

Ατομική Διπλωματική Εργασία

ΑΣΦΑΛΙΣΤΙΚΕΣ ΠΟΛΙΤΙΚΕΣ

ΚΑΙ

ΕΠΙΧΕΙΡΗΜΑΤΟΛΟΓΙΑ

Παναγιώτα Χαραλάμπους

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2015

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ασφαλιστικές Πολιτικές και Επιχειρηματολογία

Παναγιώτα Χαραλάμπους

Επιβλέπων Καθηγητής

Αντώνης Κάκας

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

Μάιος 2015

Ευχαριστίες

Πρώτα απ' όλα, θέλω να ευχαριστήσω θερμά τον επιβλέποντα της διπλωματικής εργασίας μου, Καθηγητή κ. Αντώνη Κάκα, για την πολύτιμη βοήθειά του αλλά και για τις χρήσιμες συμβουλές καθ' όλη τη διάρκεια της δουλειάς μου. Το αμείωτό του ενδιαφέρον, η καθοδήγηση και η υποστήριξη του λειτούργησαν ως βασικοί πυλώνες πάνω στους οποίους στηρίχθηκε η εργασία μου.

Ένα μεγάλο ευχαριστώ, οφείλω στην Υδρόγειο Ασφαλιστική Εταιρεία Κύπρου και κυρίως σε συγκεκριμένους υπαλλήλους για την συνεργασία και το χρόνο που πρόσφεραν σε μένα σε οποιαδήποτε στιγμή το είχα ζητήσει. Επίσης, ευχαριστώ όλους όσους ήταν δίπλα μου για την κατανόηση που έδειξαν προς εμένα, ιδιαίτερα κατά τη διάρκεια των τελευταίων μηνών της προσπάθειάς μου.

Περίληψη

Η διπλωματική μου εργασία μελετά το θέμα σχετικά με τη σημασία και την έννοια Πολιτικών Ασφάλειας στα Πληροφοριακά Συστήματα καθώς και το σημαντικό ρόλο που μπορούν να παρέχουν σε αποφάσεις που αφορούν ένα οργανισμό ή μια εταιρεία.

Κύρια προσέγγιση αυτού του θέματος, είναι συγκεκριμένα πώς Πολιτικές που αφορούν Ασφαλιστική Εταιρεία θα βοηθήσουν στην εξασφάλιση ορθών αποφάσεων και σε ποιο βαθμό.

Για την επίτευξη του στόχου, έγινε μάθηση χρήσης του συστήματος «Γοργίας», το οποίο μπορεί να μεταφράζει τις Πολιτικές Ασφάλειας μιας εταιρείας ή οργανισμού, σε κανόνες με προτεραιότητες.

Τα κεφάλαια της εργασίας παρουσιάζουν κατά σειρά: μια εισαγωγή στο θέμα των πολιτικών ασφάλειας στις μέρες μας καθώς και την αναγκαιότητά τους. Ακολούθως παρουσιάζεται μια αναλυτική προσέγγιση σχετικά με τις πολιτικές, τις μορφές που μπορούν να πάρουν, τους λόγους ύπαρξής τους, τα χαρακτηριστικά τους, τις προσεγγίσεις αλλά και τους παράγοντες στους οποίους οφείλεται η επιτυχία τους.

Στη συνέχεια, αναφέρεται ο τομέας στον οποίο επικεντρώθηκα για να αναπτύξω και να υλοποιήσω συγκεκριμένες Πολιτικές Ασφάλειας με βάσει συγκεκριμένους όρους και διευκρινήσεις που μου έχουν δοθεί από συγκεκριμένη Εταιρεία.

Ακολουθεί μια αναφορά στο λογικό προγραμματισμό και στις προτεραιότητες. Επίσης αναλύεται το σύστημα «Γοργίας» και το συντακτικό του καθώς και οι απαιτήσεις του συστήματος στο οποίο έγινε η υλοποίηση. Έπεται μια προσπάθεια που πραγματοποιήθηκε για την υλοποίηση κανόνων στο σύστημα του Γοργία καθώς και η παρουσίαση διάφορων παραδειγμάτων με πραγματικά γεγονότα και πληροφορίες που έχουν παρθεί από την Εταιρεία όπως και η αξιολόγηση που έχει γίνει συγκρίνοντάς τα με τα δικά μας αποτελέσματα. Τέλος, ακολουθούν τα συμπεράσματα της διπλωματικής εργασίας και θα προταθούν εισηγήσεις για μελλοντική εργασία.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Εισαγωγή	1
Κεφάλαιο 2	Πολιτικές Ασφάλειας.....	3
	2.1 Η έννοια της Πολιτικής Ασφάλειας	3
	2.1.1 Γιατί εφαρμόζουμε μια Πολιτική Ασφάλειας	4
	2.1.2 Είδη Πολιτικών Ασφάλειας	5
	2.1.3 Μορφές Πολιτικών Ασφάλειας	5
	2.1.4 Περιεχόμενο Πολιτικών Ασφάλειας	6
	2.1.5 Χαρακτηριστικά Πολιτικών Ασφάλειας	8
	2.2 Παράγοντες Επιτυχίας	9
	2.3 Λογικός Προγραμματισμός Χωρίς Άρνηση σαν Αποτυχία	10
	2.4 Γοργίας (Gorgias)	18
	2.5 Σύστημα Γοργίας	18
	2.6 Αναπαράσταση Γνώσης	18
	2.6.1 Σύγκρουση στο Γοργία και Επιχειρηματολογία	23
	2.7 Υπολογισμός Απαντήσεων σε Ερωτήματα	25
Κεφάλαιο 3	Πολιτικές Ασφάλειας στον Τομέα μιας Ασφαλιστικής Εταιρείας.....	27
	3.1 Ασφαλιστικές Πολιτικές	27
	3.2 Όροι Ασφάλισης Αυτοκινήτου	28
	3.3 Όροι Ακύρωσης Ασφαλιστικής Σύμβασης	29
	3.3.1 Επιστροφή Ασφαλιστρών από Ακύρωση Ασφαλιστικής Σύμβασης	30
	3.4 Αποζημίωση σε Περίπτωση Απαίτησης	31
	3.5 Όροι Ασφάλισης Υγείας	32

Κεφάλαιο 4	Αναπαράσταση Πολιτικών Ασφάλειας στο Γοργία.....	34
4.1	Σημασία Μετατροπής Κανόνων στο Γοργία	35
4.1.1	Αναπαράσταση Ασφαλιστικών Κανόνων	35
4.2	Διαγράμματα Ροής	35
4.2.1	Επεξήγηση Διαγραμμάτων Ροής	35
4.3	Μετατροπή σε Κανόνες στο Σύστημα του Γοργία	43
4.3.1	Όροι Ασφάλισης Αυτοκινήτου σε Κανόνες στο Γοργία	43
4.3.2	Όροι Ακύρωσης Ασφαλιστικής Σύμβασης σε Κανόνες στο Γοργία	44
4.3.2.1	Επιστροφή Ασφαλιστρών από Ακύρωση Ασφαλιστικής Σύμβασης σε Κανόνες στο Γοργία	47
4.3.3	Αποζημίωση σε Περίπτωση Απαίτησης σε Κανόνες στο Γοργία	47
4.3.4	Όροι Ασφάλισης Υγείας σε Κανόνες στο Γοργία	51
Κεφάλαιο 5	Σύγκριση – Αξιολόγηση.....	53
5.1	Διαδικασία Αξιολόγησης	53
5.2	Σημασία Αξιολόγησης και Σύγκρισης	54
5.3	Αξιολόγηση Αποτελεσμάτων για Ασφάλιση Αυτοκινήτου	55
5.3.1	Πίνακας Αξιολόγησης για Ασφάλιση Αυτοκινήτου	59
5.4	Αξιολόγηση Αποτελεσμάτων για Ακύρωση Ασφαλιστικής Σύμβασης	59
5.4.1	Πίνακας Αξιολόγησης για Ακύρωση από τον Ασφαλισμένο	62
5.4.2	Πίνακας Αξιολόγησης για Ακύρωση από την Εταιρεία	68
5.5	Αξιολόγηση Αποτελεσμάτων για Αποζημίωση σε Περίπτωση Απαίτησης	69
5.5.1	Πίνακας Αξιολόγησης για Αποζημίωση σε Περίπτωση Απαίτησης	81
5.6	Αξιολόγηση Αποτελεσμάτων για Ασφάλιση Υγείας	83

5.6.1 Πίνακας Αξιολόγησης για Ασφάλιση Υγείας	89
Κεφάλαιο 6 Συμπεράσματα και Μελλοντική Εργασία.....	90
6.1 Συμπεράσματα	90
6.2 Μελλοντική Εργασία	92
Βιβλιογραφία.....	93
Παράρτημα Α.....	Α-1
Παράρτημα Β.....	Β-1
Παράρτημα Γ.....	Γ-1
Παράρτημα Δ.....	Δ-1

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγή

3

1.1 Εισαγωγή

Στις μέρες μας, οι οργανισμοί και οι εταιρείες λειτουργούν σε έναν κόσμο υψηλής πολυπλοκότητας και διασύνδεσης και βασίζονται στα πληροφοριακά συστήματα προκειμένου να εκπληρώσουν τους στόχους τους και να διεκπεραιώσουν επιχειρηματικές διεργασίες.

Είναι προφανές ότι οι Πολιτικές Ασφάλειας παίζουν αρκετά σημαντικό και κύριο ρόλο σε ένα οργανισμό/εταιρεία. Πολλές εταιρείες σήμερα έχουν ως βασικό στόχο τη σωστή χρήση των Πολιτικών Ασφάλειας έτσι ώστε να εξασφαλίσουν ότι θα παίρνονται οι σωστές αποφάσεις οι οποίες αφορούν λεπτά και σημαντικά ζητήματα, τα οποία έχουν να κάνουν με τα άτομα που συνεργάζονται, δηλαδή τους πελάτες τους.

Οι Πολιτικές Ασφάλειας παρουσιάζονται συνήθως ως ένα σύνολο κανόνων και οδηγιών που οριοθετούν και οργανώνουν εσωτερικές διαδικασίες.

Μια Πολιτική Ασφάλειας αναπτύσσεται και καθορίζεται βασιζόμενη σε προκαθορισμένους κανόνες και μεθοδολογίες σε συνάρτηση με την εκάστοτε περίπτωση που έχουμε να αντιμετωπίσουμε.

Ένας οργανισμός/εταιρεία προκειμένου να έχει την δυνατότητα να επιτυγχάνει τους στόχους του/της θα πρέπει, μεταξύ άλλων, να διασφαλίσει όσο το δυνατό καλύτερα τις κατάλληλες Πολιτικές Ασφάλειας οι οποίες θα αντικατοπτρίζουν τους όρους του οργανισμού ή της εταιρείας, ούτως ώστε να παίρνονται οι σωστότερες αποφάσεις που

αφορούν διάφορα σημαντικά ζητήματα. Η εφαρμογή ενός σχεδίου ασφάλειας σήμερα, σύμφωνα με διεθνή πρότυπα και πρακτικές, αντιμετωπίζεται σαν μία σημαντική διαχειριστική λειτουργία και όχι απλά ως μία τεχνική λειτουργία.

Προκειμένου να αναπτύξουμε μία πολιτική η οποία θα είναι πλήρης και ολοκληρωμένη, θα πρέπει να ακολουθήσουμε κάποια προκαθορισμένα στάδια.

Το αρχικό στάδιο απαιτεί το διάλογο και τη συζήτηση με τον οργανισμό/εταιρεία, και πιο συγκεκριμένα, με την διοίκηση προκειμένου να καθοριστεί ο γενικός στόχος της πολιτικής ασφάλειας. Κρίσιμο ζήτημα είναι να υπάρξει η δέσμευση της διοίκησης για διάθεση των απαιτούμενων πόρων, υποστήριξη, αλλά και αλλαγή ή προσαρμογή της νοοτροπίας και φιλοσοφίας εργασίας. Επίσης είναι αναγκαίο να προσδιοριστεί η έκταση εφαρμογής της πολιτικής, καθώς είναι απαραίτητο να καθοριστούν τα τμήματα ή οι δραστηριότητες στις οποίες θα γίνει η εφαρμογή. Μέσω αυτών των διεργασιών η εταιρεία ουσιαστικά στοχεύει στην προστασία των κεφαλαίων της, την προστασία της παραγωγικής διαδικασίας, τη διαφύλαξη των εταιρικών και μη πληροφοριών και τέλος τη προστασία των πελατών και των συνεργατών με τον καλύτερο δυνατό τρόπο. Το επόμενο βήμα μας, περιλαμβάνει τον καθορισμό των πολιτικών που θα ακολουθηθούν, δηλαδή αν θα υπάρχει μία γενικευμένη πολιτική ή θα σχεδιαστούν υπό-πολιτικές οι οποίες θα είναι εξειδικευμένες σε ένα τομέα, και όλες μαζί με την ιεραρχία που θα οριστεί θα περιλαμβάνονται στην βασική πολιτική.

Η πολιτική ασφάλειας που θα σχεδιαστεί θα περιλαμβάνει στοιχεία, σε αυτοτελής υπό-πολιτικές έτσι ώστε να είναι εφικτή η εξειδίκευση σε κάθε θέμα αλλά και παράλληλα να είναι πιο ευέλικτες ως προς μελλοντικές αλλαγές και προσθήκες.

Στόχος της παρούσας διπλωματικής εργασίας, είναι ότι έχοντας κάποιους κανόνες γραμμένους σε φυσική γλώσσα, να προσπαθήσουμε να μετατρέψουμε τους κανόνες αυτούς σε Πολιτικές οι οποίες να τους αντιπροσωπεύουν κατάλληλα χωρίς να παραβιάζουν καθόλου τη σημασία τους. Με λίγα λόγια, σκοπός είναι η αυτοματοποίηση Πολιτικών Ασφάλειας, έτσι ώστε να δίνουν τα ακριβή αποτελέσματα και να είναι κατανοητές από όλους τους χρήστες. [1]

Κεφάλαιο 2

Πολιτικές Ασφάλειας

2.1 Πολιτικές Ασφάλειας	3
2.2 Παράγοντες Επιτυχίας	9
2.3 Λογικός Προγραμματισμός Χωρίς Άρνηση σαν Αποτυχία	10
2.4 Γοργίας (Gorgias)	18
2.5 Σύστημα Γοργία	18
2.6 Αναπαράσταση Γνώσης	18
2.7 Υπολογισμός Απαντήσεων σε Ερωτήματα	25

Εισαγωγή

Σ' αυτό το κεφάλαιο θα γίνει μια εκτενής αναφορά στις πολιτικές ασφάλειας και τη σημαντικότητα στη χρήση τους, καθώς επίσης και τις διάφορες μορφές και χαρακτηριστικά που μπορούν να έχουν. Επιπρόσθετα γίνεται περιγραφή του συστήματος του Γοργία. Ο σκοπός αυτής της αναφοράς είναι για να δούμε σε μεταγενέστερο στάδιο πως θα γίνει η σύνδεση των όρων της εταιρείας με λογικούς κανόνες για την αναπαράσταση των πολιτικών που τα διέπουν.

