τον έλεγχο τύπων, το σύστημα θα πρέπει να μπορεί να αποφασίσει αν τα αρχεία ικανοποιούν τις συνθήκες για να είναι ορθό το σύστημα, όσον αφορά τους τύπους, κάτω από τις συνθήκες που αναφέραμε στο προηγούμενο κεφάλαιο.

Με σκοπό να λειτουργήσει το σύστημα, χρειάζεται 3 αρχεία ως είσοδο:

- Το πρώτο αρχείο είναι το σύστημα το οποίο είναι γραμμένο σε πGP-calculus. Αυτό το αρχείο το επεξεργάζεται ο λεξικογραφικός και ο συντακτικός αναλυτής που έχει δημιουργηθεί και δημιουργεί ένα δυαδικό δέντρο το οποίο κρατά όλες τις πληροφορίες για το πρόγραμμα αυτό.

- Το δεύτερο αρχείο αφορά αποκλειστικά τις ομάδες. Δίνονται με προκαθορισμένο τρόπο οι ομάδες, οι σχέσεις μεταξύ τους καθώς και το ποιες ομάδες αποτελούν τερματικές και ποιες όχι.
- Το τρίτο αρχείο περιέχει την πολιτική στην οποία περιλαμβάνονται οι βασικοί τύποι και τα δικαιώματα που έχει κάθε Ομάδας για τον κάθε βασικό τύπο.

Η επεξεργασία των αρχείων γίνεται με την ακόλουθη σειρά. Πρώτα επεξεργάζεται το αρχείο με τις ομάδες και δημιουργείται ένας γράφος με την μορφή των λιστών γειτνίασης. Στη συνέχεια, πάνω στον γράφο προσθέτονται οι πολιτικές και τέλος δημιουργείται το δυαδικό δέντρο μέσω του προγράμματος.

4.2 Αρχεία

4.2.1 Αρχείο Ομάδων

Το αρχείο αυτό έχει μια προκαθορισμένη δομή. Αρχικά δίνονται οι ομάδες ονομαστικά με τον εξής τρόπο:

$$V: V_1, V_2, \dots, V_n$$

όπου V_i αντιπροσωπεύει το όνομα της Ομάδας i , για δήλωση των Ομάδων

$$E: (V_i, V_j)$$

$$(V_k, V_l)$$

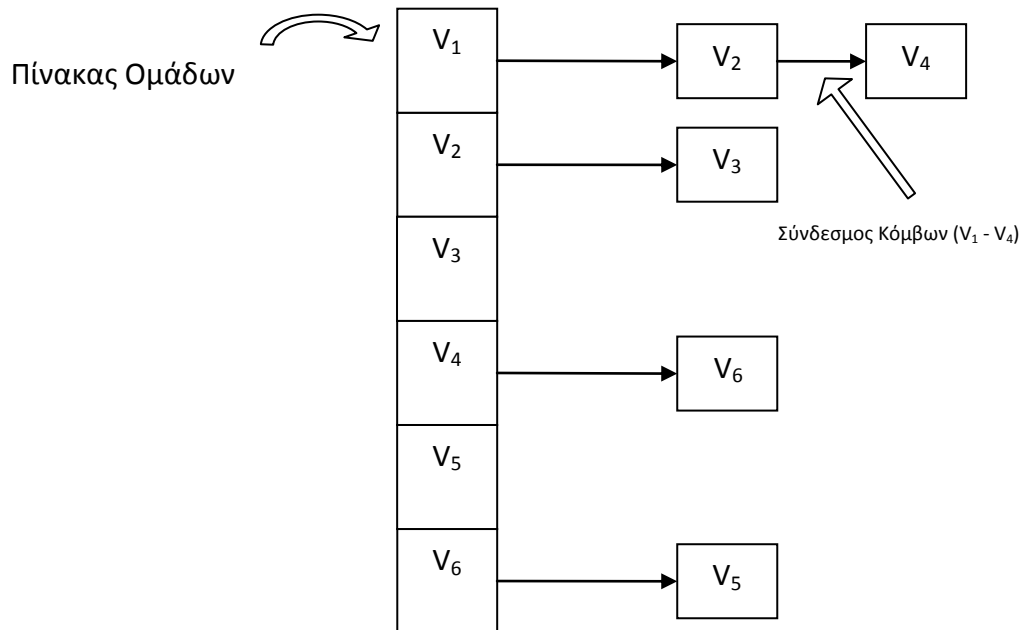
....

όπου το κάθε στοιχείο, για παράδειγμα (V_i, V_j) , συμβολίζει ότι υπάρχει ακμή στον γράφο από τον κόμβο V_i στον κόμβο V_j

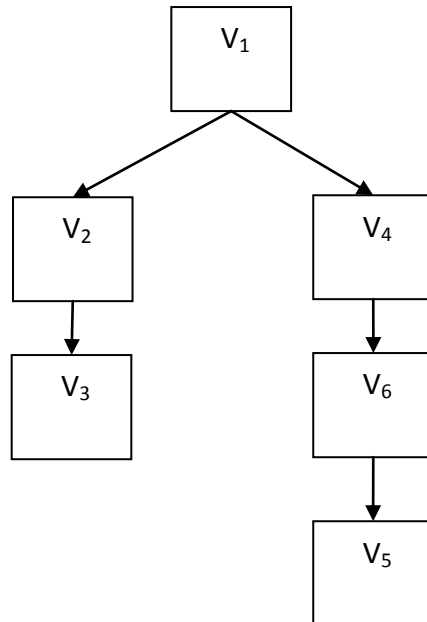
$A: V_j, V_k, \dots, V_l$, όπου V_i αντιπροσωπεύει το όνομα της Ομάδας i , για δήλωση των Ομάδων που είναι τερματικές.

Το αρχείο αυτό αποτελεί ένα βασικό στοιχείο για την εκτέλεση των εργασιών που πρέπει να γίνουν, είτε αυτές αφορούν την προσομοίωση είτε τον έλεγχο τύπων. Για να μπορέσουμε να κρατήσουμε τις πληροφορίες του αρχείου με έναν τρόπο ο οποίος να προσφέρει εύκολη πρόσβαση καθώς και τις σχέσεις μεταξύ των ομάδων συντάσσεται ένας γράφος. Ο γράφος αυτός έχει την εξής δομή, μέσω της οποίας φαίνονται οι κόμβοι, άρα οι ομάδες που απαρτίζουν το συγκεκριμένο σύστημα, καθώς και η ιεραρχία που υπάρχει.

Ο γράφος ο οποίος δημιουργείται από αυτό το αρχείο έχει την εξής μορφή:



Ο πίνακας των ομάδων, είναι μια δομή όπου φυλάσσονται όλες οι ομάδες ονομαστικά. Από κάθε κελί του πίνακα ξεκινά μια λίστα η οποία έχει ως κεφαλή το αντίστοιχο κελί του πίνακα. Όλοι οι κόμβοι που βρίσκονται σε κάθε λίστα αναπαριστούν ομάδες. Κάθε σύνδεσμος, άρα και κόμβος που υπάρχει, δηλώνει την ύπαρξη μιας κατευθυνόμενης ακμής στον γράφο των ομάδων που ξεκινά από την κεφαλή της λίστας και καταλήγει στον κάθε κόμβο της αντίστοιχης λίστας. Το παράδειγμα που φαίνεται πιο πάνω αναπαριστά τον εξής γράφο:



Το αρχείο αυτό ανοίγεται, και διαβάζεται γραμμή – γραμμή. Διαβάζεται πρώτα το V όπου και δημιουργείται ο πίνακας των λιστών γειτνίασης για την δημιουργία του γράφου. Κάθε φορά που διαβάζεται μια γραμματοσειρά που αναπαριστά ένα όνομα ομάδας, πηγαίνω στην επόμενη διαθέσιμη θέση, σε ένα πίνακα αρχικοποιημένο με 10 θέσεις και τοποθετώ το όνομα εκεί. Σε περίπτωση που ο πίνακας έχει γεμίσει, δεσμεύω μνήμη για ακόμα 10 θέσεις, αντιγράφω τον πίνακα στον καινούργιο και συνεχίζω μέχρι να ξαναγεμίσει ή μέχρις ότου δεν υπάρχουν άλλες ομάδες για διάβασμα.

Στην συνέχεια διαβάζονται οι συνδέσεις E. Για κάθε σύνδεση που διαβάζεται πηγαίνω στον τώρα έτοιμο πίνακα με τις ομάδες στην θέση της πρώτης ομάδας της σύνδεσης και προσθέτω τη δεύτερη ομάδα στην λίστα του με τις συνδέσεις. Έτσι κρατώ τον γράφο κατευθυνόμενο. Τέλος διαβάζω το A και για κάθε ομάδα που διαβάζω πάω στον πίνακα του γράφου και δηλώνω ότι η συγκεκριμένη ομάδα αποτελεί τερματικό κόμβο σε μια μεταβλητή η οποία είναι δηλωμένη μέσα σε κάθε κόμβο.

4.2.2 Αρχείο Πολιτικών

Το αρχείο αυτό περιέχει τους βασικούς τύπους και τα δικαιώματα που έχει κάθε Ομάδα σε κάθε βασικό τύπο. Αυτό το αρχείο αποτελεί μια διαδοχή από την εξής δομή:

```

Basic_Type{

    G1: rights

    G2: rights

    :

    Gk: rights

}

```

όπου Basic_Type είναι το όνομα του βασικού τύπου και G_i το όνομα της i Ομάδας. Δίπλα από κάθε G_i υπάρχουν τα δικαιώματα του για τον συγκεκριμένο βασικό τύπο. Τα δικαιώματα (rights) αυτά είναι τα ακόλουθα:

read: Αυτό δίνει το δικαίωμα στην Ομάδα να διαβάσει δεδομένα αυτού του τύπου Basic_Type.