2.1 Η έννοια της Πολιτικής Ασφάλειας ΠΣ

Η Πολιτική Ασφάλειας των Πληροφοριακών Συστημάτων περιλαμβάνει το σκοπό και τους στόχους της ασφάλειας, οδηγίες, διαδικασίες, κανόνες, ρόλους και υπευθυνότητες που αφορούν την προστασία των ΠΣ ενός οργανισμού.

Οι απαιτήσεις για την ασφάλεια του ΠΣ που πρέπει να ικανοποιεί η Πολιτική Ασφάλειας προέρχονται από όλους τους εμπλεκόμενους στη χρήση και λειτουργία του ΠΣ ενός οργανισμού, όπως είναι οι χρήστες και διαχειριστές των ΠΣ, η διοίκηση του οργανισμού η οποία μπορεί να παίζει σημαντικό ρόλο, οι πελάτες του οργανισμού καθώς και οι νομικές και κανονιστικές διατάξεις που διέπουν τη λειτουργία του.

Η Πολιτική Ασφάλειας διατυπώνεται σε ένα έγγραφο, το οποίο θα πρέπει να γνωρίζουν και να εφαρμόζουν όλοι οι χρήστες μιας εταιρείας. Οι οδηγίες και οι διαδικασίες που περιλαμβάνονται στην Πολιτική Ασφάλειας υλοποιούνται με την εφαρμογή των μέτρων προστασίας ή ασφάλειας.

Η Πολιτική Ασφάλειας μαζί με το σύνολο των μέτρων προστασίας αποτελούν το Σχέδιο Ασφάλειας (Security Plan) για τα πληροφοριακά συστήματα ενός οργανισμού.

2.1.1 Γιατί εφαρμόζουμε μια Πολιτική Ασφάλειας;

Είναι πολύ σημαντικό για ένα οργανισμό/εταιρεία να υπάρχει ένα Σχέδιο Ασφάλειας από Πολιτικές το οποίο θα ακολουθείται πιστά από όλα τα εμπλεκόμενα μέλη του οργανισμού/εταιρείας, το οποίο θα καθορίζει τον τρόπο συμπεριφοράς του/της αλλά και τον αναμενόμενο τρόπο αντιμετώπισης οποιοδήποτε περιστάσεων.

- γιατί χρειαζόμαστε ένα συστηματικό και ολοκληρωμένο πλαίσιο που θα καθοδηγήσει την υλοποίηση των μέτρων ασφάλειας
- γιατί λειτουργεί ως το μέσο για την επικοινωνία των εμπλεκόμενων στα ζητήματα ασφάλειας (χρήστες, διοίκηση, διαχειριστές συστημάτων κλπ.)
- γιατί δε διαθέτουμε απεριόριστους πόρους (σε χρήματα, χρόνο, ανθρώπινο δυναμικό)
- γιατί έτσι θεμελιώνεται η σημασία της ασφάλειας του ΠΣ για όλα τα μέλη του οργανισμού
- γιατί συμβάλλει στη δημιουργία κουλτούρας ασφάλειας
- γιατί υπάρχουν περιπτώσεις στις οποίες αποτελεί νομική υποχρέωση

- γιατί αποτελεί σημαντικό παράγοντα εμπιστοσύνης στις σχέσεις του οργανισμού με συνεργαζόμενους φορείς και πελάτες

2.1.2 Είδη Πολιτικών Ασφάλειας

Οι πολιτικές ασφάλειας μπορούν να διαχωριστούν σε δύο κατηγορίες.

Η πρώτη κατηγορία είναι οι Τεχνικές Πολιτικές Ασφάλειας (computer-oriented) που χωρίζονται στις πιο κάτω υποκατηγορίες

- Πολιτικές Ασφάλειας Πληροφοριών οι οποίες υλοποιούν συγκεκριμένους κανόνες πρόσβασης στα δεδομένα, όπως διακριτό έλεγχο προσπέλασης (Discretionary Access Control) ή υποχρεωτικό έλεγχο προσπέλασης (Mandatory Access Control).
- Πολιτικές Ασφάλειας Λειτουργικών Συστημάτων
- Πολιτικές Ασφάλειας Δικτύων Υπολογιστών

Η δεύτερη κατηγορία είναι οι Οργανωσιακές Πολιτικές Ασφάλειας (human-oriented) που έχουν ως υποκατηγορία τις:

- Πολιτικές Ασφάλειας Πληροφοριακών Συστημάτων

2.1.3 Μορφές Πολιτικών Ασφάλειας

Η μορφή και ο τρόπος εφαρμογής μιας Πολιτικής μπορεί να διαφοροποιηθούν ανάλογα με τον οργανισμό και το ΠΣ. Μπορούμε να πούμε πως οι Πολιτικές Ασφάλειας μπορούν να πάρουν τρεις ξεχωριστές μορφές οι οποίες αναλύονται και πιο κάτω μαζί με τις υποκατηγορίες τους.

- Ατομικές Πολιτικές Ασφάλειας (individual security policies): Ανά σύστημα ή εφαρμογή (π.χ. Πολιτική Ασφάλειας για τη χρήση του e-mail)

- Αποσπασματική διαχείριση της ασφάλειας ΠΣ, μεγάλη πολυπλοκότητα στη συντήρηση των πολιτικών
- Αποτελεσματικές όταν υπάρχουν αυτόνομες εφαρμογές και υπολογιστικά συστήματα που δε συνδέονται μεταξύ τους
- Αναλυτικές Πολιτικές Ασφάλειας (Comprehensive Security Policies)
 - Ενιαίο έγγραφο που αναφέρεται σε όλα τα υπολογιστικά συστήματα, τις εφαρμογές και τις διαδικασίες του ΠΣ
 - Είναι μεγάλες σε όγκο, όχι πολύ εύχρηστες
 - Οι οδηγίες και διαδικασίες που περιλαμβάνονται είναι σε γενικό επίπεδο, χωρίς λεπτομέρειες
 - Αρθρωτές Πολιτικές Ασφάλειας (Modular Security Policies)
 - Ενιαίο έγγραφο με παραρτήματα που περιγράφουν τις επιμέρους πολιτικές
 - Μπορεί να είναι σε μορφή υπερκειμένου (hypertext)

2.1.4 Περιεχόμενο Πολιτικών Ασφάλειας

Θα ήταν πιο σωστό και ολοκληρωμένο οι Πολιτικές Ασφάλειας να έχουν ένα σωστό και ολοκληρωμένο περιεχόμενο. Οι οδηγίες και τα μέτρα προστασίας που καθορίζει η πολιτική ασφάλειας ΠΣ θα πρέπει να καλύπτουν τις ακόλουθες κατηγορίες απαιτήσεων ασφάλειας: Ζητήματα προσωπικού, φυσική ασφάλεια, έλεγχος πρόσβασης στο ΠΣ, διαχείριση υλικού και λογισμικού, νομικές υποχρεώσεις, διαχείριση της πολιτικής ασφάλειας, οργανωτική δομή, σχέδιο συνέχισης λειτουργίας.

Αρχικά τα ζητήματα προσωπικού έχουν ως στόχο τις οδηγίες και τα μέτρα ασφάλειας που ανήκουν σε αυτή την κατηγορία που είναι η μείωση της επικινδυνότητας που οφείλεται σε ανθρώπινα λάθη, απάτη, κλοπή ή κατάχρηση των πόρων του ΠΣ. Αναφέρονται κυρίως σε ρόλους και υπευθυνότητες για την προστασία των αγαθών του ΠΣ, διαδικασίες επιλογής νέου προσωπικού, εκπαίδευση και ενημέρωση των χρηστών αλλά και αντιμετώπιση και αναφορά περιστατικών παραβίασης της ασφάλειας.

Η φυσική ασφάλεια αφορά τα μέτρα που έχουν ως κύριο στόχο την αποτροπή της μη εξουσιοδοτημένης πρόσβασης στους χώρους του ΠΣ και της καταστροφής των αγαθών του. Αναφέρονται κυρίως σε έλεγχο φυσικής πρόσβασης σε κρίσιμους χώρους (π.χ. Server room), προστασία της υγείας των χρηστών.

Στη συνέχεια ο έλεγχος πρόσβασης αναφέρεται στην πρόσβαση των χρηστών του ΠΣ στις πληροφορίες, τα υπολογιστικά συστήματα και τις εφαρμογές που θα πρέπει να καθορίζονται με βάση τις επιχειρηματικές ανάγκες και τις απαιτήσεις ασφάλειας.

Η διαχείριση Υλικού και Λογισμικού είναι η Προμήθεια και Συντήρηση Υλικού δηλαδή, οδηγίες που στοχεύουν στη διατήρηση του επιθυμητού επιπέδου ασφάλειας, προσδιορίζοντας τις διαδικασίες για την αγορά και τη συντήρηση του υλικού (π.χ. απαίτηση προμήθειας πιστοποιημένων προϊόντων), και η Ανάπτυξη και Συντήρηση Λογισμικού, δηλαδή οι οδηγίες που περιλαμβάνονται πρέπει να καλύπτουν τις ακόλουθες περιπτώσεις όπως η αγορά έτοιμων προϊόντων (πακέτων λογισμικού) από εξωτερικούς προμηθευτές, η ανάπτυξη και συντήρηση λογισμικού από αναδόχους και η εσωτερική ανάπτυξη και συντήρηση των εφαρμογών.

Οι Νομικές Απαιτήσεις που αναφέρονται συμμόρφωση με το νομικό και κανονιστικό πλαίσιο

Οι Διαδικασίες Διαχείρισης της Πολιτικής Ασφάλειας. Εδώ θα πρέπει να προσδιορίζονται οι απαιτούμενες δραστηριότητες για την εφαρμογή, που αφορούν, την αξιολόγηση και αναθεώρηση της Πολιτικής και τον έλεγχο εφαρμογής και τον καθορισμό των ενεργειών που προβλέπονται στις περιπτώσεις μη τήρησής από τους χρήστες.

Στα περιεχόμενα των Πολιτικών Ασφάλειας περιλαμβάνεται και η Οργανωτική Δομή η οποία για να εφαρμοστεί θα πρέπει να υπάρχει η αντίστοιχη οργανωτική και διοικητική δομή, δηλαδή η δημιουργία των κατάλληλων ρόλων και κατανομή υπευθυνοτήτων καθώς επίσης και η δημιουργία διαδικασιών για τον εντοπισμό και την αναφορά περιστατικών παραβίασης της ασφάλειας.

Τέλος στην κατηγορία που αναφέρεται στο Σχέδιο Συνέχισης Λειτουργίας, πρέπει να περιλαμβάνονται οι οδηγίες που αφορούν τις απαιτούμενες ενέργειες μετά την

πραγματοποίηση ενός σημαντικού περιστατικού παραβίασης της ασφάλειας, ώστε οι λειτουργίες του οργανισμού να εξακολουθήσουν να πραγματοποιούνται με κάποιους εναλλακτικούς τρόπους, έως ότου αντιμετωπιστεί το πρόβλημα ασφάλειας του ΠΣ.

2.1.5 Χαρακτηριστικά Πολιτικών Ασφάλειας

Η πολιτική ασφαλείας μπορεί να είναι είτε ενιαία, ώστε να καλύπτει όλα τα πληροφοριακά συστήματα και τις διαδικασίες που άπτονται της επεξεργασίας δεδομένων προσωπικού χαρακτήρα, είτε να αποτελείται από τμήματα όπου το κάθε ένα να αναφέρεται σε κάποιο υπό-σύστημα επεξεργασίας δεδομένων προσωπικού χαρακτήρα ή επιμέρους τομέα ασφαλείας (όπως επιμέρους πολιτική για κάποιο συγκεκριμένο τομέα ασφαλείας, κλπ.). Στη δεύτερη περίπτωση, οι επιμέρους πολιτικές αποτελούν παραρτήματα της γενικότερης πολιτικής ασφαλείας και μνημονεύονται σε αυτή.

Όταν αναπτύσσουμε μια Πολιτική Ασφάλειας, επιδιώκουμε αρχικά την πληρότητα, την επικαιρότητα αλλά και την γενικευσιμότητα. Οι οδηγίες και τα μέτρα προστασίας πρέπει να καλύπτουν το σύνολο των αγαθών του ΠΣ και όλες τις λειτουργίες του, όπου αυτό αφορά την πληρότητα, να λάβουμε υπόψη τη σύγχρονη τεχνολογία και τις διαφαινόμενες εξελίξεις όσον αφορά την επικαιρότητα και με κάποιες τροποποιήσεις ή προσθήκες να μπορεί η Πολιτική να καλύπτει μικρές αλλαγές ή επεκτάσεις στο ΠΣ δηλαδή να είναι γενικεύσιμη.

Εκτός από τα πιο πάνω κύρια χαρακτηριστικά, επίσης είναι πολύ σημαντικό να λαμβάνουμε υπόψη μας πρέπει να λαμβάνουμε υπόψη ότι μια Πολιτική Ασφάλειας απευθύνεται στο σύνολο των μελών του οργανισμού, και θα πρέπει να είναι εύκολα κατανοητή από όλους δηλαδή να προσφέρει σαφήνεια και ευκολία στην κατανόησή της. Η τεχνολογική ανεξαρτησία που στοχεύει στην περιγραφή των μέτρων ασφαλείας, τα οποία μέτρα δε θα πρέπει να δεσμεύουν τον οργανισμό σε συγκεκριμένα προϊόντα και τεχνολογίες. Εξίσου σημαντική μπορεί να θεωρηθεί η καταλληλότητα, δηλαδή η κάλυψη αναγκών του οργανισμού και τέλος τα μέτρα προστασίας θα πρέπει να μπορούν να εφαρμοστούν χωρίς να δυσχεραίνουν δυσανάλογα τις δραστηριότητες των χρηστών του

ΠΣ, με λίγα λόγια οι Πολιτικές Ασφάλειας θα πρέπει να προσφέρουν και εφαρμοσιμότητα.[3]

2.2 Παράγοντες Επιτυχίας

Ως συμπέρασμα από τα πιο πάνω μπορούμε να πούμε ότι το περιεχόμενο, η μορφή και ο τρόπος εφαρμογής μιας Πολιτικής Ασφάλειας μπορεί να διαφοροποιηθούν ανάλογα με τον οργανισμό, τις απαιτήσεις και τις υπάρχουσες ανάγκες του οργανισμού καθώς επίσης και το ΠΣ. Μια Πολιτική Ασφάλειας επιτυγχάνει καλύτερα τους στόχους της όταν υποστηρίζει τους επιχειρηματικούς στόχους του οργανισμού, η ανώτερη διοίκηση του οργανισμού υποστηρίζει και συμμετέχει ενεργά στην εφαρμογή της, είναι κατάλληλη για το συγκεκριμένο περιβάλλον όπου εφαρμόζεται και πληρεί όλες τις απαιτούμενες προδιαγραφές καθώς επίσης και αν υποστηρίζεται από ένα κατάλληλο σύστημα που μπορεί να υποστηρίξει κατάλληλα τις πολιτικές της Εταιρείας. Επιπρόσθετα σημαντικό είναι οι χρήστες να εκπαιδεύονται και να ενημερώνονται κατάλληλα έτσι ώστε να μπορούν να ανταποκριθούν και να χρησιμοποιούν το σύστημα με σιγουριά και ορθότητα.

Το περιεχόμενο και οι διαδικασίες εφαρμογής της Πολιτικής Ασφάλειας θα πρέπει να αναθεωρούνται σε τακτικά χρονικά διαστήματα και έπειτα από σημαντικά περιστατικά παραβίασης των πολιτικών, θα πρέπει να γίνονται και έκτατες αναθεωρήσεις.

Χρήσιμο είναι να υπάρχουν διαδικασίες αξιολόγησης της αποτελεσματικότητάς της, ώστε να αναθεωρείται αναλόγως και να εφαρμόζεται σταδιακά, ανάλογα με το βαθμό της αλλαγής που επιφέρει η εφαρμογή της Πολιτικής στις δραστηριότητες των χρηστών.

Για τους πιο πάνω λόγους οι πολιτικές ασφάλειας πρέπει να διέπονται από τις αρχές της εμπιστευτικότητας, της διαθεσιμότητας, της ακεραιότητας και της υπευθυνότητας. Έτσι η αποτελεσματικότητα μιας πολιτικής ασφάλειας έγκειται στις τεχνικές και στα εργαλεία που θα χρησιμοποιούν οι διαχειριστές του συστήματος με τα οποία θα μπορούν να αναλύουν και να εξακριβώνουν την ορθότητα των διαφόρων κανόνων που διέπουν τη συγκεκριμένη πολιτική.

2.3 Λογικός Προγραμματισμός Χωρίς Άρνηση σαν Αποτυχία

Στα πλαίσια του Λογικού Προγραμματισμού χωρίς Άρνηση σαν Αποτυχία τα λογικά προγράμματα είναι μη-μονοτονικές θεωρίες όπου κάθε πρόγραμμα είναι μια ομάδα προτάσεων από τις οποίες επιλέγουμε ένα συνεπές υποσύνολο από το οποίο εξάγουμε συμπεράσματα. Οι προτάσεις αυτές γράφονται σε συνηθισμένη γλώσσα Λογικού Προγραμματισμού (π.χ *prolog*), με τη διαφορά ότι χρησιμοποιείται κλασική άρνηση και όχι Άρνηση σαν Αποτυχία [5].

Έστω ότι έχουμε το πιο κάτω πρόγραμμα:

$\text{fly}(x) \leftarrow \text{bird}(x)$

$\neg \text{fly}(x) \leftarrow \text{penguin}(x)$

$\text{bird}(x) \leftarrow \text{penguin}(x)$

$\text{bird}(\text{Tweety})$

όπου η σχέση προτεραιότητας δίνει στο δεύτερο κανόνα μεγαλύτερη προτεραιότητα από τον πρώτο. Με βάση αυτό το πρόγραμμα μπορούμε να συμπεράνουμε ότι $\text{fly}(\text{Tweety})$ επειδή εξάγεται από τον πρώτο και τελευταίο κανόνα και δεν υπάρχει τρόπος να εξάγουμε το συμπέρασμα $\neg \text{fly}(\text{Tweety})$.

Αν τώρα προσθέσουμε στο πρόγραμμα τον κανόνα $\text{penguin}(\text{Tweety})$, βλέπουμε ότι μπορούμε να παράγουμε και $\text{fly}(\text{Tweety})$ και $\neg \text{fly}(\text{Tweety})$. Όμως το δεύτερο συμπέρασμα είναι ισχυρότερο από το πρώτο, αφού το μέρος του προγράμματος που παράγει το $\neg \text{fly}(\text{Tweety})$ χρησιμοποιεί τον δεύτερο κανόνα ο οποίος - με βάση τη σχέση προτεραιότητας - είναι πιο ισχυρός από τον πρώτο κανόνα, ο οποίος ανήκει στο μέρος του προγράμματος που παράγει το $\text{fly}(\text{Tweety})$.

Ακολουθούν οι βασικοί ορισμοί για το πλαίσιο του Λογικού Προγραμματισμού χωρίς Άρνηση σαν Αποτυχία:

Πρόγραμμα: Ένα πρόγραμμα $(K, <)$ είναι ένα σύνολο κανόνων K και μια σχέση προτεραιότητας $<$ πάνω στους κανόνες K .

Επίθεση: Έστω πρόγραμμα $(K, <)$ και $T, T' \subseteq K$. Το T' *επιτίθεται* (*attacks*) στο T αν υπάρχει $k, T_1 \subseteq T'$ και $T_2 \subseteq T$ τέτοια ώστε :

$$(i) \quad T_1 \vdash_{\min} k \text{ και } T_2 \vdash_{\min} \neg k$$

$$(ii) \quad (\exists r' \in T_1, r \in T_2 \text{ τ.ω } r' < r) \Rightarrow (\exists r' \in T_1, r \in T_2 \text{ τ.ω } r < r').$$

$T \vdash_{\min} k$ σημαίνει ότι το T είναι ένα ελάχιστο σύνολο για το οποίο ισχύει $T \vdash k$.

Έστω πρόγραμμα, στο οποίο οι κανόνες είναι :

$$r_1 : p \leftarrow q$$

$$r_2 : q$$

$$r_3 : \neg p \leftarrow r$$

$$r_4 : r$$

και η σχέση προτεραιότητας είναι κενή.

Το σύνολο $T_1 = \{ p \leftarrow q, q \}$ *επιτίθεται* στο σύνολο $T_2 = \{ \neg p \leftarrow r, r \}$, αφού το T_1 εξάγει το συμπέρασμα p ενώ το T_2 το συμπέρασμα $\neg p$ και δεν υπάρχει κανόνας στο T_2 με μεγαλύτερη προτεραιότητα από κάποιο κανόνα στο T_1 . Ομοίως το σύνολο T_2 *επιτίθεται* στο T_1 για τους ίδιους λόγους. Αν τώρα προσθέσουμε τη προτεραιότητα $r_1 > r_2$, το σύνολο T_1 εξακολουθεί να *επιτίθεται* στο σύνολο T_2 , αλλά το αντίθετο δεν ισχύει. Αυτό συμβαίνει επειδή υπάρχει κάποιος κανόνας στο T_1 (ο r_1) ο οποίος έχει μεγαλύτερη προτεραιότητα από

κάποιο κανόνα του T_2 (τον r_2), ενώ κανένας κανόνας του T_2 δεν έχει μεγαλύτερη προτεραιότητα από κάποιο κανόνα του T_1 .

Συνεπές σύνολο κανόνων: Έστω σύνολο κανόνων T . Το T είναι *συνεπές σύνολο κανόνων* αν για κάθε σταθερό άτομο (*ground literal*) k τ.ω $T \vdash k$ δεν ισχύει $T \vdash \neg k$.

Για παράδειγμα το σύνολο $\{ a, \neg a \leftarrow b \}$ είναι συνεπές σύνολο κανόνων, ενώ το σύνολο $\{ a, \neg a \leftarrow b, b \}$ δεν είναι.

Αποδεκτό: Έστω πρόγραμμα $(K, <)$ και T κλειστό υποσύνολο του K . Το T είναι *αποδεκτό* (*admissible*) αν :

- (i) το T είναι συνεπές σύνολο κανόνων και
- (ii) για κάθε $T' \subseteq K$ αν το T' επιτίθεται στο T τότε το T επιτίθεται στο T' .

Ασθενές Συμπέρασμα: Έστω πρόγραμμα $(K, <)$ και k ένα σταθερό άτομο (*ground literal*). Τότε το k είναι *ασθενές συμπέρασμα* (*credulous conclusion*) του προγράμματος αν το k ισχύει σε κάποιο μέγιστο αποδεκτό υποσύνολο (*maximal admissible set*) του K .

Προφανώς, η έννοια του ασθενούς συμπεράσματος δεν είναι πολύ δυνατή, αφού για να ισχύει κάποιο άτομο ασθενώς αρκεί να ισχύει σε κάποιο μέγιστο αποδεκτό υποσύνολο των κανόνων K του προγράμματος, παρόλο που μπορεί να μην ισχύει σε κάποιο άλλο. Μια πιο ισχυρή έννοια είναι αυτή του *ισχυρού συμπεράσματος* (*sceptical conclusion*).

Ισχυρό Συμπέρασμα: Έστω πρόγραμμα $(K, <)$ και k ένα σταθερό άτομο. Τότε το k είναι *ισχυρό συμπέρασμα* (*sceptical conclusion*) του προγράμματος αν το k ισχύει σε κάθε μέγιστο αποδεκτό υποσύνολο (*maximal admissible set*) του K .

Ακολουθεί η μέθοδος υπολογισμού για το πλαίσιο Λογικού Προγραμματισμού χωρίς Άρνηση σαν Αποτυχία.[5]

Ο υπολογισμός γίνεται με χρήση δύο ειδών παραγωγής (*derivations*) : Τύπου Α και Τύπου Β. Οι παραγωγές Τύπου Α προσπαθούν να αποδείξουν τον αρχικό στόχο ή να αμυνθούν (*defence*) εναντίον των επιθέσεων που δέχονται. Οι επιθέσεις αυτές γίνονται μέσω παραγωγής Τύπου Β.

Αρχικά ξεκινά μια παραγωγή Τύπου Α η οποία προσπαθεί να αποδείξει τον αρχικό στόχο. Κατά τη διάρκεια της απόδειξης μπορεί να ξεκινήσουν παραγωγές Τύπου Β οι οποίες επιτίθενται στην αρχική παραγωγή. Οι παραγωγές Τύπου Β ξεκινούν με ένα κανόνα r , με κεφαλή k , ο οποίος χρησιμοποιήθηκε σε κάποια παραγωγή τύπου Α. Απαραίτητη προϋπόθεση για να ξεκινήσει η παραγωγή Τύπου Β είναι να υπάρχει κάποιος κανόνας r' με κεφαλή $\neg k$, ο οποίος να έχει μεγαλύτερη προτεραιότητα από τον r . Η παραγωγή Τύπου Β θα προσπαθήσει να αποδείξει τον κανόνα αυτό.

Κατά τη διάρκεια της απόδειξης σε μια παραγωγή Τύπου Β, μπορεί να ξεκινήσουν άλλες παραγωγές Τύπου Α που στόχο έχουν να αμυνθούν κατά της επίθεσης από την Τύπου Β παραγωγή. Οι παραγωγές αυτές Τύπου Α ξεκινούν με κάποιο κανόνα s με κεφαλή l που χρησιμοποιήθηκε σε κάποια παραγωγή Τύπου Β και προσπαθούν να αποδείξουν κάποιο κανόνα s' με κεφαλή $\neg l$. Απαραίτητη προϋπόθεση είναι ο κανόνας s' να μην είναι χαμηλότερης προτεραιότητας από τον s .

Αν η αρχική παραγωγή Τύπου Α πετύχει, ενώ καμία παραγωγή Τύπου Β δεν πετύχει τότε ο αρχικός στόχος πετυχαίνει, άρα αποτελεί ασθενές συμπέρασμα της θεωρίας. Όλοι οι κανόνες που χρησιμοποιούνται σε παραγωγές Τύπου Α φυλάγονται και αποτελούν αποδεκτό υποσύνολο της θεωρίας το οποίο αποδεικνύει τον αρχικό στόχο.

Με τον υπολογισμό αυτό δηλαδή, μπορούμε να αποφασίσουμε πότε ένας στόχος αποτελεί ασθενές συμπέρασμα της θεωρίας. Για να είναι ένας στόχος ισχυρό συμπέρασμα της θεωρίας πρέπει να πετύχει ο υπολογισμός για το στόχο αυτό αλλά να αποτύχει ο υπολογισμός για την άρνησή του.

Σημειώνουμε εδώ ότι επιτρέπεται η επαναχρησιμοποίηση κανόνων, χωρίς να χρειάζεται να αποδειχθεί ξανά ο ίδιος κανόνας. Δηλαδή, αν βρισκόμαστε σε κάποια παραγωγή Τύπου A και χρειαστεί να αποδείξουμε κάποιο κανόνα τον οποίο ξανασυναντήσαμε σε κάποια παραγωγή Τύπου A σε προηγούμενο επίπεδο υποθέτουμε ότι αποδείχθηκε και προχωρούμε. Αυτό ισχύει επειδή ο κανόνας αυτός είτε έχει ήδη αποδειχθεί στο επίπεδο όπου τον συναντήσαμε ή θα αποδειχθεί στη συνέχεια για το επίπεδο εκείνο. Έτσι δεν υπάρχει λόγος να ξαναρχίσουμε την απόδειξη του συγκεκριμένου κανόνα.

Ο πιο πάνω υπολογισμός είναι ορθός, δηλαδή κάθε φορά που πετυχαίνει, το σύνολο που επιστρέφει είναι αποδεκτό και ο στόχος που τέθηκε είναι ασθενές συμπέρασμα της θεωρίας, ενώ κάθε φορά που αποτυγχάνει ο στόχος που τέθηκε δεν είναι ασθενές συμπέρασμα της θεωρίας.

Ακολουθεί η μελέτη ενός παραδείγματος.

Έστω πρόγραμμα με τους εξής κανόνες:

$$r_1 : \text{fly}(X) \leftarrow \text{bird}(X)$$

$$r_2 : \neg \text{fly}(X) \leftarrow \text{penguin}(X)$$

$$r_3 : \text{penguin}(X) \leftarrow \text{walkslikepeng}(X)$$

$$r_4 : \neg \text{penguin}(X) \leftarrow \neg \text{flatfeet}(X)$$

$r_5 : \text{bird}(X) \leftarrow \text{penguin}(X)$

$r_6 : \text{bird}(t)$

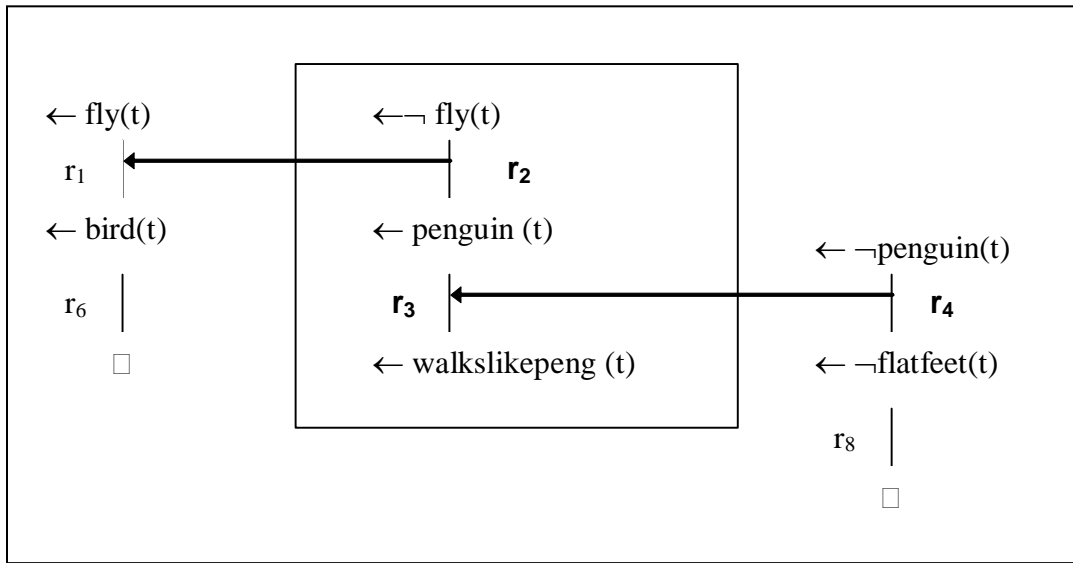
$r_7 : \text{walklikepeng}(t)$

$r_8 : \neg \text{flatfeet}(t)$

και με προτεραιότητες : $r_2 > r_1$ και $r_4 > r_3$.