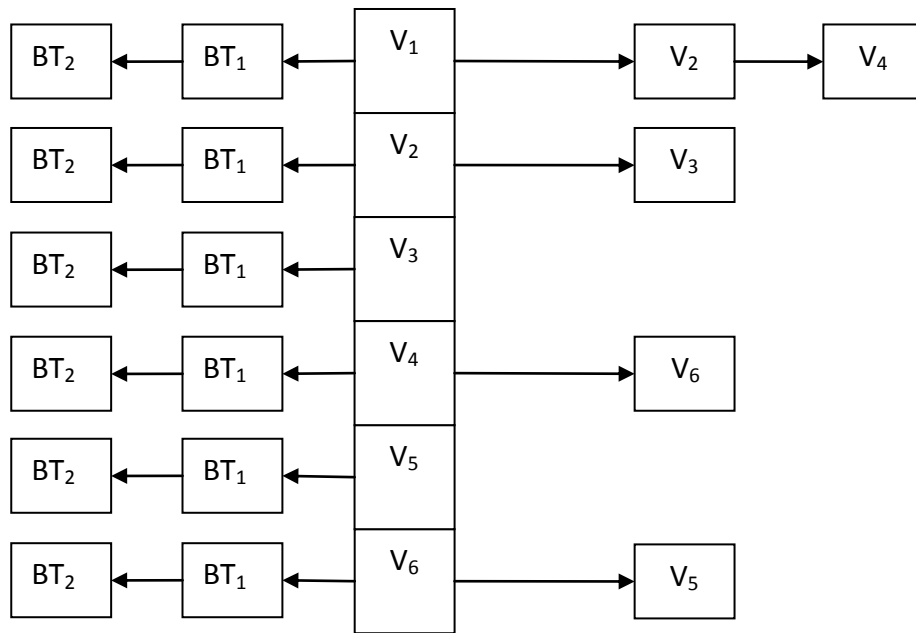
write: Αυτό δίνει το δικαίωμα στην Ομάδα να γράψει δεδομένα αυτού του τύπου Basic_Type.

forward: Αυτό δίνει το δικαίωμα στην Ομάδα να προωθήσει δεδομένα του τύπου Basic_Type.

exclude: Αυτό εμποδίζει την Ομάδα στο να έχει οποιοδήποτε δικαίωμα στα δεδομένα τύπου Basic_Type.

nonextrusion: Αυτό εμποδίζει την πληροφορία να εξέλθει από την Ομάδα που αναφέρεται.

Για κάθε βασικό τύπο δημιουργείται ένας κόμβος για κάθε μια ομάδα δηλωμένη στον πίνακα των ομάδων και κρατά μέσα της πληροφορίες οι οποίες αφορούν τα δικαιώματα που έχει η συγκεκριμένη ομάδα για τον βασικό τύπο. Στο παράδειγμα που δόθηκε πριν, ο γράφος θα άλλαζε ως εξής για την δήλωση δύο βασικών τύπων:



4.2.3 Αρχείο Συστήματος

Το σύστημα πρέπει να είναι γραμμένο στο πGP-calculus με την μόνη διαφορά ότι τα ελεύθερα κανάλια θα πρέπει να είναι δηλωμένα στην αρχή του αρχείου, μαζί με τον τύπο τους. Χρησιμοποιείται ειδικός συμβολισμός ο οποίος θα αναφερθεί στο υποκεφάλαιο 4.4. Για την επεξεργασία αυτού του αρχείου χρησιμοποιήθηκαν τα εργαλεία Flex και Bison.

Αυτά τα εργαλεία λειτουργούν μαζί με το αρχείο συστήματος το οποίο διαβάζεται σιγά - σιγά από τον λεξικογραφικό αναλυτή Flex και εισαγάγει τα δεδομένα που αναγνώρισε στον συντακτικό αναλυτή Bison με σκοπό την αναγνώριση του αρχείου ως ένα σύνολο.

4.3 Εργαλεία

4.3.1 Εργαλείο Flex

Το Flex όπως αναφέρθηκε είναι ένας λεξικογραφικός αναλυτής. Δημιουργήθηκε από τον V. Paxson το 2001 και λειτουργεί σε περιβάλλοντα τα οποία είναι βασισμένα στο περιβάλλον UNIX. Αποτελεί την δωρεάν έκδοχή του λεξικογραφικού αναλυτή LEX. Η χρονική πολυπλοκότητα των προγραμμάτων που χρησιμοποιούν τον λεξικογραφικό αυτό αναλυτή είναι $O(n)$ όπου n δηλώνει το μέγεθος του αρχείου το οποίο θα αναγνωριστεί από τον αναλυτή. Χρησιμοποιεί κανονικές εκφράσεις για να αναγνώσει ένα αρχείο. Οι κανονικές εκφράσεις δηλώνονται στην αρχή του αρχείου καθώς και το τι αναπαριστούν. Για παράδειγμα αν θέλουμε να δηλώσουμε μια κανονική έκφραση που να αναγνωρίζει αριθμούς, στο Flex θα δηλωθεί ως ακολούθως:

NUMBER "[0-9]+"

Για να δηλώσουμε κανονικές εκφράσεις στο Flex χρησιμοποιούμε τον εξής συμβολισμό:

- “ x ”, όπου x μπορούμε να βάλουμε οποιοδήποτε σύμβολο ή συμβολοσειρά για να αναγνωρίζουμε το συγκεκριμένο σύμβολο.
- $[x-y]$, όπου x και y μπορούμε να βάλουμε οποιαδήποτε σύμβολα για να αναγνωρίζουμε ένα σύμβολο μεταξύ του x και y στον πίνακα ASCII.
- x^* , όπου στο x μπορεί να εισαχθεί οποιοσδήποτε από τις προηγούμενες κανονικές εκφράσεις και σημαίνει ότι η κανονική έκφραση μπορεί να επαναληφθεί από 0 μέχρι άπειρες φορές.
- x^+ , όπου στο x μπορεί να εισαχθεί οποιοσδήποτε από τις πρώτες δύο κανονικές εκφράσεις και σημαίνει ότι η κανονική έκφραση μπορεί να επαναληφθεί από 1 μέχρι άπειρες φορές.
- $x?y$, όπου στο x και y μπορεί να μπει οποιαδήποτε κανονική έκφραση και δηλώνει ότι το x μπορεί να αναγνωριστεί αλλά μπορεί και όχι, αλλά το y πρέπει να αναγνωριστεί για να αναγνωριστεί η κανονική έκφραση ως σύνολο.

Οι εκφράσεις αυτές αναγνωρίζονται και εκτελείται κώδικας αποστολής της γραμματοσειράς που αναγνωρίστηκε στο Bison καθώς και το τι αντιπροσωπεύει.

4.3.2 Εργαλείο Bison

Το εργαλείο Bison όπως προαναφέρθηκε είναι ένας συντακτικός αναλυτής. Δημιουργήθηκε από τον R. Corbett ως η δωρεάν έκδοση του συντακτικού αναλυτή Yacc. Χρησιμοποιείται για την σύνταξη της γραμματικής που απαρτίζει την γλώσσα. Για να συνταχτεί ένας κανόνας γραμματικής, χρησιμοποιείται μια συγκεκριμένη δομή. Για παράδειγμα αν θέλουμε να συντάξουμε τον εξής κανόνα γραμματικής

$$S : x(y) \mid !S$$

θα ορίσουμε τον κανόνα για το S ως εξής:

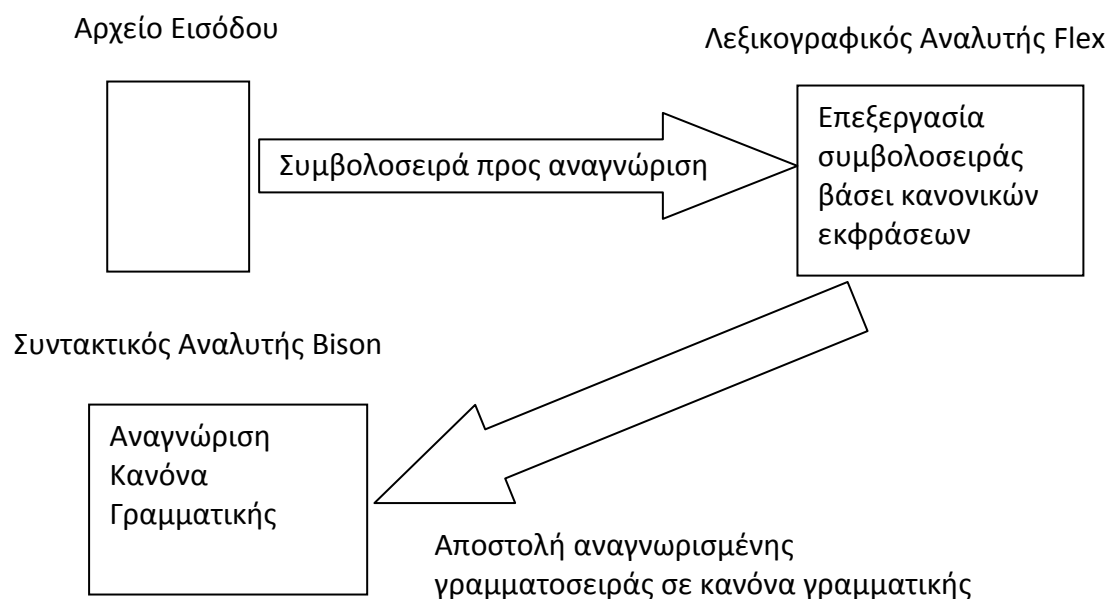
S: x '(' y ')' { /*CODE IN C*/ }

| '!' S { /*CODE IN C*/ }

;

Μέσα στις αγκυλωτές παρενθέσεις θα εκτελεστεί κώδικας την στιγμή που θα αναγνωριστεί πλήρως ο κανόνας στα αριστερά τους.