Έστω ότι ο αρχικός μας στόχος είναι να αποδείξουμε $fly(t)$. Η όλη απόδειξη φαίνεται στο σχήμα 4.1, όπου η παραγωγή στο τετράγωνο είναι Τύπου B, ενώ οι άλλες δύο είναι Τύπου A.

Με βάση τη Μέθοδο Υπολογισμού θα αρχίσει μια παραγωγή Τύπου A, η οποία θα προσπαθήσει να αποδείξει τον στόχο $fly(t)$. Για απόδειξη του $fly(t)$ χρησιμοποιείται ο κανόνας r_1 . Βλέπουμε ότι ο κανόνας r_2 επιτίθεται στον κανόνα r_1 , αφού η κεφαλή του είναι η άρνηση της κεφαλής του r_1 και έχει μεγαλύτερη προτεραιότητα. Έτσι ξεκινά μια παραγωγή Τύπου B που προσπαθεί να αποδείξει την άρνηση του $fly(t)$. Βλέπουμε ότι η επίθεση αυτή δεν πετυχαίνει γιατί υπάρχει μια παραγωγή Τύπου A η οποία αμύνεται κατά της επίθεσης αυτής. Η παραγωγή αυτή προσπαθεί να αποδείξει τον κανόνα r_4 , ο οποίος έχει υψηλότερη προτεραιότητα και αντίθετη κεφαλή από τον κανόνα r_3 που

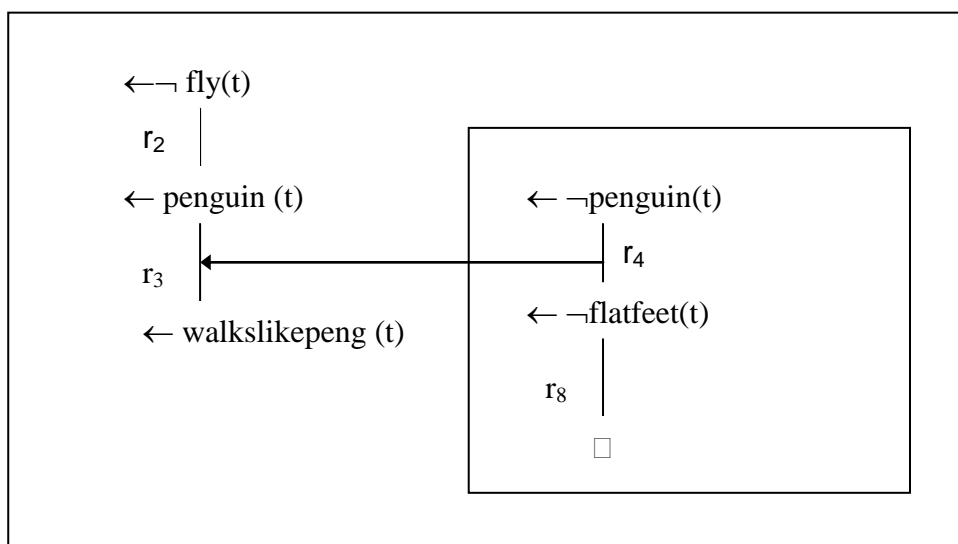


Σχήμα 2.1 Ο υπολογισμός $fly(T)$, τα τόξα συμβολίζουν επίθεση.

χρησιμοποιείται στην επίθεση. Αφού η παραγωγή αυτή πετυχαίνει, άρα η επίθεση απέτυχε. Τέλος αφού η αρχική παραγωγή Τύπου Α πετυχαίνει, το $fly(t)$ αποδεικνύεται. Το σύνολο κανόνων που κατασκευάστηκε είναι : $\{ r_1, r_4, r_6, r_8 \}$.

Έστω τώρα ότι η προτεραιότητα $r_4 > r_3$ δεν υπήρχε. Η παραπάνω απόδειξη παραμένει αληθής, αφού η αναγκαία προϋπόθεση για να μπορέσει κάποιος κανόνας να χρησιμοποιηθεί για να αμυνθεί κατά κάποιας επίθεσης, είναι ο κανόνας αυτός να μην έχει χαμηλότερη προτεραιότητα από τον κανόνα που χρησιμοποιήθηκε στην επίθεση. Άρα στο παράδειγμά μας δεν χρειάζεται ο κανόνας r_4 να έχει ψηλότερη προτεραιότητα από τον r_3 , αλλά απλώς να μην έχει χαμηλότερη προτεραιότητα.

Ας προσπαθήσουμε τώρα να αποδείξουμε την άρνηση του πιο πάνω στόχου, δηλαδή να αποδείξουμε $\neg fly(t)$. Ο υπολογισμός φαίνεται στο Σχήμα 4.2. Όπως βλέπουμε ο υπολογισμός αποτυγχάνει, αφού δεν μπορεί να αμυνθεί κατά της επίθεσης που αρχίζει με τον κανόνα r_4 . Αφού η επίθεση πετυχαίνει και δεν υπάρχει άμυνα, ο αρχικός υπολογισμός αποτυγχάνει, δηλαδή ο στόχος $\neg fly(t)$ δεν είναι ασθενές συμπέρασμα της θεωρίας.



Σχήμα 2.2 Υπολογισμός του $\neg fly(t)$ ο οποίος αποτυγχάνει

Αφού ο στόχος $fly(t)$ αποτελεί ασθενές συμπέρασμα της θεωρίας ενώ η άρνηση του στόχου αυτού ($\neg fly(t)$) αποτυγχάνει, το $fly(t)$ αποτελεί ισχυρό συμπέρασμα της θεωρίας.

Σημειώνεται ότι η εναλλαγή παραγωγών Τύπου A με παραγωγές Τύπου B μπορεί να συνεχιστεί σε μεγαλύτερο βάθος.

2.4 Γοργίας (Gorgias)

Ο Γοργίας ήταν αρχαίος Έλληνας φιλόσοφος, ο οποίος έλεγε ότι τα πάντα στη φύση είναι μια κατασκευή, μια απάτη των ανθρώπινων αισθήσεων. Έτσι δεν υπάρχει αντικειμενική πραγματικότητα και η γνώση για τον κόσμο είναι αδύνατη. Κατά τον Γοργία, ακόμα και να υπήρχε αντικειμενική πραγματικότητα κανείς δεν θα το γνώριζε, αλλά ακόμα και να το γνώριζε δεν θα μπορούσε να το διδάξει στους υπόλοιπους. Οι ανθρώπινες αισθήσεις δεν επιτρέπουν την επικοινωνία με την αντικειμενική πραγματικότητα, αν υποθετικά υπάρχει αυτή. Έτσι, λοιπόν, κάθε γνώση είναι σχετική και όχι η γνώση του όντος, της ουσίας.

2.5 Σύστημα Γοργίας

Το σύστημα Γοργίας (Gorgias) [2] , είναι ένα γενικό πλαίσιο επιχειρηματολογίας που συνδυάζει τις ιδέες του κατά συλλογισμού προτίμησης και της απαγωγής με τρόπο ώστε να συντηρούνται τα οφέλη και των δύο. Μπορεί να αποτελέσει τη βάση για συλλογισμό με προσαρμοζόμενες πολιτικές προτεραιοτήτων εν όψει ατελής πληροφορίας από τα δυναμικά και εξελισσόμενα περιβάλλοντα. Για παράδειγμα, στην αγορά ενός αυτοκινήτου, κάποιος προτιμά ορισμένα χαρακτηριστικά, κάποιος άλλος προτιμά διαφορετικά, στο προγραμματισμό, κάποιες προθεσμίες μπορεί να είναι πιο σημαντικές από κάποιες άλλες κτλ.

2.6 Αναπαράσταση Γνώσης

Για την αναπαράσταση γνώσης , χρησιμοποιούμε σύνθετους όρους στην Prolog [7], έτσι ώστε να αναπαραστήσουμε κανόνες, τις συγκρούσεις και τις προτιμήσεις μεταξύ τους. Η σύνταξη γίνεται με κανόνες της μορφής [6]:

rule(Signature, Head, Body)

όπου το Head είναι ένας σύνθετος όρος, το Body είναι μια λίστα από συνθήκες και το Signature είναι ένας σύνθετος όρος που αποτελείται από το επιθυμητό προσδιοριστικό του κανόνα καθώς επίσης και επιλεγμένες μεταβλητές από τα Head και Body.

Οι αρνητικοί όροι είναι σύνθετοι όροι της μορφής $neg(L)$. Για παράδειγμα, η αναπαράσταση

$rule(r1(X), fly(X), [bird(X)])$.

$rule(r2(X), neg(fly(X)), [penguin(X)])$.

δηλώνει ότι κάτι πετά αν είναι πουλί κι ότι αν κάτι είναι πιγκουίνος δε μπορεί να πετάξει.

Πρέπει να σημειωθεί ότι τόσο οι συνηθισμένοι κανόνες όσο και οι κανόνες με τις προτεραιότητες μπορούν να συνταχθούν με τον ίδιο ακριβώς τρόπο, χρησιμοποιώντας το ειδικό κατηγορημα

$prefer((Label1, Label2)$

που δηλώνει ότι ο κανόνας με προσδιοριστικό Label1 έχει μεγαλύτερη προτεραιότητα από τον κανόνα με προσδιοριστικό το Label2 αν οι συνθήκες στο σώμα του κανόνα ισχύουν.

Ο ρόλος της σχέσης των προτεραιοτήτων είναι να κωδικοποιήσει τοπικά την σχετική βαρύτητα μεταξύ συγκρουόμενων κανόνων.

Για παράδειγμα,

$rule(r1(X), fly(X), [bird(X)])$.

$rule(r2(X), neg(fly(X)), [penguin(X)])$.

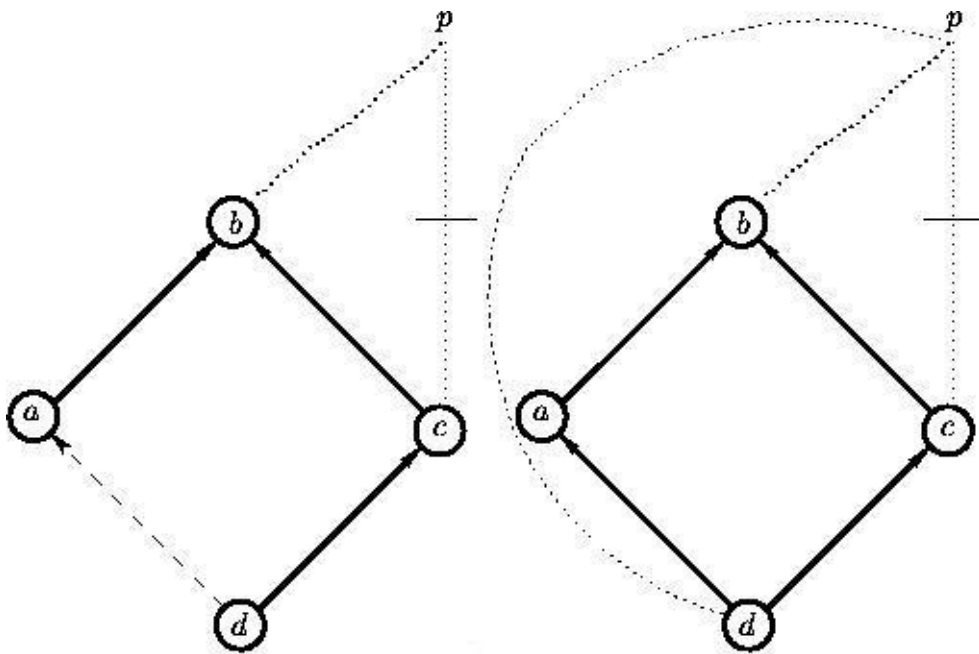
$rule(pr1(X), prefer(r2(X), r1(X)), [])$.

Σ' αυτό το παράδειγμα έχουμε δύο κανόνες όπου κάθε κανόνας αποτελείται από τρία μέρη όπως έχουμε δει: αναγνωριστικό (signature) του κανόνα($r1(X)$), κεφαλή (head) του κανόνα ($fly(X)$) και σώμα(body) του κανόνα($[bird(X)]$). Το πιο πάνω δηλώνει ότι κάτι πετά αν είναι πουλί. Αν είναι πιγκουίνος όμως δεν μπορεί να πετά. Ο κανόνας $rule(pr1(X)$,

$prefer(r2(X), r1(X), [])$ δηλώνει ότι ο $r2(X)$ αποτελεί πιο δυνατή δήλωση από τον $r1(X)$, δηλαδή, αν κάτι είναι πιγκουΐνος και πουλί, θα υπερισχύσει ο κανόνας ότι δεν μπορεί να πετάξει.

Τέτοιοι κανόνες προτίμησης ονομάζονται στατικοί (static) διότι δεν εξαρτώνται από προαπαιτούμενες συνθήκες ενώ οι κανόνες προτίμησης με ένα τουλάχιστο literal στο σώμα τους ονομάζονται δυναμικοί (dynamic).

Στη συνέχεια θα εξετάσουμε ένα παράδειγμα δυναμικών προτιμήσεων. Στο σχήμα 2.1 φαίνεται μια ιεραρχία κληρονομιάς.



Σχήμα 2.3 Ιεραρχία κληρονομιάς

Μια ιεραρχία κληρονομιάς (όπως φαίνεται στο πιο πάνω σχήμα) αποτελείται από ένα άκυκλο γράφο που αναπαριστά τις κατάλληλες σχέσεις των υποκλάσεων των κλάσεων και μια συλλογή από υποκλάσεις που αναφέρονται στις ιδιότητες των αντικειμένων. Τα διακεκομμένα βέλη δείχνουν τις ιδιότητες των αντικειμένων και τα συνεχόμενα βέλη δείχνουν τη πιθανή προσθήκη σχέσης μεταξύ του a και του d το οποίο θα αναλύσουμε αργότερα. Η βασική περιγραφή στο λογικό πλαίσιο Gorgias είναι αυτό που φαίνεται πιο κάτω:

```
rule(f1, subclass(a,b), []).
rule(f2, subclass(c,b), []).
rule(f3, subclass(d,c), []).
rule(f4, is_in(x1,a), []).
rule(f5, is_in(x2,c), []).
rule(f6, is_in(x2,c), []).
rule(d1(X), has(X,p), [is_in(X,b)]).
rule(d2(X), neg(has(X,p)), [is_in(X, c)]).
rule(pr1, prefer(d2(X), d1(X)), []).
```

όπου το $has(X,P)$ είναι για το «στοιχείο X έχει την ιδιότητα P». Ακολουθούν τα ανεξάρτητα αξιώματα για το συγκεκριμένο παράδειγμα:

```
% General properties of subclass and is_in
rule(r1(C0,C2), subclass(C0,C2), [C0 \|= C1, C1 \|= C2, C0 \|= C2,
subclass(C0,C1), subclass(C1,C2)]).
```

rule(r2(X,C1), is_in(X,C1), [subclass(C0,C1), is_in(X,C0)]).