Σχηματικά, θα αναπαριστούσαμε την σχέση του αρχείου τους συστήματος, του αρχείου το εργαλείου Flex και το αρχείο του εργαλείου Bison ως εξής:



4.4 Υλοποίηση

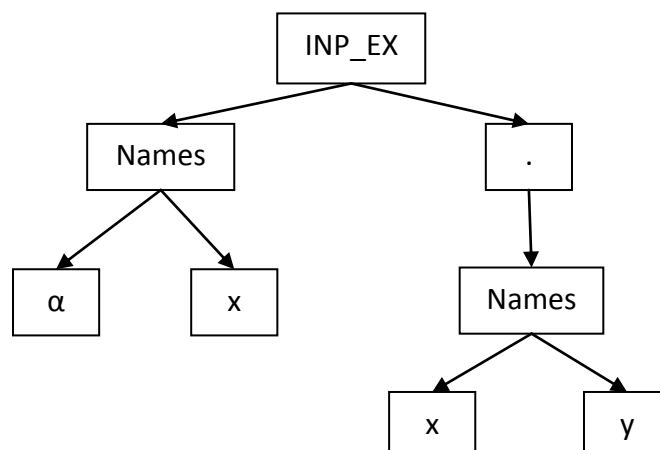
4.4.1 Αντιστοίχιση Συμβόλων

Λόγω του ότι οι άλγεβρες διεργασιών αναπτύχθηκαν για θεωρητικές μελέτες και όχι για προγραμματισμό, ο χρήστης θα δυσκολευτεί να γράψει συμβολισμούς στον υπολογιστή όπως το $\bar{x}$. Έτσι πρέπει να δημιουργήσουμε μια αντιστοίχιση μεταξύ των συμβόλων που υπάρχουν στην θεωρία με αυτά που χρησιμοποιούνται στο σύστημα. Στον παρακάτω πίνακα θα παραθέσω τους συμβολισμούς οι οποίοι διαφέρουν από την θεωρία, στο σύστημα.

Θεωρία	Σύστημα
(vx)	$(\text{new } x : T)$
(vG)	$(\text{new } G)$
$\bar{x}(y)$	$\sim x(y)$

4.4.2 Διαδικασία Δημιουργίας Δυαδικού Δέντρου Συστήματος

Το σύστημα, με τη χρήση των πιο πάνω εργαλείων κωδικοποιείται σε ένα δυαδικό δέντρο. Σκοπός αυτής της κωδικοποίησης είναι η αναπαράσταση του συστήματος με τέτοιο τρόπο έτσι ώστε να διακρίνεται η σειρά με την οποία θα πρέπει να εκτελούνται οι διεργασίες για να λάβουμε ένα σωστό αποτέλεσμα. Με σκοπό να φανεί αυτή η εξάρτηση θα δώσω αυτό το από παράδειγμα. Έστω η απλή διεργασία $P = \alpha(x).x(y)$. Σε αυτή τη διαδικασία, δεν μπορεί να επικοινωνήσει το κανάλι $x(y)$ αν δεν επικοινωνήσει πρώτα το κανάλι $\alpha(x)$. Πρέπει να αναφέρουμε ότι στο δυαδικό δέντρο αυτό, προηγείται η εκτέλεση των αριστερών φύλλων ενός κόμβου και μετά των δεξιών. Έτσι κατασκευάζεται το εξής δέντρο:



Ο κόμβος INP_EXP, είναι κόμβος ο οποίος παρουσιάζει ότι το υπόδενδρο που ακολουθεί αποτελεί κανάλι εισόδου. Κάθε κόμβος περιέχει ένα «τύπο», ο οποίος είναι ευθέως ανάλογος με το τι αντιπροσωπεύει ο κόμβος αυτός και χαρακτηρίζει το υπόδενδρο που ακολουθεί.

Υπάρχουν οι εξής τύποι κόμβων:

Τύπος	Τι αντιπροσωπεύει
INP_EXP	Το υπόδενδρο που ακολουθεί είναι κανάλια που χρησιμεύουν για είσοδο, $\alpha(x)$
OUTP_EXP	Το υπόδενδρο που ακολουθεί είναι κανάλια που χρησιμεύουν για έξοδο, $\bar{\alpha}(x)$
NEW_EXP	Αποτελεί τον κόμβο για το (vx) και το υπόδενδρο που ακολουθεί είναι το πεδίο που αφορά.
NEWG_EXP	Αποτελεί τον κόμβο για το (vG) και το υπόδενδρο που ακολουθεί είναι το πεδίο που αφορά.
BANG_EXP	Αποτελεί τον κόμβο για το $!$ και το υπόδενδρο που ακολουθεί είναι το πεδίο που αφορά.
DOT_EXP	Αποτελεί τον τελεστή «.»
PAR_EXP	Αποτελεί τον τελεστή « »

Σε περίπτωση μοναδικών τελεστών, όπως το «!» ή το (vx) προστίθεται ένας κόμβος με ένα παιδί πάνω από το δέντρο που θα δημιουργηθεί από τα επόμενα.

Το δέντρο αυτό δημιουργείται βήμα-βήμα με την αναγνώριση κάθε κανόνα από το αρχείο Bison. Κάθε φορά που αναγνωρίζεται ένας κανόνας, τότε δημιουργείται ένας κόμβος που κρατά τις πληροφορίες του κανόνα που αναγνωρίστηκε. Για να αναγνωριστεί ένας κανόνας είναι αναγκαίο να βρεθεί σε τερματική περίπτωση. Οι κανόνες, λόγω της προθεσμίας της τερματικής περίπτωσης, αναγνωρίζονται από κάτω προς τα πάνω. Άρα, στο δικό μας παράδειγμα, θα αναγνωριστούν πρώτα οι κανόνες για τα $\alpha(x)$ και $x(y)$ και μετά ο κανόνας που τα ενώνει, κάνοντας την δημιουργία του δέντρου αυτού εφικτή. Δηλαδή, όταν

αναγνωριστεί ο κανόνας για το «.» τότε δημιουργείται ένας νέος κόμβος που την κρατά, και θέτει ως παιδιά του τους 2 άλλους κόμβους που αναγνωρίστηκαν πιο πριν.

4.4.3 Διαδικασία προσομοίωσης

Όταν τελεύσει η αναγνώριση των αρχείων και η δημιουργία του δέντρου, δίνεται στον χρήστη η επιλογή αν θα τρέξει τον προσομοιωτή ή αν θα κάνει έλεγχο των τύπων. Αν επιλέξει να τρέξει τον προσομοιωτή, τότε εκτελούνται οι παρακάτω ενέργειες.

Αρχικά γίνεται μια περιήγηση στο δέντρο και μεταφέρονται όλες οι διεργασίες οι οποίες είναι δηλωμένες έξω από την διεργασία main που αποτελεί το σύστημα, και αντιγράφεται το σώμα τους στη θέση που βρίσκεται το όνομα της διεργασίας. Στη συνέχεια γίνεται μια περιήγηση μέσα στο δυαδικό δέντρο και συλλέγονται όλα τα ζευγάρια καναλιών τα οποία μπορούν να επικοινωνήσουν. Πριν την περιήγηση αυτή, γίνεται μια μετονομασία στα κανάλια τα οποία ανήκουν εντός του πεδίου ενός κόμβου (vx) για να μην υπάρξει λάθος ζευγάρωμα με κανάλια τα οποία είναι έξω από το πεδίο. Αν ένα κανάλι μπορεί να επικοινωνήσει με πάνω από ένα άλλο κανάλι, τότε υπάρχει ως ζευγάρι και με τα δύο κανάλια αυτά. Αν ο χρήστης επιλέξει να κάνει εκείνος τις μεταβάσεις τότε παρουσιάζονται όλα τα ζεύγη σε αυτόν για να κάνει την επιλογή, αλλιώς επιλέγεται ένα τυχαία. Μετά λαμβάνονται οι εξής αποφάσεις:

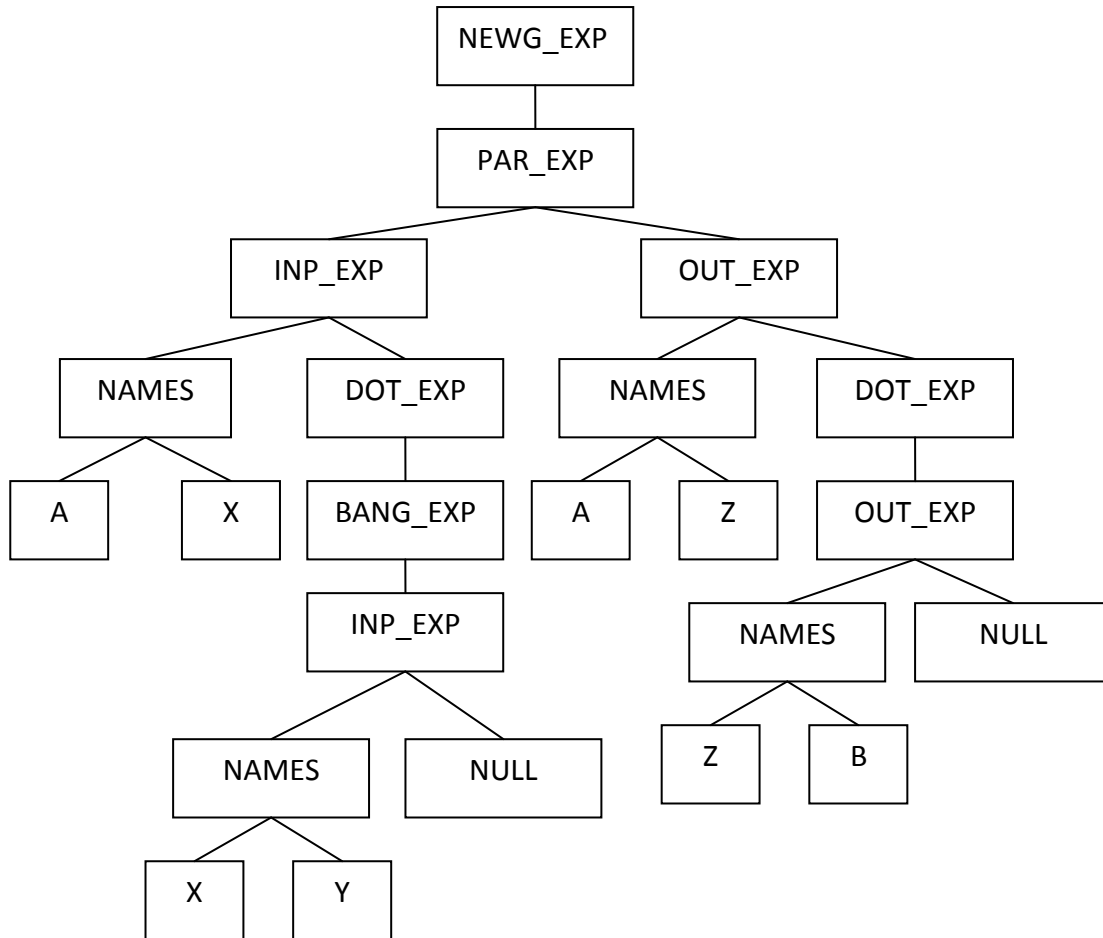
- Αν ένα ή δύο από τα κανάλια επικοινωνίας βρίσκεται κάτω από «!» τότε δεν θα υπάρξει αλλαγή στο δέντρο μετά την εφαρμογή του κανόνα μετάβασης για τα κανάλια που αφορά το «!».
- Μόλις επικοινωνήσουν τα δύο κανάλια, αν δεν ισχύει η περίπτωση που είπαμε πιο πάνω, αφαιρείται ο κόμβος που τα αντιπροσωπεύει από το δέντρο. Επιπλέον γίνονται και όλες οι αναγκαίες αλλαγές στο δέντρο. Για παράδειγμα, εάν γίνει η επικοινωνία ενός καναλιού $\alpha(x).P$, πρέπει όλες οι εμφανίσεις του x μέσα στο P να αντικατασταθεί με το όνομα του καναλιού το οποίο θα λάβει το α . Δηλαδή αν επικοινωνήσει με το $\bar{\alpha}(y)$ πρέπει να αντικατασταθούν όλες οι εμφανίσεις του x με y .

Αυτή η διαδικασία περιήγησης και μετάβασης καταστάσεων, επαναλαμβάνεται μέχρι να μην υπάρχουν άλλες δυνατές μεταβάσεις.

Ως παράδειγμα θα παρουσιάσω γραφικά τι αλλαγές γίνονται κατά την εφαρμογή κανόνων μετάβασης. Αρχικά έχουμε το εξής σύστημα:

$$S = (vG : T_1)(a(x:T_2).!x(y:T_3)|\bar{a}(z).\bar{z}(b))$$

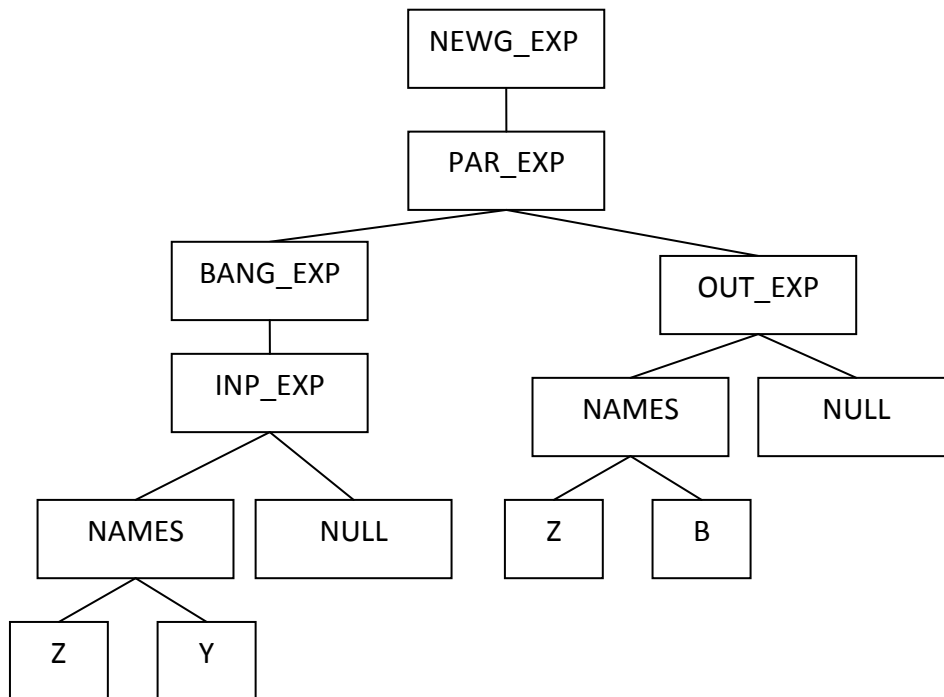
Γραφικά το σύστημα θα δημιουργήσει το εξής δένδρο:



Αρχικά εκτελείται η επικοινωνία πάνω στο κανάλι α. Το σύστημα τότε θα μεταβεί στην εξής κατάσταση:

$$S = (vG : T_1)(!z(y:T_3)|\bar{z}(b))$$

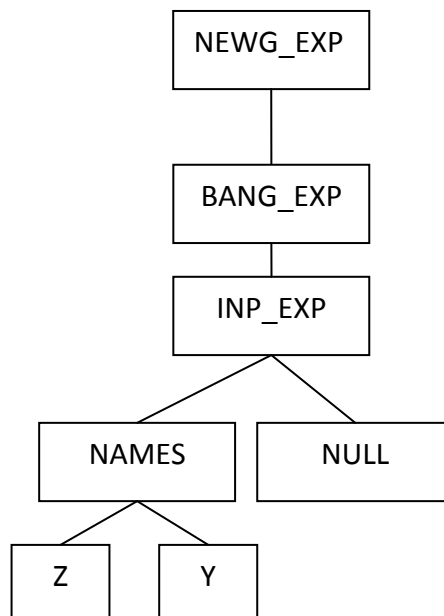
Γραφικά, στο σύστημα η δομή θα έπαιρνε την εξής μορφή:



Στη συνέχεια θα εκτελεστεί επικοινωνία πάνω στο κανάλι z. Άρα το σύστημα θα μεταβεί στην εξής κατάσταση:

$$S = (vG : T_1)(!z(y:T_3))$$

Το ένα από τα 2 z δεν φεύγει λόγω του τελεστή !. Η γραφική αναπαράσταση του δένδρου θα πάρει την εξής μορφή:



4.4.4 Διαδικασία Ελέγχου Τύπων

Για να γίνει ο έλεγχος των τύπων, πρέπει να δημιουργήσουμε το περιβάλλον Γ . Αρχικά το Γ είναι άδειο. Προσθέτουμε μέσα όλες τις ομάδες από το αρχείο των ομάδων. Επιπλέον προσθέτουμε τα κανάλια τα οποία είναι ελεύθερα, και άρα δηλωμένα στην αρχή του αρχείου του συστήματος. Στη συνέχεια γίνεται μια περιήγηση μέσα στο δέντρο προσθέτοντας στο Γ όλα τα κανάλια τα οποία δεν είναι ελεύθερα, και ενημερώνοντας τα ελεύθερα κανάλια, κρατώντας τον τελευταίο τύπο που συναντήθηκε κατά την πλοήγηση. Η πλοήγηση αυτή γίνεται αναδρομικά με DFS. Όταν το DFS φτάσει σε τερματική περίπτωση και αρχίσει να κλίνει αναδρομικά τις κλήσεις της συνάρτησης που το εκτελεί, δημιουργούμε το Γ' . Το Γ' είναι ένα περιβάλλον, το οποίο αρχικά είναι άδειο και είναι υλοποιημένο με την χρήση μιας λίστας, και για κάθε κανάλι που συναντούμε το προσθέτουμε στο Γ' αν δεν υπάρχει, ενώ αν υπάρχει εκτελούμε τον τελεστή $\oplus$ μεταξύ της νέας και προηγούμενης του εμφάνισης. Υπενθυμίζω ότι το DFS γίνεται σε δυαδικό δέντρο, άρα σε περίπτωση 2 παιδιών ενός κόμβου, θα φτάσουν 2 Γ' σε αυτόν τον κόμβο και αυτά θα ενωθούν εκτελώντας τον κανόνα που είδαμε στο Κεφάλαιο 3.1

$$\Gamma_1 \cup \Gamma_2 = \Gamma_1 \setminus \Gamma_2 \cdot \Gamma_2 \setminus \Gamma_1 \cdot \{x: T_1 \oplus T_2 | x: T_1 \in \Gamma_1, x: T_2 \in \Gamma_2\}$$

Μόλις συναντήσουμε τον πρώτο κόμβο ο οποίος περιέχει (νG) όπου G το όνομα μιας ομάδας, προσθέτουμε την ομάδα στην αρχή του Γ . Τότε, το περιβάλλον Γ θεωρείται ως διεπιφάνεια Δ η οποία έχει την μορφή $\langle G_1, \dots, G_n, x_1 \dots x_m \rangle$, όπου G_i αναπαριστούν ομάδες ενώ τα x_i κανάλια. Από τη στιγμή που θα συναντήσουμε το πρώτο (νG) δεν θα συναντήσουμε άλλα κανάλια. Μπορούμε όμως να συναντήσουμε τελεστή «|». Τότε, αν θεωρήσουμε ότι το Δ_1 είναι η διεπιφάνεια στα αριστερά του | και το Δ_2 η διεπιφάνεια στα δεξιά, τότε δημιουργούμε την διεπιφάνεια Δ' η οποία έχει την μορφή $\Delta = \langle \Delta_1, \Delta_2 \rangle$. Προγραμματιστικά γινόταν συγχώνευση των δύο λιστών.