% Closed world assumptions for simple hierarchies

rule(d3(X,C), neg(is_in(X,C)), []).

rule(d4(A,B), neg(subclass(A,B)), []).

Οι δύο πρώτοι κανόνες εκφράζουν γενικές ιδιότητες του subclass και του is_in. Οι τελευταίοι δύο κανόνες εκφράζουν τις απλές ιεραρχίες στα πλαίσια του κλειστού κόσμου σε σχέση με απλές ιεραρχίες κληρονομικότητας και εδώ χρησιμοποιούνται απλά για χάριν του παραδείγματος, δηλαδή για να μπορέσει να προκύψει η σύγκρουση (διαφορετικά, θα ήταν πιο φυσικό αν απλά παραλείπονταν). Είναι εύκολο να ελεχθεί ότι αναφερόμαστε σε ένα συνεπές πεδίο και ότι το λογικό πρόγραμμα μπορεί να κατασκευάσει αποδεκτά επιχειρήματα για το has(x1,p) και για το neg(has(x2,p)).

Υποθέτουμε ότι στο προηγούμενο παράδειγμα, προσθέτουμε αυτό τον κανόνα:

rule(f7, subclass(d,a), []).

Με τον πιο πάνω κανόνα αναπαριστούμε το γεγονός ότι το d επίσης μπορεί να είναι υποκλάση του a και με το κανόνα *rule(d4, prefer(d1(X), d2(X)), [is_in(X,a)]),* εκφράζουμε ότι ο κανόνας d1(X) έχει προτεραιότητα έναντι του κανόνα d2(X) όταν ισχύει το is_in(X,a). Για το x3 μπορεί να αποδειχθεί ότι d1(X) < d2(X) και d2(X) > d1(X) και ότι τα has(x3,p), neg(has(x3,p)) είναι συνέπειες ενός αποδεκτού συνόλου. Ένας τρόπος να επιλυθεί αυτή η σύγκρουση είναι να αναδιατυπώσουμε τους κανόνες d3(X) και d4(X) ως εξής:

rule(d3'(X), prefer(d2(X), d1(X)), [is_in(X,c), neg(is_in(X,a))]).

rule(d4'(X), prefer(d1(X), d2(X)), [is_in(X,a), neg(is_in(X,c))]).

Παρόλαυτα όμως, οι πιο πάνω αναθεωρημένοι κανόνες δεν αντιπροσωπεύουν τις προδιαγραφές της συγκεκριμένης περιοχής με φυσικό τρόπο γιατί μια τέτοια υλοποίηση θα οδηγούσε στην ραγδαία αύξηση στον αριθμό των literals στο σώμα των κανόνων υποβαθμίζοντας έτσι τις υψηλές επιδιώξεις της γλώσσας. Όμως, ο Gorgias μπορεί να δεχτεί τους κανόνες δηλώνοντας τις προτεραιότητες πέρα από τις προτεραιότητες. Για παράδειγμα ο κανόνας :

rule(d5(X), prefer(d4(X), d3(X)), [is_in(X,a)]),

δηλώνει ότι ένα στιγμιότυπο του d έχει την ιδιότητα του p, εκτός από το γεγονός ότι το d είναι και υποκλάση του c.

2.6.1 Σύγκρουση στον Γοργία και Επιχειρηματολογία

Βασικό στοιχείο της επιχειρηματολογίας, είναι η διαφωνία. Η διαφωνία εμφανίζεται στο Γοργία με τη σύγκρουση. Τα συγκρουόμενα επιχειρήματα μπορούν να συνυπάρξουν σε μια βάση γνώσεων και, κατά τη διάρκεια της επιχειρηματολογίας ολοκληρώνουν ένα κατηγορημα με διαφορετικές θέσεις άρνησης.

Ένα απλό παράδειγμα, μιας σχέσης προσόντων, είναι «προτιμά τους μονοτονικούς κανόνες». Διαισθητικά, το «προτιμάς τους μονοτονικούς κανόνες» θέτει την τιμή αλήθειας σε ένα κατηγορημα μονοτονικής επιλογής παρά από ένα κατηγορημα προεπιλογής.

Η σχέση προσόντων αποτελείται από κανόνες της μορφής:

attacks0(?QR, ?LT, +Culprit, +Argument, ?CounterArgument).

όπου, το QR είναι το όνομα μιας συγκεκριμένης περίπτωσης της σχέσης προσόντων , το LT είναι ο τύπος του επιπέδου που μπορεί να ισχύει η σχέση(είτε το 'A' είτε το 'D' για το επίπεδο της επίθεσης ή άμυνας, αντίστοιχα. Το Culprit είναι μια ετικέτα ή μια υπόθεση πάνω στην οποία θα κτίσουμε το CounterArgument. Για παράδειγμα,

attacks0('GEN', _, ass(L), _, CA) :- complement(L, NL), rule(Sig, NL, Body), resolve(Body, Resolvent), CA = [Sig/Resolvent].

δείχνει τη σχέση προσόντων GEN η οποία μπορεί να ισχύει είτε σε επίπεδο επίθεσης είτε σε επίπεδο άμυνας και κατασκευάζει μια σειρά επιχειρημάτων CA υπό τον όρο ότι το δεδομένο επιχείρημα περιέχει μια υπόθεση (το culprit στη περίπτωση μας) ass(L)

Ακολουθεί ακόμα ένα παράδειγμα επιχειρηματολογίας με το οποίο φαίνεται πως ένας στόχος μπορεί να αποδειχθεί επαναλαμβάνοντας τους ίδιους κανόνες.

rule(r1(X), mammal(X), [monotreme(X)]).

rule(r2(X), mammal(X), [hasFur(X)]).

rule(r3(X), neg(mammal(X)), [laysEggs(X)]).

rule(r4(X), neg(mammal(X)), [hasBill(X)]).

rule(r5, monotreme(platypus), []).

rule(r6, hasFur(platypus), []).

rule(r7, laysEggs(platypus), []).

rule(r8, hasBill(platypus), []).

rule(pr1, prefer(r1(_), r3(_)), []).

rule(pr2, prefer(r2(_), r4(_)), []).

rule(pr3, prefer(r4(_), r1(_)), []).

rule(pr4, prefer(r3(_), r2(_)), []).

Το δεύτερο τμήμα των κανόνων αποτελεί τους κανόνες προτίμησης οι οποίοι δηλώνουν ποιος κανόνας έχει περισσότερη ισχύ από έναν άλλο. Για παράδειγμα αν κάνουμε την ερώτηση

?prove([(mammal(platypus))],Delta),

θα πάρουμε την απάντηση

Delta = [nott(pr1), pr1, nott(pr2), r2(platypus), r6, pr2, r5, r1(platypus)] ;

Delta = [nott(pr2), pr2, nott(pr1), r1(platypus), r5, pr1, r6, r2(platypus)] ;

No

Όπως φαίνεται από τις απαντήσεις και στις δύο απαντήσεις χρησιμοποιούμε τους κανόνες r5 και r6.

2.7 Υπολογισμός απαντήσεων σε ερωτήματα

Σε αδρές γραμμές, η κατασκευή μιας λύσης για μια ερώτηση μπορεί να γίνει αυξητικά και σταδιακά, αρχίζοντας από ένα βασικό/αρχικό επιχείρημα το οποίο ενισχύεις στη συνέχεια με άλλα επιχειρήματα. Πιο συγκεκριμένα επικεντρωνόμαστε σε μια ειδική κατηγορία επιχειρημάτων, τα λεγόμενα αποδεκτά επιχειρήματα (admissible). Ένα επιχείρημα είναι αποδεκτό αν μπορεί να υπερασπιστεί τον εαυτό του σε κάθε επίθεση. Ένα επιχείρημα όμως το οποίο επιτίθεται στον εαυτό του δεν μπορεί να είναι αποδεκτό (admissible). Αυτό προκύπτει από το γεγονός ότι δεν μπορεί να υπάρξει επίθεση κατά του κενό συνόλου.

Ο υπολογισμός ενός αποδεκτού επιχειρήματος αποτελεί μια αλληλουχία δύο φάσεων: στη πρώτη φάση ο στόχος επιλύεται και ανάγεται σε ένα κλειστό σύνολο από το οποίο αποδεικνύεται ο στόχος. Στη δεύτερη φάση, το αρχικό επιχείρημα επεκτείνεται με κατάλληλα αμυντικά επιχειρήματα για κάθε επίθεση εναντίον του αρχικού συνόλου. Αλλά, μετά την επέκταση του αρχικού επιχειρήματος, ενδεχομένως να προκύψουν νέες συγκρούσεις και τότε το σύστημα επαναλαμβάνει τη διαδικασία ώσπου δεν υπάρχει άλλη επίθεση εναντίον της βασικής άμυνας.

Τα ερωτήματα υποβάλλονται στο σύστημα ως εξής:

prove (Goals, Delta).

όπου, Goals είναι μια λίστα από θετικούς ή αρνητικούς όρους και Delta η μεταβλητή για ένα αποδεκτό επιχείρημα για τη συγκεκριμένη ερώτηση. Για παράδειγμα ένα αποδεκτό επιχείρημα Delta για το ερώτημα prove([neg(fly(tweety))],Delta) στο πρόβλημα του Tweety που αναλύεται στη συνέχεια είναι Delta = [f2, r2(tweety)].

Το πρόβλημα του Tweety που αναφέρεται στο Tweety ότι μπορεί να πετάξει από τα γεγονότα ότι ο Tweety είναι πουλί και ότι τα πουλιά πετούν, ανακατασκευάζεται από την ανακάλυψη ότι ο Tweety είναι πιγκουΐνος. Το πρόβλημα κανονικοποιείται ως εξής:

rule(r1(X), fly(X), [bird(X)]).

rule(r2(X), neg(fly(X)), [penguin(X)]).

rule(f1, bird(tweety), []).

rule(f2, penguin(tweety), []).

rule(pr1(X), prefer(r2(X), r1(X)), []).

Κεφάλαιο 3

Πολιτικές Ασφάλειας στον Τομέα μιας Ασφαλιστικής Εταιρείας

3.1 Ασφαλιστικές Πολιτικές	27
3.2 Όροι Ασφάλισης Αυτοκινήτου	28
3.3 Όροι Ακύρωσης Ασφαλιστικής Σύμβασης	29
3.4 Αποζημίωση σε Περίπτωση Απαίτησης	31
3.5 Όροι Ασφάλισης Υγείας	32

Εισαγωγή

Στο συγκεκριμένο κεφάλαιο περιγράφεται ο τομέας στον οποίο θα ασχοληθούμε λεπτομερώς για όλο το υπόλοιπο της εργασίας καθώς επίσης και η περιγραφή των διάφορων κατηγοριών ασφάλισης σε φυσική γλώσσα όπου αργότερα θα χρησιμοποιήσουμε για την μετατροπή τους σε Πολιτικές Ασφάλειας στο σύστημα του Γοργία.

3.1 Ασφαλιστικές Πολιτικές

Είναι γνωστό ότι, κάθε εταιρεία στις μέρες μας προσπαθεί όλο και πιο πολύ να αυτοματοποιήσει πολλές από τις λειτουργίες της. Αυτό γίνεται για πολλούς λόγους όπως για παράδειγμα λόγους που αφορούν τον χρόνο, την ευκολία χρήσης ενός συστήματος από τους υπαλλήλους αλλά κυρίως την ευκολία λήψης αποφάσεων όσον αφορά διάφορα ζητήματα για τα οποία, ένα μικρό λάθος μπορεί να επιφέρει μεγάλη χρηματική ζημιά στην εταιρεία.

Ένας από τους τομείς λοιπόν, στον οποίο γίνεται έντονη η ανάγκη για αυτοματοποίηση Πολιτικών Ασφάλειας είναι οι Ασφάλειες με τις οποίες ασχολούνται διάφορες

Ασφαλιστικές Εταιρείες. Η ραγδαία ανάπτυξή τους σήμερα, έχει φέρει στο προσκήνιο το θέμα της δημιουργίας των πολιτικών με την ύψιστη σημασία στη λήψη αποφάσεων.

Σε μια Ασφαλιστική Εταιρεία υπάρχουν πολλά τμήματα για τα οποία προσφέρεται Ασφάλιση για ένα πελάτη, παράλληλα, υπάρχουν και άλλα τόσα θέματα για τα οποία πρέπει να παρθούν οι κατάλληλες αποφάσεις, που κάποιες φορές έχουν να κάνουν με μια απλή απάντηση «ΝΑΙ» ή «ΟΧΙ», και κάποιες άλλες φορές πρέπει να δώσουν σαν απάντηση ένα ποσό, το οποίο βέβαια πρέπει να είναι και σωστό.

Με αυτόν το σημαντικό τομέα Πολιτικών θα ασχοληθούμε για όλο το υπόλοιπο της διπλωματικής εργασίας, όπου θα προσπαθήσουμε, μέσα από τη δύναμη της επιχειρηματολογίας να πετύχουμε τη δημιουργία ανάπτυξης Ασφαλιστικών Πολιτικών.

3.2 Όροι Ασφάλισης Αυτοκινήτου

Ένα από τους πιο σημαντικούς τομείς που προσφέρει ασφάλιση μια Ασφαλιστική Εταιρεία είναι η Ασφάλιση Αυτοκινήτου. Εκτός από σημαντική ασφάλιση, είναι επίσης και αναγκαία, εφόσον κανείς σήμερα δεν μπορεί να έχει αυτοκίνητο και να κυκλοφορά χωρίς να έχει ασφάλεια για το συγκεκριμένο. Για να μπορέσει ένας υποψήφιος πελάτης να αποκτήσει Ασφάλιση Αυτοκινήτου, υπάρχουν κάποιοι όροι και προϋποθέσεις, οι οποίοι πρέπει να πληρούνται.

Ως αρχική προϋπόθεση, ο υποψήφιος πελάτης πρέπει να έχει άδεια οδήγησης, η οποία να καλύπτει απόλυτα ότι ο οδηγός μπορεί να οδηγήσει τη συγκεκριμένη κατηγορία μηχανοκίνητου. Επιπρόσθετα πρέπει να είναι μέλος της Ευρωπαϊκής Ένωσης.