Όταν τελείωνε η διαδικασία αυτή, γινόταν έλεγχος των καναλιών στο κατά πόσο τηρούσαν τους τύπους. Αν από το τελικό Δ ο τύπος που είχε το κανάλι x είναι υποτύπος από το κανάλι x στο Γ τότε είναι ορθό. Αλλιώς υπάρχει λάθος στους τύπους. Αυτό γινόταν με την προσπέλαση της λίστας του Δ και με τη συσχέτιση του κάθε κόμβου με τον αντίστοιχο κόμβο στη λίστα του Γ .

Ως τελευταίο βήμα, γίνεται η σύγκριση του Δ με την πολιτική. Μέχρι στιγμής έχουμε μια δομή η οποία κρατά το Δ σε μορφή μιας λίστας. Πάμε και βρίσκουμε κάθε κανάλι και βλέπουμε πως χρησιμοποιείται από τον τύπο που έχει στο Δ συλλέγοντας έτσι τα δικαιώματα τα οποία χρειάζεται. Μετά κοιτάζουμε μια-μια τις ομάδες στις οποίες ανήκει. Αν από τις ομάδες αυτές καλύπτονται τα δικαιώματα τότε αυτό το κανάλι ικανοποιεί την πολιτική. Συνεχίζουμε μέχρι να εξαντληθούν τα κανάλια. Αν όλα τα κανάλια ικανοποιούν την πολιτική, τότε και το σύστημα την ικανοποιεί. Αν έστω και ένα κανάλι δεν ικανοποιεί την πολιτική, τότε ο έλεγχος αυτός αποτυγχάνει και άρα το σύστημα δεν ικανοποιεί την πολιτική.

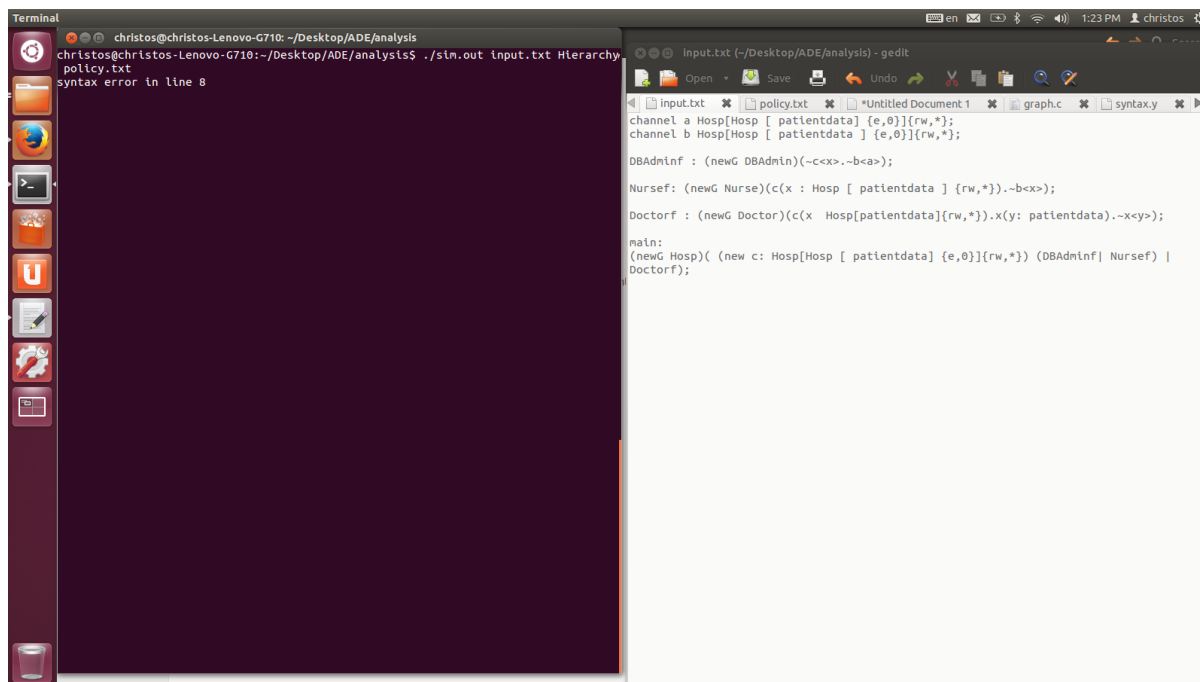
Αλγόριθμος σε Βηματική Μορφή:

1. Αναγνώρισε τις ομάδες από τον γράφο και βάλε τις στο περιβάλλον Γ
2. Αναγνώρισε τα ελεύθερα κανάλια από τις δηλώσεις και πρόσθεσέ τα στο περιβάλλον Γ.
3. Εκτέλεσε κατά βάθος αναζήτηση στο δυαδικό δένδρο και βρες τα υπόλοιπα κανάλια και βάλε τα στο περιβάλλον Γ.
4. Εκτέλεσε κατά βάθος αναζήτηση και δημιούργησε την διεπιφάνεια Δ:
 - Κάθε φορά που βρίσκεις ομάδα κατά την περιήγηση βάζε την στο Δ
 - Κάθε φορά που βρίσκεις κανάλι, πρόσθετε το κανάλι με την χρήση που γίνεται πάνω σε αυτό το κανάλι και αν υπάρχει ήδη πρόσθετε την νέα χρήση του. Δηλαδή αν δούμε το κανάλι α σε θέση εισόδου και υπήρχε ήδη σε θέση εξόδου, τότε του δίνουμε και τις 2 χρήσεις.
 - Κατεβαίνοντας στο δένδρο αν συναντηθεί δήλωση νέου ονόματος μπαίνει μπροστά στην λίστα και η ανάλογη δήλωση αφαιρείται από την λίστα κατά την άνοδο στο δένδρο για να αποφευχθούν συγκρούσεις ονομάτων.
5. Μόλις τελειώσει η περιήγηση κατά βάθος κάνε έλεγχο με την πολιτική:
 - Για κάθε κανάλι έλεγξε αν οι ομάδες που προηγούνται έχουν τα δικαιώματα για να υποστηρίξουν την πραγματική χρήση του καναλιού.

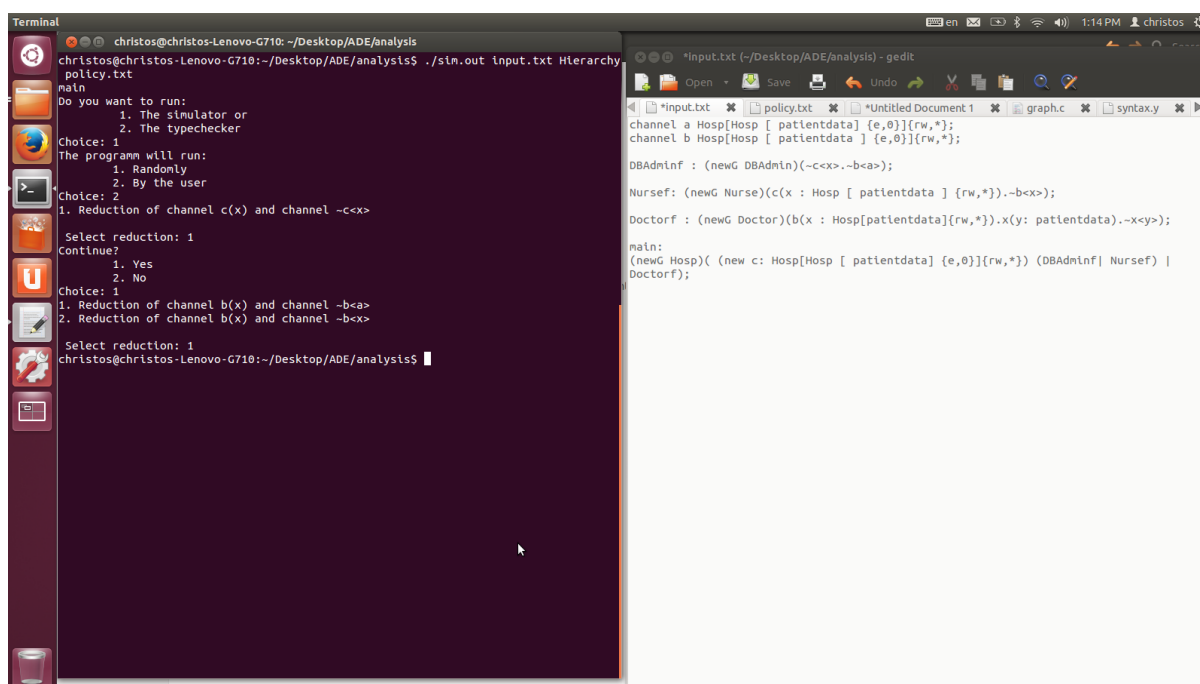
4.4.5 Στιγμιότυπα Οθόνης

Σε αυτό το υποκεφάλαιο θα παρουσιαστούν στιγμιότυπα οθόνης δείχνοντας την χρήση του συστήματος.