Επίσης, οι πινακίδες εγγραφής του αυτοκινήτου θα πρέπει να είναι απαραίτητα κυπριακές. Η ασφάλιση ισχύει μονάχα για τα όρια της Κυπριακής Επικράτειας, εκτός των Κυπριακών ορίων, η ασφάλιση παύει να είναι σε ισχύ, εκτός αν ο οδηγός εφοδιαστεί με πιστοποιητικό διεθνούς Ασφάλισης, τη λεγόμενη «Πράσινη Κάρτα» η οποία ισχύει για τα κράτη μέλη της Ευρωπαϊκής Ένωσης.

Πολύ σημαντικό ρόλο για ένα υποψήφιο πελάτη ο οποίος θα θελήσει να αποκτήσει Ασφάλιση Αυτοκινήτου, είναι το ποινικό του μητρώο. Το υποψήφιο πρόσωπο, βάσει των όρων και κανόνων θα πρέπει να είναι νόμιμο. Αν έχει φάκελο με «μαύρο» ποινικό μητρώο,

τότε του απαγορεύεται η απόκτηση Ασφάλισης Αυτοκινήτου. Σ' αυτή τη προϋπόθεση εξαιρείται η περίπτωση στη οποία έχουν περάσει περισσότερα από δέκα (10) χρόνια από την ημέρα που έχει γραφεί στο φάκελο του υποψήφιου πελάτη οτιδήποτε θα μπορούσε να του «μαυρίσει» το ποινικό του μητρώο. Για λιγότερα από δέκα χρόνια, ισχύει η πρώτη περίπτωση στην οποία απαγορεύεται η απόκτηση Ασφάλισης Αυτοκινήτου.

Επομένως, αφού ένας υποψήφιος πελάτης αποφασίσει να αποκτήσει Ασφάλιση Αυτοκινήτου, θα πρέπει να ελεγχθεί με βάσει τους πιο πάνω όρους και προϋποθέσεις. Αν πληρεί όλα τα πιο πάνω, τότε μπορεί να αποκτήσει και την Ασφάλιση, και πλέον να λέγεται ασφαλισμένος πελάτης (Insured Person).

3.3 Όροι Ακύρωσης Ασφαλιστικής Σύμβασης

Κάθε πελάτης όπως έχει δικαίωμα να αποκτήσει μια Ασφάλιση, έχει επίσης το δικαίωμα να την ακυρώσει οποιαδήποτε στιγμή και για οποιοδήποτε προσωπικό του λόγο. Αυτό μπορεί να γίνει από την πλευρά του πελάτη. Από την μεριά της η Εταιρεία, έχει κι αυτή το δικαίωμα ακύρωσης μιας Ασφαλιστικής Σύμβασης με γραπτή δήλωση.

Ο Ασφαλισμένος, το άτομο δηλαδή το οποίο έχει αποκτήσει κάποιου είδους Ασφάλιση, οποιαδήποτε στιγμή μπορεί να καταγγείλει, την Ασφαλιστική Σύμβαση στην Εταιρεία ή στον διαμεσολαβητή που τον έχει αναλάβει (agent). Με τον τρόπο αυτό, στέλλεται στην Εταιρεία ή στον διαμεσολαβητή το αίτημα παραπόνου. Μετά την πάροδο τριάντα (30) ημερών από την περιέλευση του αιτήματος στην Εταιρεία, επέρχονται τα αποτελέσματα. Έτσι γίνεται η ακύρωση της Ασφαλιστικής Σύμβασης, από αίτημα του Ασφαλισμένου, εφόσον η Εταιρεία απαντήσει θετικά στο αίτημά του.

Όπως έχουμε προαναφέρει, η Εταιρεία μπορεί με γραπτή δήλωση να καταγγείλει την Ασφαλιστική Σύμβαση, μόνο για παράβαση ουσιώδους όρου από τον Ασφαλισμένο, βαρυνόμενη με την απόδειξη της παράβασης. Η καταγγελία δίνεται στον Ασφαλισμένο και γνωστοποιείται ότι η μη συμμόρφωση του Ασφαλισμένου με τον παραβιασθέντα ουσιώδη

όρο, εντός τριάντα (30) ημερών από την επίδοση της καταγγελίας, επιφέρει την ακύρωση της Ασφαλιστικής Σύμβασης.

Σε περίπτωση όμως, που η Εταιρεία έχει στείλει τη γραπτή καταγγελία στον Ασφαλισμένο, αλλά δεν έχει φτάσει στα χέρια του, τα αποτελέσματα επέρχονται ανεξάρτητα από την άρνηση του Ασφαλισμένου να τα παραλάβει, ή τη μη ανεύρεση της διεύθυνσης της διαμονής του ή τη μη προσέλευση του στο ταχυδρομείο για την παραλαβή της, εκτός εάν ο Ασφαλισμένος αποδείξει ανυπαίτια ότι δεν είχε τη δυνατότητα να λάβει την γραπτή καταγγελία. Τότε, έχοντας αυτά υπόψη, αφού πλέον δεν ισχύει η ακύρωση, όμως τώρα βρίσκεται στα χέρια του ασφαλισμένου η καταγγελία, υπολογίζονται για δεύτερη φορά οι τριάντα (30) ημέρες, για την ίδια διαδικασία από την αρχή.

3.3.1 Επιστροφή Ασφαλιστρών από Ακύρωση Ασφαλιστικής Σύμβασης

Αρχικά, είναι σημαντικό να αναφέρουμε ότι η διάρκεια του Ασφαλιστηρίου Συμβολαίου είναι ένα (1) έτος, με έναρξη από την ημερομηνία που αναγράφεται στο Ασφαλιστήριο Συμβόλαιο. Επίσης για κάθε συμβόλαιο καθορίζεται ένα ποσό, τα λεγόμενα ασφάλιστρα. Τα ασφάλιστρα που θα αναγράφονται στο Συμβόλαιο προσδιορίζονται συνδυαστικά με βάση διάφορους παράγοντες.

Μετά από όλη τη διαδικασία ακύρωσης μιας Ασφαλιστικής Σύμβασης επέρχεται το θέμα επιστροφής ασφαλιστρών στο δικαιούχο. Έχουμε πει ότι, ο Ασφαλισμένος μπορεί οποιαδήποτε στιγμή να ακυρώσει μια Ασφαλιστική Σύμβαση, επομένως, αφού αυτό μπορεί να γίνει μέσα στο χρονικό διάστημα στο οποίο ισχύει η Ασφάλιση, τότε θεωρείται λογικό να πρέπει να επιστραφούν στον Ασφαλισμένο τα μη δεδουλευμένα ασφάλιστρα, για το υπόλοιπο του χρόνου που απομένει.

Αυστηρή προϋπόθεση για την επιστροφή των ασφαλιστρών είναι, ο Ασφαλισμένος να μην έχει στην Εταιρεία φάκελο από ατύχημα. Εάν δηλαδή ο Ασφαλισμένος, έχει στην Εταιρεία προηγούμενο ιστορικό φάκελο ατυχήματος, τότε δε του επιτρέπεται κανένα ποσό επιστροφής ασφαλιστρών κατά τη διαδικασία της ακύρωσης.

Αντίθετα όμως με αυτή τη περίπτωση, υπολογίζεται το ποσό επιστροφής ασφαλιστρών στο δικαιούχο όπως του αρμόζει. Με μια συγκεκριμένη φόρμουλα υπολογισμού, επιστρέφονται

τα μη δεδουλευμένα ασφάλιστρα, για το υπόλοιπο του χρόνου που έχει απομείνει, με βάση το συμβόλαιο.

Έτσι λοιπόν, στον Ασφαλισμένο θα επιστραφεί το Έντυπο Ακύρωσης της Ασφαλιστικής Σύμβασης μαζί με το ποσό των ασφαλίσεων το οποίο και δικαιούται. Το ποσό αυτό θα είναι ανάλογο με το ποσό του συμβολαίου ή σε περίπτωση που έχει υποβληθεί οποιαδήποτε απαίτηση δεν επιστρέφεται κανένα ασφάλιστρο.

3.4 Αποζημίωση σε Περίπτωση Απαίτησης

Η αποζημίωση είναι το ποσό που υποχρεούται να καταβάλει η Εταιρεία στον Ασφαλισμένο σε περίπτωση ζημιάς από καλυπτόμενο κίνδυνο σύμφωνα με τους όρους της Ασφαλιστικής Σύμβασης. Μπορεί να παίξει έναν από τους πιο σημαντικούς λόγους για τους οποίους μπορεί να επιλέξει ένα άτομο να δεσμευτεί με μια Ασφαλιστική Σύμβαση.

Αφού ο Ασφαλισμένος έχει Ασφαλιστική Σύμβαση, η οποία βέβαια είναι σε ισχύ, δηλαδή δεν έχει λήξει, απαραίτητη προϋπόθεση είναι το ατύχημα να έχει γίνει εντός των Κυπριακών Ορίων, ούτως ώστε να έχει το δικαίωμα να αιτήσει αποζημίωση.

Μπορούμε να διαχωρίσουμε την αποζημίωση που θα πάρει ο δικαιούχος σε δύο (2) διαφορετικές μορφές, την ολική (full) και την μέγιστη (maximum) αποζημίωση. Οι μορφές αυτές διαφοροποιούνται με βάση τη ζημιά που θα προκύψει από το ατύχημα σε σχέση με το ποσό που αναγράφεται στο συμβόλαιο του Ασφαλισμένου. Αν το ποσό της ζημιάς που θα προκύψει, είναι μεγαλύτερο από το αναγραφόμενο ποσό του συμβολαίου, τότε ο Ασφαλισμένος μπορεί να πάρει την μέγιστη αποζημίωση. Αλλιώς, αν το ποσό της ζημιάς είναι μικρότερο από το συμφωνημένο ποσό, τότε δικαιούται ολική αποζημίωση.

Σημαντικός παράγοντας για την παροχή αποζημίωσης, είναι το ατύχημα να μη προήλθε από παράνομη οδήγηση. Παράνομη μπορεί να θεωρηθεί η οδήγηση χωρίς άδεια ή με άδεια οδήγησης η οποία έχει λήξει. Επιπρόσθετα, ο οδηγός πρέπει να είναι γραμμένος πάνω στην Ασφαλιστική Σύμβαση ως νόμιμο άτομο το οποίο καλύπτεται να οδηγά το συγκεκριμένο

αυτοκίνητο. Αν ένας από αυτούς τους πιο πάνω λόγους δεν τηρείται, τότε μπορεί να θεωρηθεί παράνομη οδήγηση και ο Ασφαλισμένος να μη μπορεί να πάρει κανενός είδους αποζημίωση.

Επιπλέον, αν η Αστυνομία που θα εξετάσει το δυστύχημα, επιβεβαιώσει ότι ο Ασφαλισμένος οδηγούσε κάτω από την επήρεια αλκοόλ ή ναρκωτικών έχοντας ως αποτέλεσμα την παραβίαση του κώδικα οδικής κυκλοφορίας, ή ακόμα ότι ενέργησε με δόλο και προκάλεσε οποιαδήποτε ζημιά, τότε αυτά καθιστούν επιπρόσθετους λόγους μη παροχής αποζημίωσης στον Ασφαλισμένο.

3.5 Όροι Ασφάλιση Υγείας

«Η υγεία είναι ο πραγματικός πλούτος και όχι το χρυσάφι και το ασήμι»,

Μαχάτμα Γκάντι

Υγεία, το πολυτιμότερο αγαθό που χάρισε η φύση στον άνθρωπο. Το χαρισματικό δημιουργήμα βασίζεται στην υγεία, για να ονειρευτεί, να σχεδιάσει το μέλλον, να προχωρήσει στη ζωή, να κάνει οικογένεια, να υλοποιήσει τα όνειρα του, τις σκέψεις και τους στόχους του.

Ο άνθρωπος κάνει ότι είναι δυνατόν για να προστατέψει την υγεία του. Αν και το τελειότερο ον, δεν μπορεί να την εξασφαλίζει πάντοτε. Η απόλυτη θωράκιση της δεν είναι δυνατή, ο κλονισμός της, μπορεί να είναι καταστροφικός και όχι μόνο βιολογικά. Τα έξοδα αποκατάστασης της, είναι πολλές φορές δυσβάστακτα και απαγορευτικά.

Μια Ασφαλιστική Εταιρεία, μέσα στα πλαίσια της πολιτικής της για προσφορά των καταλλήλων σχεδίων ασφάλισης στον άνθρωπο, προσφέρει λύσεις αναλαμβάνοντας το οικονομικό βάρος, προσφέροντας το ασφαλιστικό πρόγραμμα υγείας.

Απαραίτητες προϋποθέσεις για να αποκτήσει ένα άτομο Ασφάλιση Υγείας, πρέπει να έχει συμπληρώσει το 18^ο έτος της ηλικίας του, αλλά να μην ξεπερνά τα 60 χρόνια (18-60) καθώς επίσης και να είναι μέλος της Ευρωπαϊκής Ένωσης. Αντίθετα όμως αν έχει

βεβαρυμμένο ποινικό μητρώο για περισσότερα από δέκα (10) χρόνια αυτό του στερεί το δικαίωμα. Στη συνέχεια θα πρέπει να περάσει από τη διαδικασία συμπλήρωσης του κατάλληλου ερωτηματολογίου, που αφορά το συγκεκριμένο πρόγραμμα ασφάλισης υγείας που επιθυμεί να αποκτήσει. Ακολούθως, στέλλεται στα γραφεία το αίτημά του, για να μπορέσει να γίνει έλεγχος των δεδομένων του από το αρμόδιο προσωπικό. Αν ελεγχθούν όλα τα στοιχεία του με επιτυχία, τότε μπορεί με μια αρχική προκαταβολή, η οποία θα ξεπερνά τα εκατόν (100) ευρώ, η Ιατρική Ασφάλιση που διάλεξε να ξεκινήσει να είναι σε ισχύ. Αν όμως, η δεύτερη προκαταβολή δεν ξεπερνά τα εκατόν (100) ευρώ, τότε το συμβόλαιο παύει να είναι σε ισχύ.

Κεφάλαιο 4

Αναπαράσταση Πολιτικών Ασφάλειας στο Γοργία

4.1 Σημασία Μετατροπής Κανόνων στο Γοργία	34
4.2 Διαγράμματα Ροής	35
4.3 Μετατροπή σε Κανόνες στο Σύστημα του Γοργία	43

Εισαγωγή

Σ' αυτό το κεφάλαιο γίνεται μια άμεση σύνδεση με το προηγούμενο. Θα γίνει λεπτομερής αναπαράσταση των πιο πάνω Πολιτικών Ασφάλειας, όπως έχουν περιγραφεί, αρχικά ως διαγράμματα ροής δεδομένων και στη συνέχεια στο σύστημα του Γοργία καθώς και επεξήγησή τους μέσω διαφορετικών περιπτώσεων.

4.1 Σημασία Μετατροπής Κανόνων στο Γοργία

Για την μετατροπή και την αυτοματοποίηση των κανόνων από φυσική γλώσσα στην οποία είναι γραμμένοι, σε Κανόνες Πολιτικής, θα μας βοηθήσει το σύστημα του Γοργία για το οποίο έχουμε μιλήσει σε προηγούμενο κεφάλαιο (Κεφ.2).