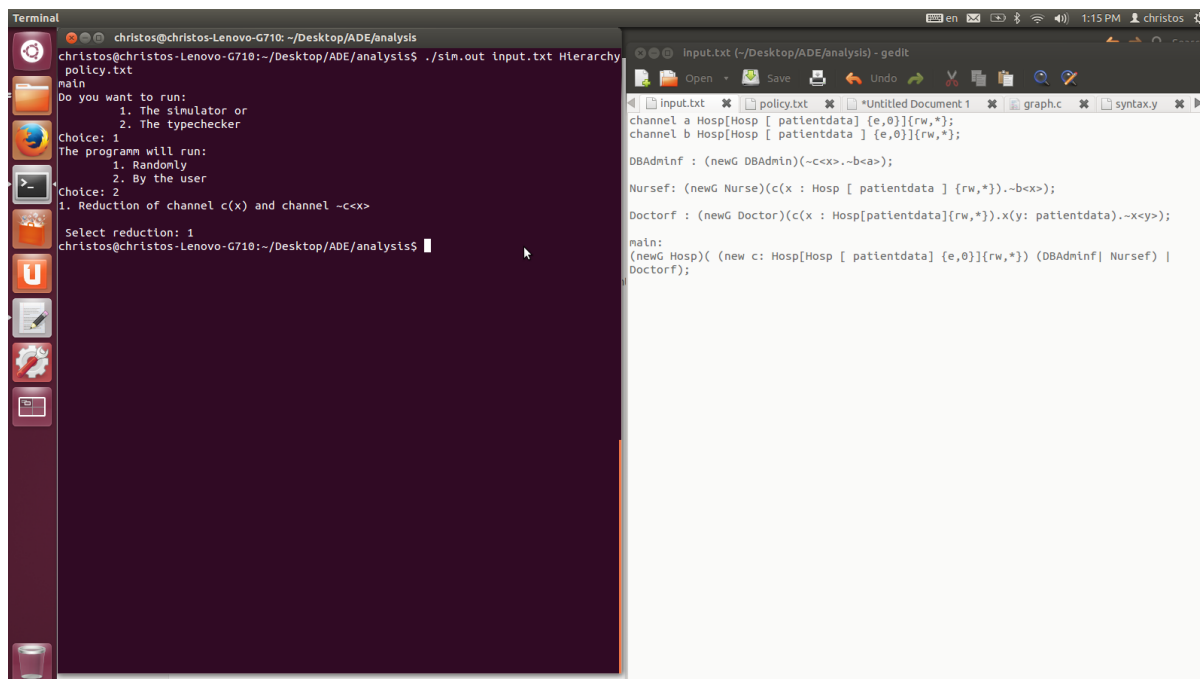
Αρχικά θα δούμε τις ενέργειες του προσομοιωτή:



Σε αυτό το παράδειγμα βλέπουμε ότι στην όγδοη γραμμή του αρχείου συστήματος λείπει ο χαρακτήρας «:» κάνοντας έτσι την γραμματική να αδυνατεί να αναγνωρίσει την είσοδο βγάζοντας έτσι μήνυμα λάθους.

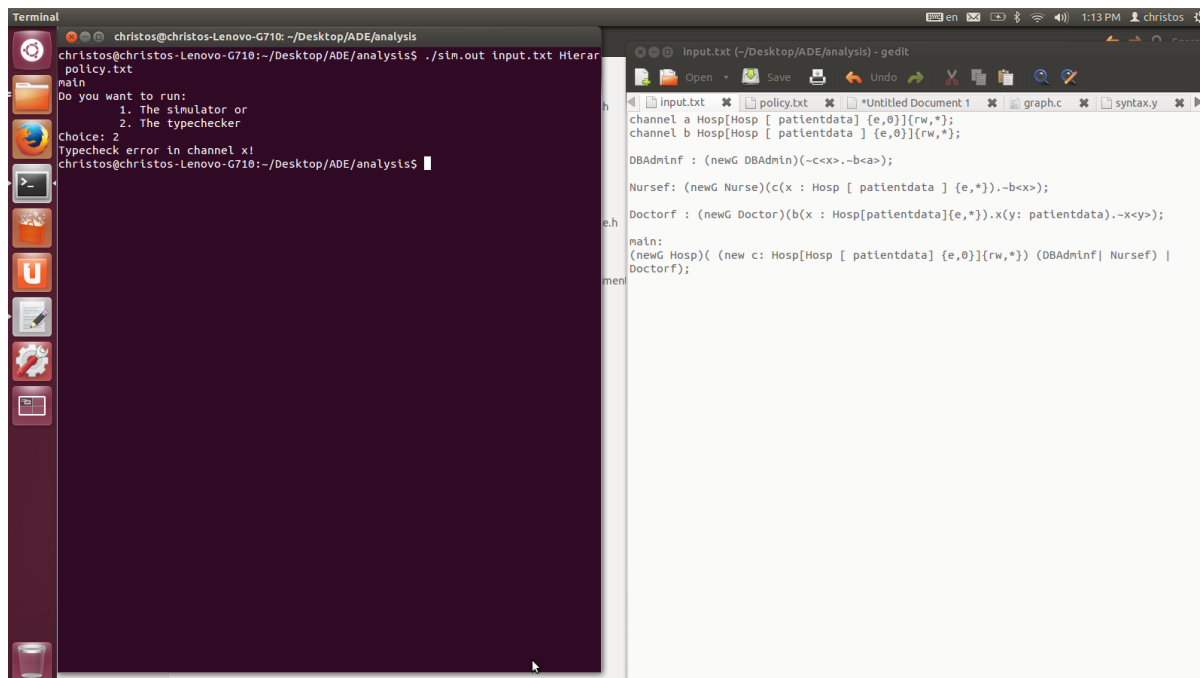


Εδώ βλέπουμε μια σωστή είσοδο. Εντοπίζεται μια μόνο επικοινωνία στο κανάλι x και μετά υπάρχει ακόμα 1 στο κανάλι b αλλά μεταξύ 2 διαφορετικών οντοτήτων.

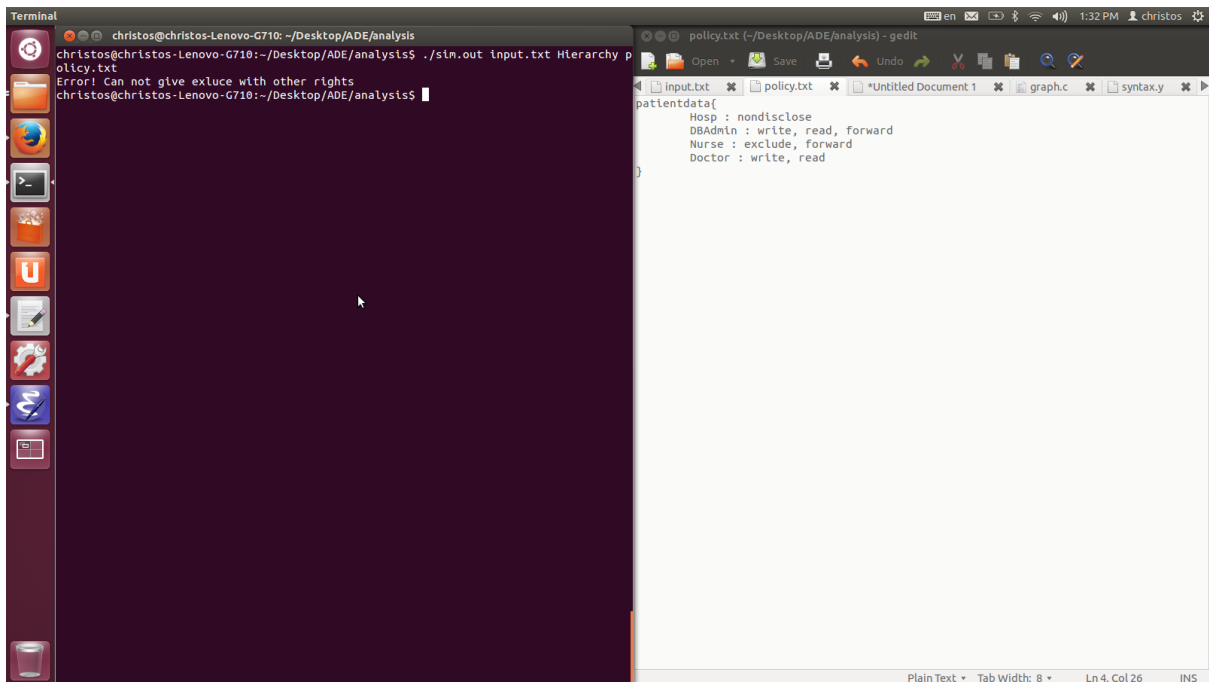


Εδώ παρατηρούμε ότι υπάρχουν 2 είσοδοι στο κανάλι c και μία έξοδος αλλά εντοπίζεται μόνο μια επικοινωνία. Αυτό είναι σωστό διότι το η έξοδος καθώς και μια από τις εισόδους βρίσκονται κάτω από το πεδίο δήλωσης νέου ονόματος (vc).