Μια Ασφαλιστική Εταιρεία βρίσκεται αντιμέτωπη πολλές φορές με αποφάσεις για πολύ λεπτά ζητήματα που αφορούν τόσο τον πελάτη όσο και την εικόνα της Εταιρείας. Θεωρούμε το σύστημα του Γοργία ως το καταλληλότερο για την αυτοματοποιημένη λήψη αποφάσεων μέσω προτιμήσεων και συγκρούσεων σε διάφορους κανόνες. Σε πολλές περιπτώσεις υπάρχουν κανόνες οι οποίοι υπερισχύουν έναντι κάποιων άλλων κανόνων κάτω από συγκεκριμένες προϋποθέσεις. Στις περιπτώσεις αυτές το σύστημα πρέπει να είναι σε θέση να ανταπεξέλθει και να αποφασίζει με το σωστότερο τρόπο.

4.1.1 Αναπαράσταση Ασφαλιστικών Κανόνων

Η ορθή αναπαράσταση των κανόνων αποτελεί τη βασική προϋπόθεση για την εξαγωγή του επιθυμητού αποτελέσματος. Όπως έχουμε δει, οι κανόνες στο Γοργία έχουν την εξής μορφή: *rule(Signature, Head, Body)*.

Στην δική μας περίπτωση για την οποία θα κάνουμε χρήση των κανόνων για να αναπαραστήσουμε τις Πολιτικές Ασφάλειας, στο Body θα έχουμε τη λίστα από τις συνθήκες που θέτει η Ασφαλιστική Εταιρεία ως βασικές προϋποθέσεις για να ισχύει ο κανόνας. Στο Head θα βρίσκεται το αποτέλεσμα του κανόνα, αν και μόνο αν ισχύουν όλες οι προϋποθέσεις στο Body, και τέλος στο Signature θα είναι το επιθυμητό προσδιοριστικό του κανόνα σε συνδυασμό επιλεγμένων μεταβλητών από το Head και Body.

4.2 Διαγράμματα Ροής (FlowCharts)

Για την καλύτερη και πιο αναλυτική επεξήγηση του τρόπου λειτουργίας των Πολιτικών Ασφάλειας μέσα από τους φυσικούς Ασφαλιστικούς κανόνες, δημιουργήσαμε διαγράμματα ροής δεδομένων που αναπαριστούν τη διαδικασία που χρειάζεται για την λήψη αποφάσεων καθώς και ποια στοιχεία είναι απαραίτητα για κάθε απόφαση.[4]

4.2.1 Επεξήγηση Διαγραμμάτων Ροής

Στο διάγραμμα ροής 4.1 (Όροι Ασφάλισης Αυτοκινήτου Κεφ. 3.2) φαίνονται οι περιπτώσεις στις οποίες θα επιτρέπεται σε ένα άτομο να αποκτήσει Ασφάλιση Αυτοκινήτου και κάτω από ποιες προϋποθέσεις.

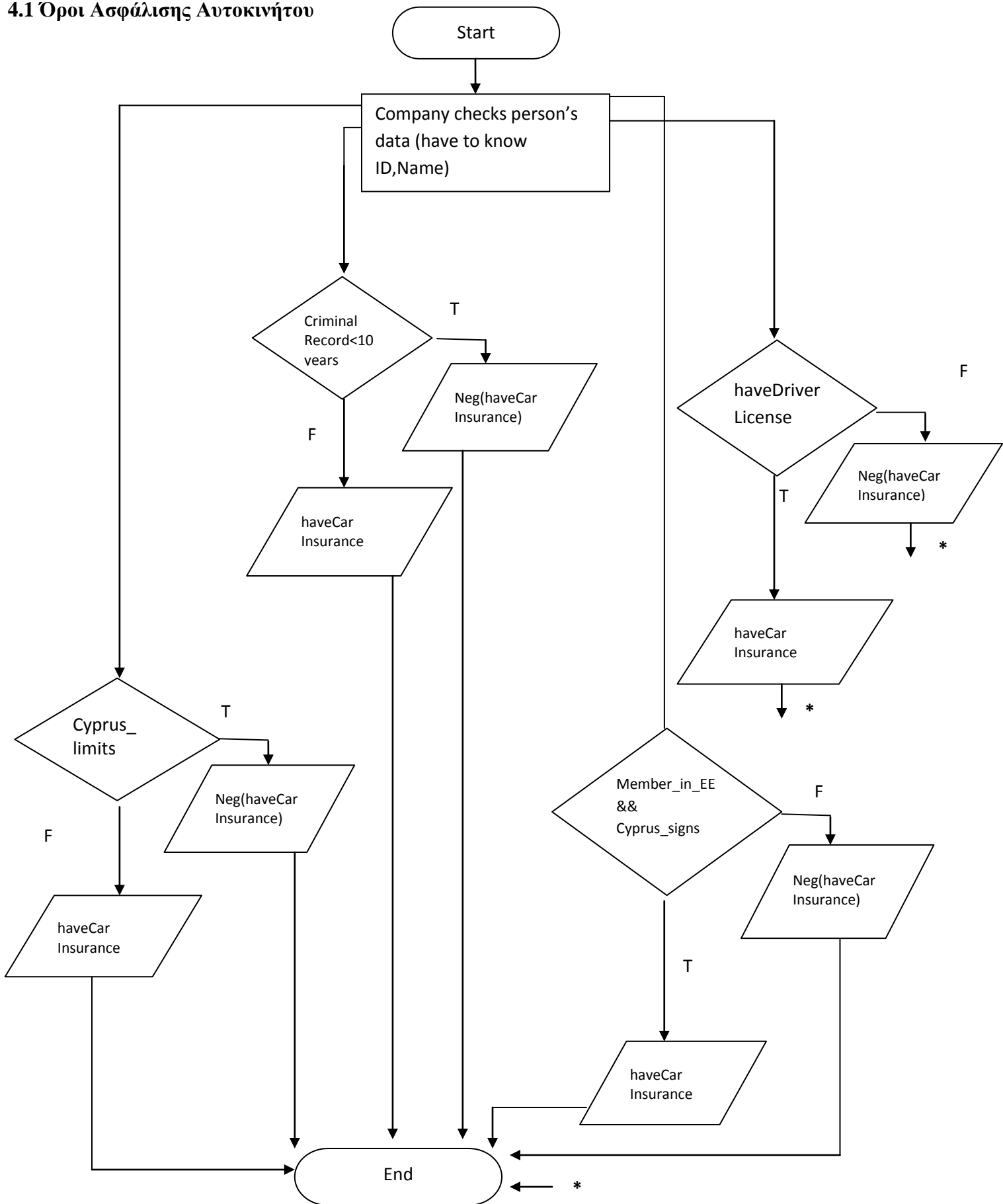
Στο διάγραμμα ροής 4.2 (Όροι Ακύρωσης Ασφαλιστικής Σύμβασης Κεφ. 3.3) γίνεται περιγραφή της διαδικασίας για την ακύρωση μιας Ασφαλιστικής Σύμβασης κάτω από

συγκεκριμένους όρους. Φαίνεται ξεκάθαρα ότι διαχωρίζεται σε δύο περιπτώσεις, ακύρωση από τον Ασφαλισμένο, ή ακύρωση από την Εταιρεία.

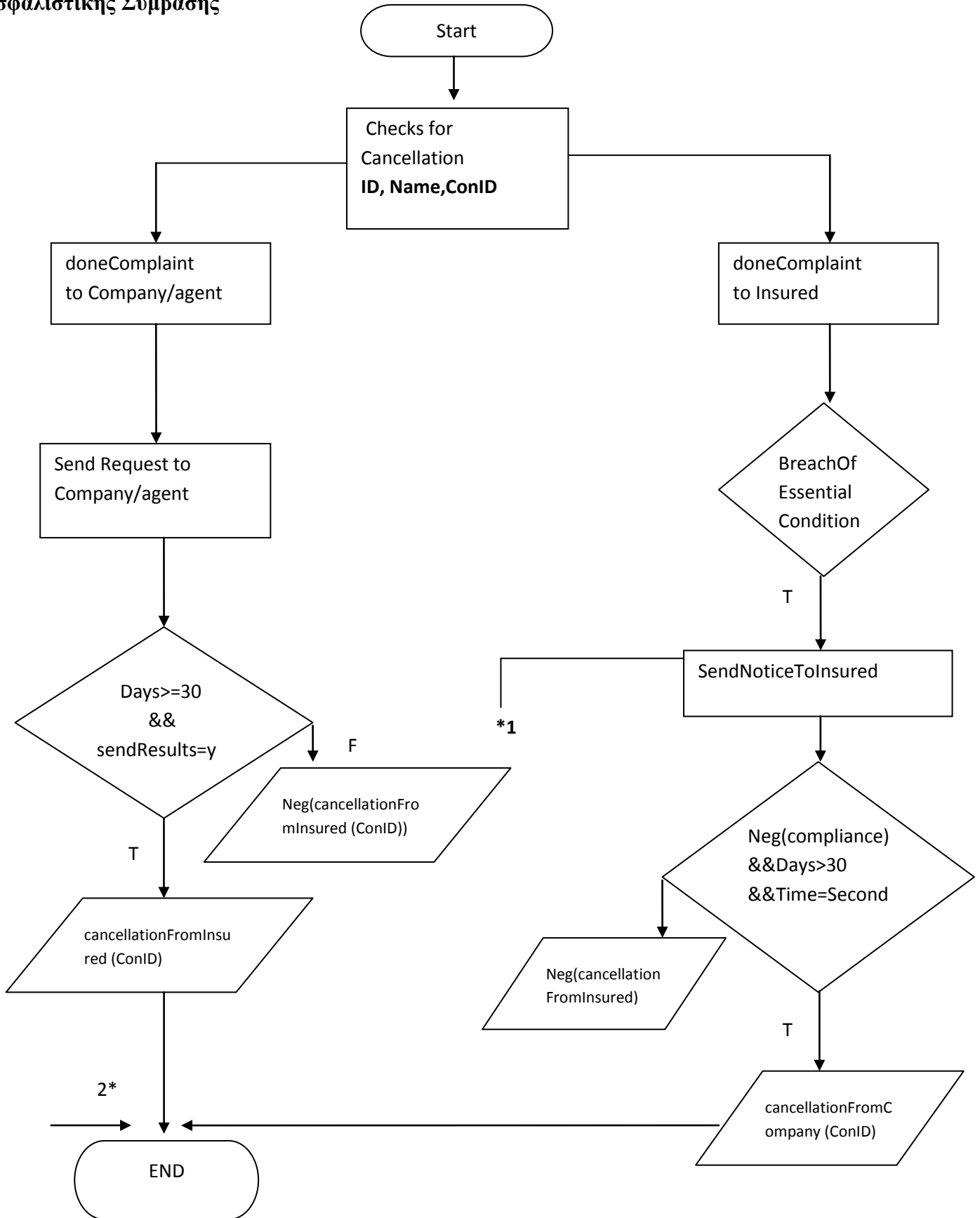
Στο διάγραμμα ροής 4.3 (Αποζημίωση σε Περίπτωση Απαίτησης Κεφ. 3.4) περιγράφεται η διαδικασία που απαιτείται για να δικαιούται ο Ασφαλισμένος αποζημίωση σε περίπτωση απαίτησης. Το θέμα της αποζημίωσης επειδή είναι αρκετά σημαντικό και λεπτό, χρειάζεται πολλούς ελέγχους από την Εταιρεία, όπου φαίνονται στο Διάγραμμα Ροής από τις συνθήκες.

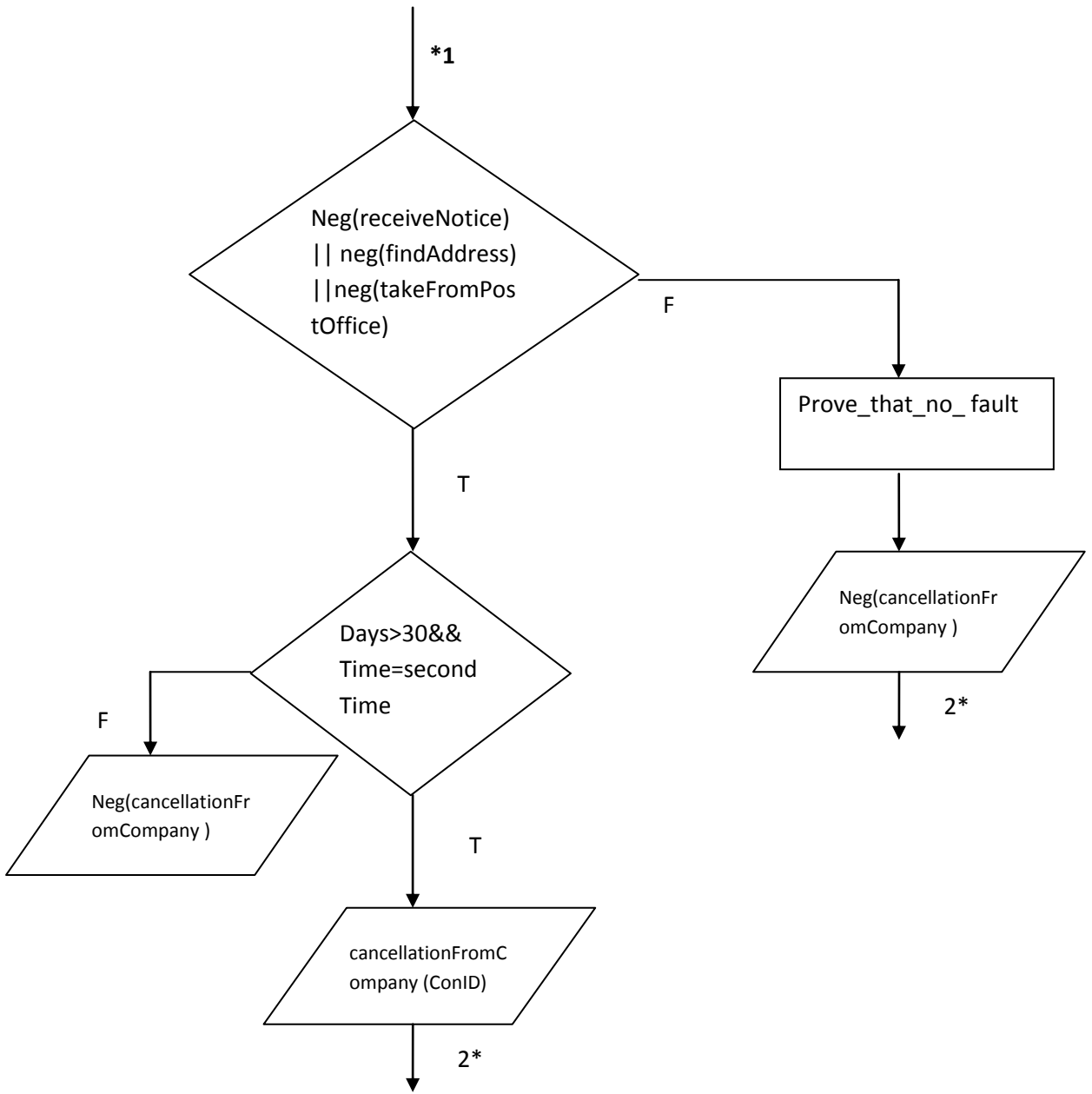
Στο διάγραμμα ροής 4.4 (Όροι Ασφάλισης Υγείας Κεφ. 3.5) φαίνεται η διαδικασία η οποία γίνεται για να μπορέσει να αποκτήσει ένα άτομο Ασφάλιση Υγείας, καθώς και οι προϋποθέσεις που απαιτούνται για την έναρξη της Ασφάλισης.