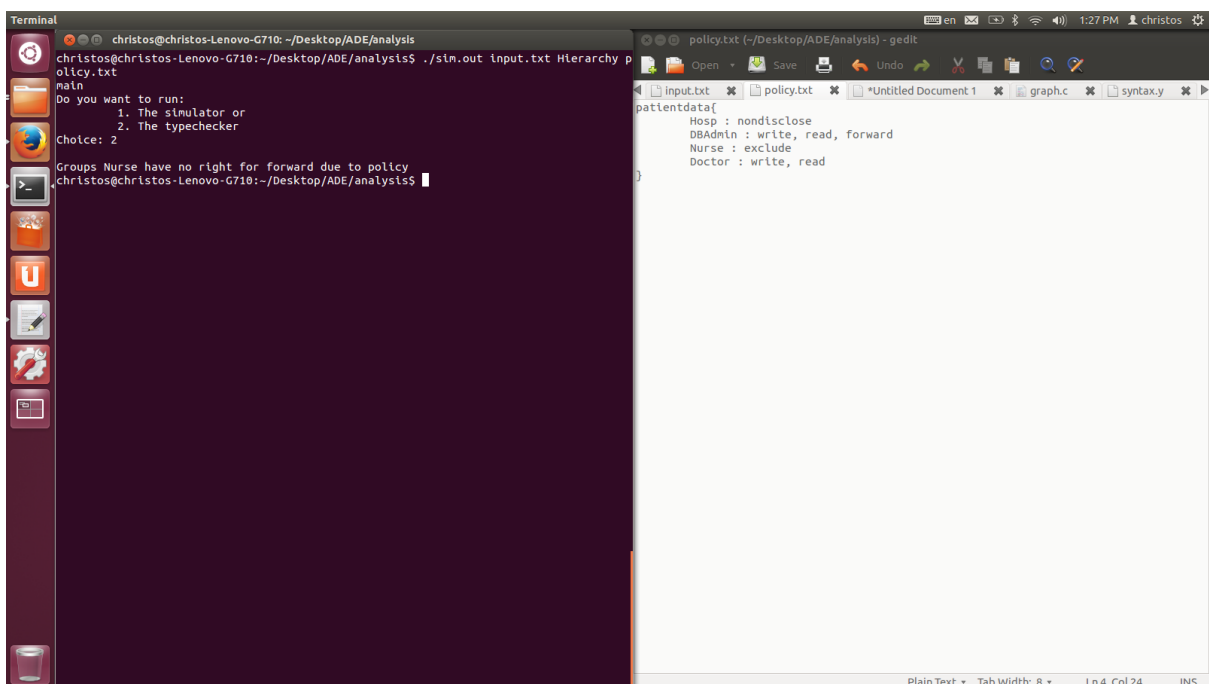
Τώρα θα δούμε τις ενέργειες κατά τον έλεγχο τύπων



Αφαιρέσαμε τα δικαιώματα γραφής και ανάγνωσης από τον τύπο του καναλιού x και αυτό εντοπίστηκε από το σύστημα

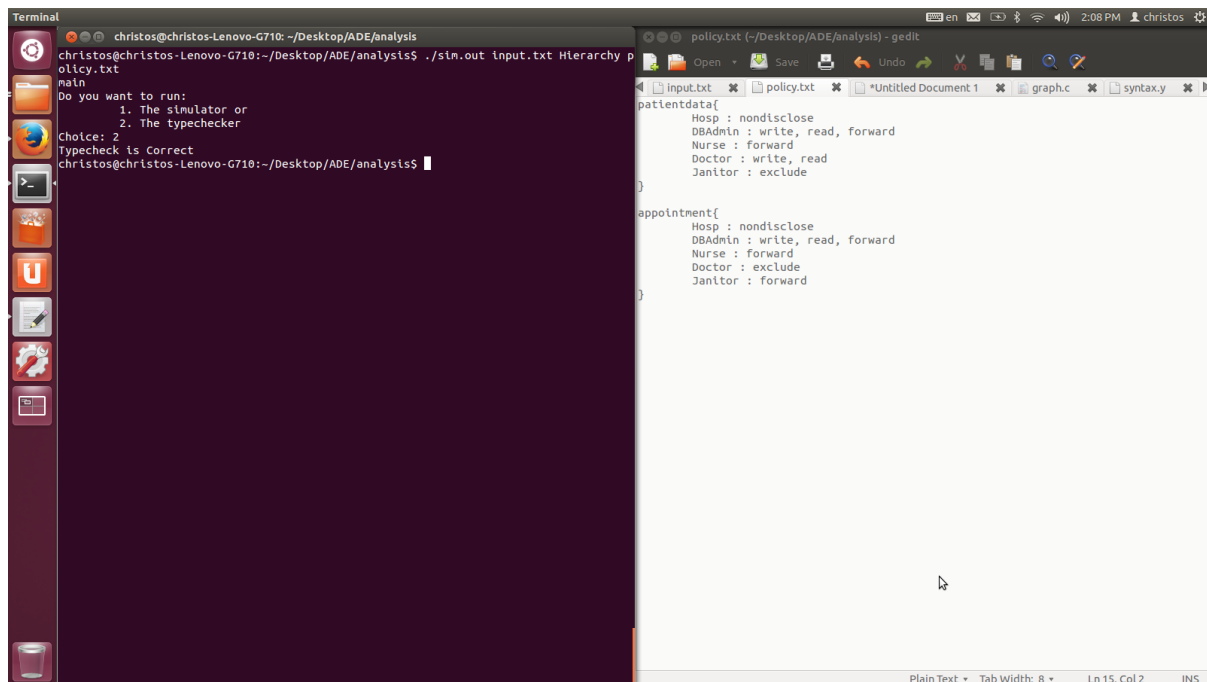


Εδώ φαίνεται μια λάθος εισαγωγή πολιτικής. Δηλώσαμε δικαιώματα αποστολής καθώς και το `exclude` που σημαίνει απουσία δικαιωμάτων. Το σύστημα το εντόπισε.



Εδώ αφαιρέσαμε από την νοσοκόμα το δικαίωμα αποστολής, ενώ η διεργασία της έκανε αποστολή. Αυτό εντοπίστηκε από το σύστημα.

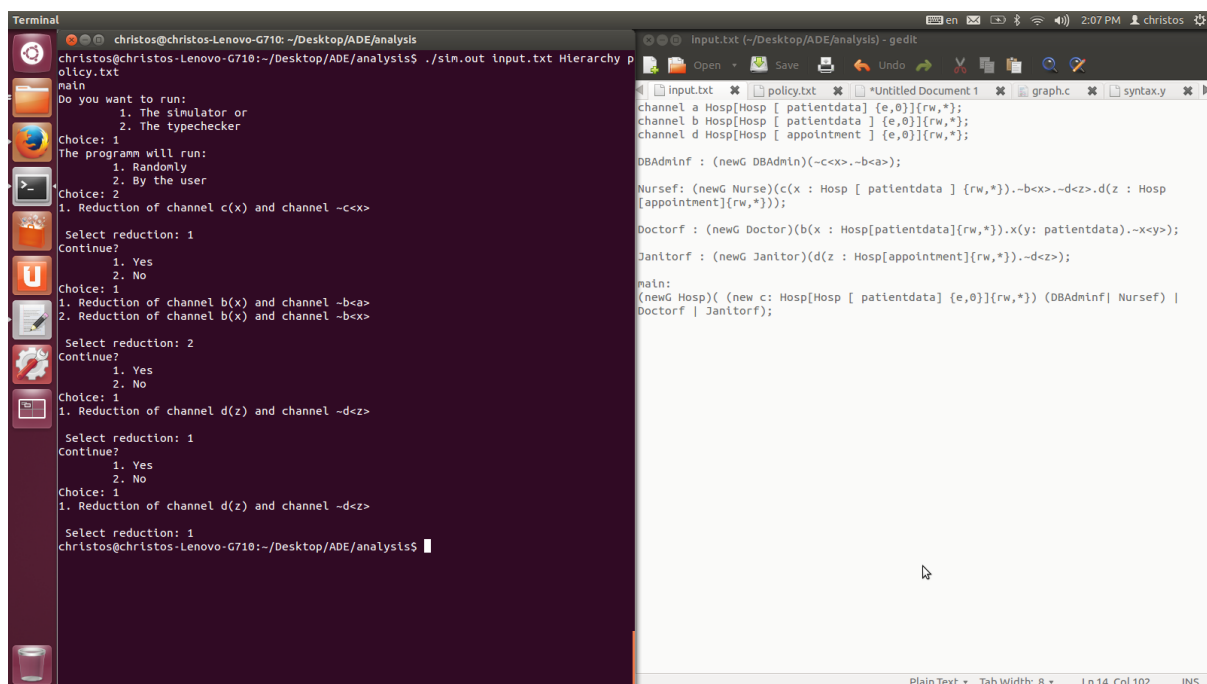
Στην συνέχεια κάναμε την είσοδο πιο πολύπλοκη προσθέτοντας ακόμα ένα βασικό τύπο, τα ραντεβού(appointments), καθώς και τον επιστάτη ο οποίος δεν έχει δικαιώματα στα δεδομένα του ασθενή (patientdata) αλλά μπορεί να προωθήσει τα ραντεβού.



```
Terminal
christos@christos-Lenovo-G710: ~/Desktop/ADE/analysis
christos@christos-Lenovo-G710:~/Desktop/ADE/analysis$ ./sim.out input.txt Hierarchy p
olicy.txt
main
Do you want to run:
  1. The simulator or
  2. The typechecker
Choice: 2
Typecheck is Correct
christos@christos-Lenovo-G710:~/Desktop/ADE/analysis$

policy.txt (-/Desktop/ADE/analysis) - gedit
input.txt  policy.txt  *Untitled Document 1  graph.c  syntax.y
patientdata{
  Hosp : nondisclose
  DBAdmin : write, read, forward
  Nurse : forward
  Doctor : write, read
  Janitor : exclude
}
appointment{
  Hosp : nondisclose
  DBAdmin : write, read, forward
  Nurse : forward
  Doctor : exclude
  Janitor : forward
}
```

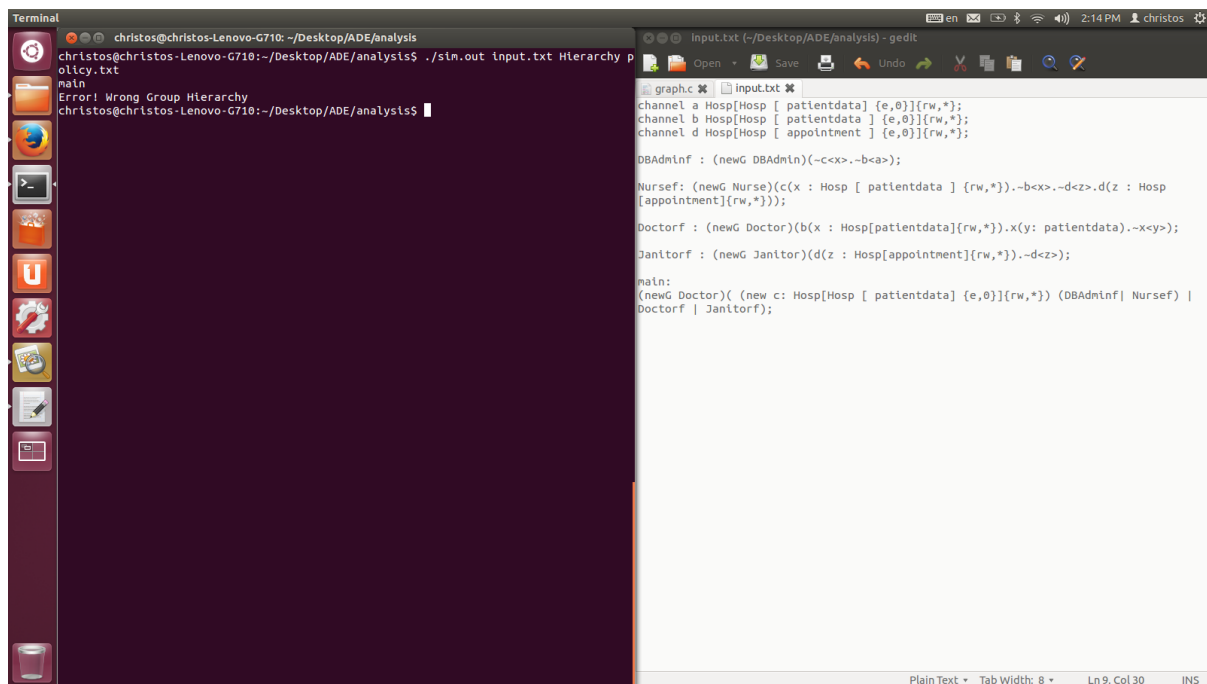
Το σύστημα αναγνώρισε την είσοδο.



```
Terminal
christos@christos-Lenovo-G710: ~/Desktop/ADE/analysis
christos@christos-Lenovo-G710:~/Desktop/ADE/analysis$ ./sim.out input.txt Hierarchy p
olicy.txt
main
Do you want to run:
  1. The simulator or
  2. The typechecker
Choice: 1
The program will run:
  1. Randomly
  2. By the user
Choice: 2
1. Reduction of channel c(x) and channel -c<x>
Select reduction: 1
Continue?
  1. Yes
  2. No
Choice: 1
1. Reduction of channel b(x) and channel -b<x>
2. Reduction of channel b(x) and channel -b<x>
Select reduction: 2
Continue?
  1. Yes
  2. No
Choice: 1
1. Reduction of channel d(z) and channel -d<z>
Select reduction: 1
Continue?
  1. Yes
  2. No
Choice: 1
1. Reduction of channel d(z) and channel -d<z>
Select reduction: 1
christos@christos-Lenovo-G710:~/Desktop/ADE/analysis$

input.txt (-/Desktop/ADE/analysis) - gedit
input.txt  policy.txt  *Untitled Document 1  graph.c  syntax.y
channel a Hosp[Hosp [ patientdata ] {e,0}]{rw,*};
channel b Hosp[Hosp [ patientdata ] {e,0}]{rw,*};
channel d Hosp[Hosp [ appointment ] {e,0}]{rw,*};
DBAdminf : (newG DBAdmin)(-c<x>.-b<a>);
Nursef : (newG Nurse)(c(x : Hosp [ patientdata ] {rw,*}).-b<x>.-d<z>.d(z : Hosp
[appointment]{rw,*}));
Doctorf : (newG Doctor)(b(x : Hosp[patientdata]{rw,*}).x(y: patientdata).-x<y>);
Janitorf : (newG Janitor)(d(z : Hosp[appointment]{rw,*}).-d<z>);
main:
(newG Hosp)( (new c: Hosp[Hosp [ patientdata ] {e,0}]{rw,*}) (DBAdminf | Nursef) |
Doctorf | Janitorf);
```

Εδώ βλέπουμε ένα παράδειγμα του προσομοιωτή με αυτό το μεγαλύτερο παράδειγμα.



Τέλος , πειράξαμε τις ομάδες κάνοντας τις να μην έχουν συνοχή αφού η μόνη ομάδα «ρίζα» είναι το νοσοκομείο. Εδώ βλέπουμε ότι στην διεργασία main φύγαμε το νοσοκομείο (Hosp) και βάλαμε την ομάδα Doctor δημιουργώντας έτσι λάθος στην ιεραρχία των ομάδων.

Κεφάλαιο 5

Συμπεράσματα

5.1 Συμπεράσματα	57
5.2 Μελλοντικές Εργασίες	58

5.1 Συμπεράσματα

Όταν τελείωσε η υλοποίηση, τρέξαμε τα πρώτα παραδείγματα. Εισάγαμε συστήματα τα οποία ήταν γραμμένα σε πGP-Calculus. Τα αποτελέσματα τα οποία πήραμε ήταν ικανοποιητικά. Το σύστημα ανταποκρίθηκε στις προσδοκίες μας δίνοντας σωστά αποτελέσματα όσον αφορά τον έλεγχο των τύπων. Εντοπίζει τυχόν παρατυπίες που συμβαίνουν με τους τύπους, καθώς και ασυμβατότητες που υπάρχουν μεταξύ των τύπων των καναλιών, τις χρήσεις των καναλιών και της πολιτικής που είναι ορισμένη. Επιπλέον, σε κάθε βήμα, εντοπίζει τις σωστές μεταβάσεις που μπορούν να γίνουν και τις παρουσιάζει στο χρήστη.

Βλέποντας τα αποτελέσματα φτάσαμε στο συμπέρασμα, ότι με την χρήση της θεωρίας που αναλύθηκε στα προηγούμενα κεφάλαια, και με την απαιτούμενη γνώση, μπορούμε να κωδικοποιήσουμε τα συστήματα και να βρούμε, με την χρήση αυτοματοποιημένου συστήματος, αν το σύστημα διατηρεί την ιδιωτικότητα ως προς τους τομείς που μελετήσαμε. Άρα, μπορούμε να πούμε ότι ο αρχικός μας στόχος επιτεύχθηκε.

Επιπλέον, το σύστημα αποτελεί μια στατική μέθοδο για τον έλεγχο της ιδιωτικότητας. Αυτό σημαίνει ότι από τη στιγμή που ένα σύστημα θα περάσει τον έλεγχο, τότε μπορούμε να πούμε ότι η πολιτική αυτού του συστήματος διατηρεί την ιδιωτικότητα.

5.2 Μελλοντικές Εργασίες

Όσον αφορά τις μελλοντικές εργασίες, θα μπορούσε να βελτιστοποιηθεί το σύστημα με πολλούς τρόπους. Αρχικά θα μπορούσε το σύστημα να αποκτήσει ένα γραφικό περιβάλλον για να είναι πιο προσιτή η χρήση του. Επιπλέον, θα μπορούσε να αναπτυχθεί ανεκτικότητα στα λάθη από τον χρήστη. Μέχρι στιγμής, αν ο χρήστης εισάγει λάθος δεδομένα, ή λάθος δομημένα δεδομένα, λαμβάνει μήνυμα λάθους και τερματίζει η εκτέλεση του συστήματος. Θα μπορούσε ως μελλοντική δουλειά, να βρίσκει όλα τα τυχόν λάθη που υπάρχουν, να τα παρουσιάζει και μετά να τερματίζει.

Επιπλέον θα μπορούσε να γίνει επιπλέον έρευνα πάνω στο θέμα για να αναπτυχθούν ομάδες οι οποίες να είναι, με κάποιον τρόπο, δηλαδή να μπορεί ένα κανάλι το οποίο ανήκει σε μια ομάδα G , κάτω από κάποιες συνθήκες, να αλλάξει ομάδα και να ανήκει τελικά σε μια άλλη ομάδα, έστω G' . Αυτό θα ήταν μεγάλη ανάπτυξη για το σύστημα καθώς θα μπορούσε να αναπαραστήσει συστήματα με καλύτερη ακρίβεια.

Μια άλλη βελτιστοποίηση που μπορεί να γίνει, είναι μέσω επέκτασης της γλώσσας να καλυφθούν και άλλες πτυχές της ιδιωτικότητας όπως η ταυτοποίηση και η συνάθροιση.

Βιβλιογραφία

1. Tschantz, M.C. and J.M. Wing, *Formal methods for privacy*. 2009: Springer.
2. Solove, D.J., *A taxonomy of privacy*. University of Pennsylvania Law Review, 2006: p. 477-564.
3. Geambasu, R., et al. *Vanish: Increasing Data Privacy with Self-Destructing Data*. in *USENIX Security Symposium*. 2009.
4. Milner, R., *The polyadic π -calculus: a tutorial*. 1993: Springer.
5. Cardelli, L., G. Ghelli, and A.D. Gordon, *Secrecy and group creation*, in *CONCUR 2000—Concurrency Theory*. 2000, Springer. p. 365-379.
6. Kouzapas, D. and A. Philippou, *A Typing System for Privacy*. BEAT 2: p. 51.