4.1 Όροι Ασφάλισης Αυτοκινήτου

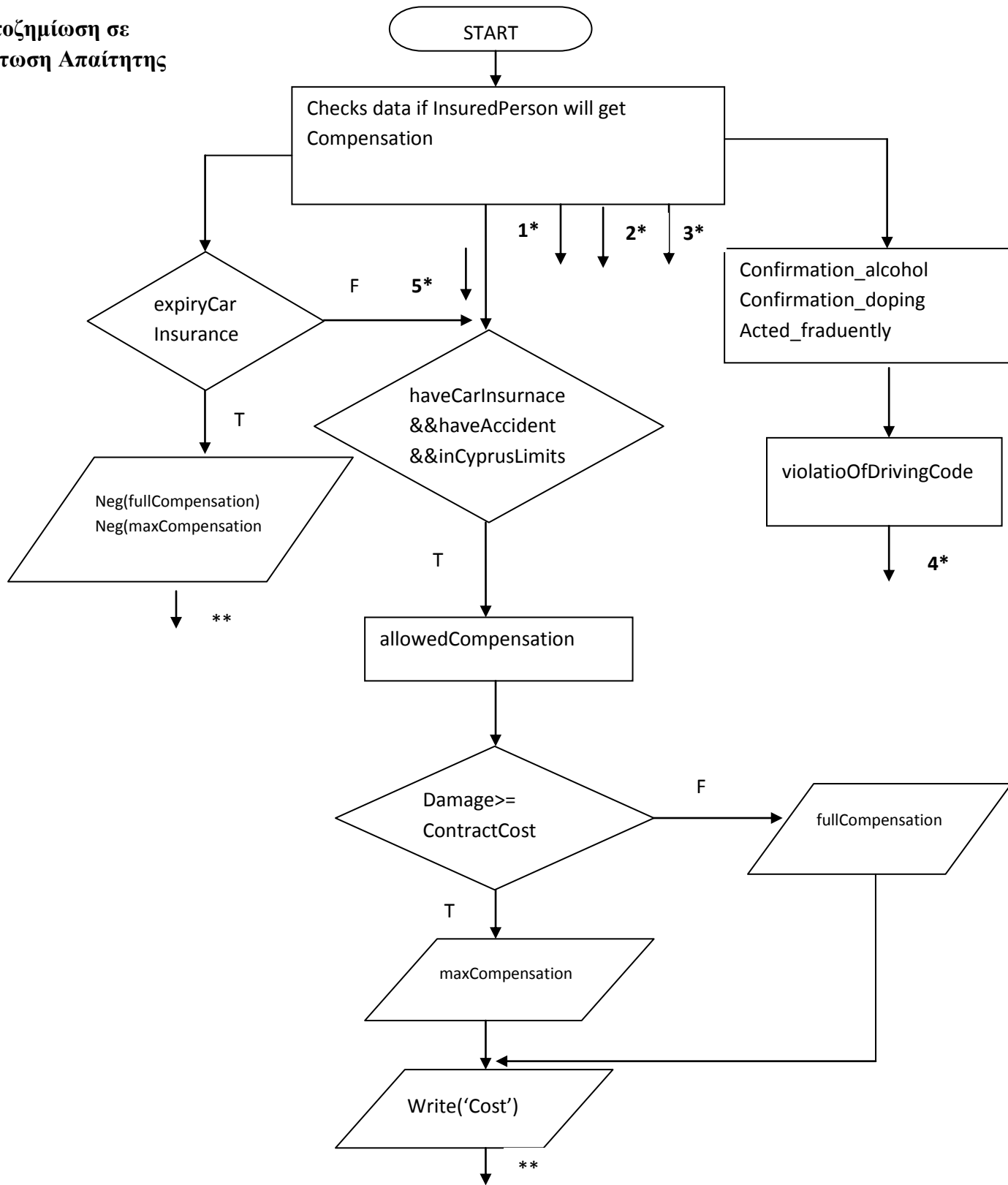


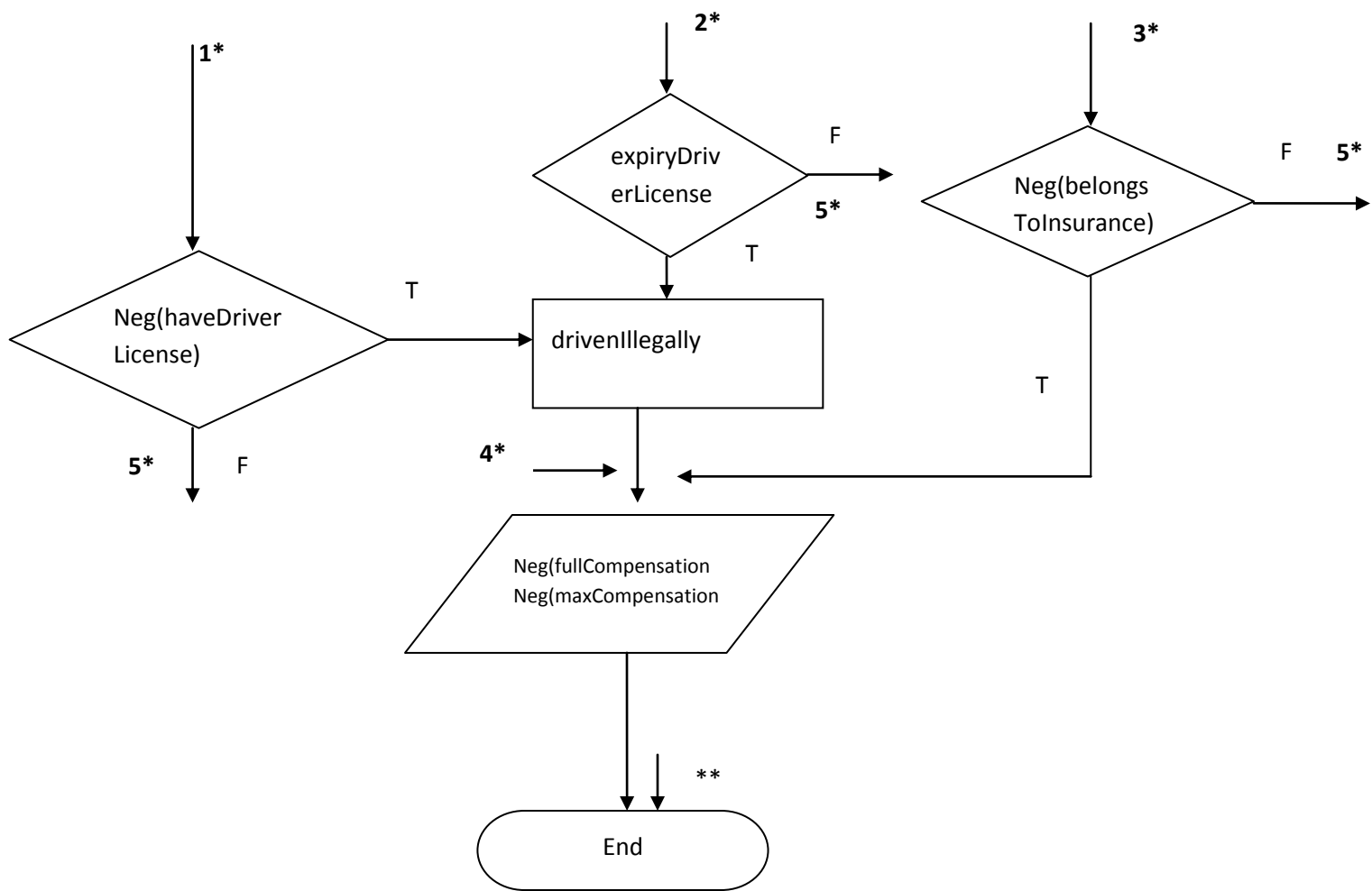
4.2 Όροι Ακύρωσης
Ασφαλιστικής Σύμβασης



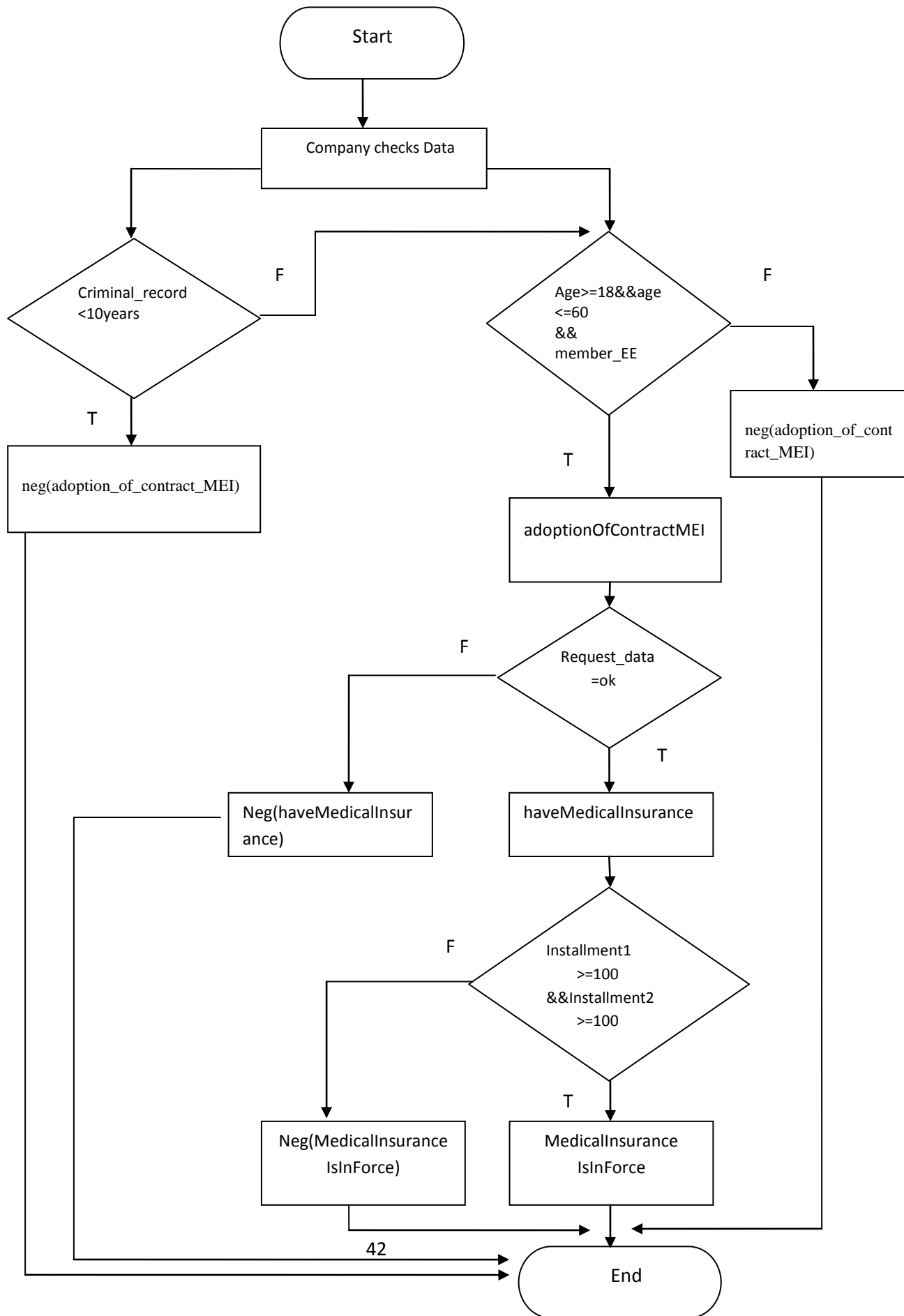


4.3 Αποζημίωση σε
Περίπτωση Απαιτήτης





4.4 Όροι Ασφάλισης Υγείας



4.3 Μετατροπή σε Κανόνες στο Σύστημα του Γοργία

4.3.1 Όροι Ασφάλισης Αυτοκινήτου σε Κανόνες στο Γοργία

Στο διάγραμμα ροής 4.1 όπως αναφέραμε πιο πάνω, περιγράφονται οι περιπτώσεις στις οποίες μπορεί να αποκτήσει ένα άτομο Ασφάλιση Αυτοκινήτου ή όχι (`haveCarInsurance(Person)`). Πιο κάτω μετατρέπονται οι συγκεκριμένοι όροι σε Κανόνες με την βοήθεια του συστήματος του Γοργία.

Ο κανόνας αυτός αποτελεί τη βασική περίπτωση στην οποία επιτρέπεται σ ένα άτομο να αποκτήσει Ασφάλιση Αυτοκινήτου.

```
rule(r1(Person),haveCarInsurance(Person),  
[member_in_EE(Person),cyprus_signs(Person)]).
```

Οι πιο κάτω τρεις κανόνες χρησιμοποιούν το αρνητικό όρο `neg()`, όπου υποδηλώνουν τις περιπτώσεις στις οποίες δεν επιτρέπεται η απόκτηση της Ασφάλισης Αυτοκινήτου, κάτω από τις συγκεκριμένες τρεις περιπτώσεις.

```
rule(r2(Person), neg(haveCarInsurance(Person)), [neg(in_cyprus_limits(Person))]).
```

```
rule(r3(Person), neg(haveCarInsurance(Person)), [criminal_record(Person)]).
```

```
rule(r4(Person), neg(haveCarInsurance(Person)), [neg(have_driver_license(Person))]).
```

Εκτός από τους κανόνες και τα κατηγορήματα που χρησιμοποιούνται, υπάρχει ανάγκη για την χρήση κανόνων προτεραιότητας, όπως μας δίνει τη δυνατότητα το σύστημα του Γοργία.

Ο κανόνας `r3` έχει μεγαλύτερη προτεραιότητα από τον κανόνα `r1`, δηλαδή σε περίπτωση που ένα άτομο έχει μαύρο ποινικό μητρώο, αντικρούει τον κανόνα που του επιτρέπει να αποκτήσει Ασφάλιση Αυτοκινήτου, ακόμα κι αν είναι μέλος της Ευρωπαϊκής Ένωσης και έχει κυπριακές πινακίδες αυτοκινήτου. Όμως, αυτό μπορεί να συμβεί μόνο σε περίπτωση που ο υποψήφιος Ασφαλισμένος έχει αποκτήσει μαύρο ποινικό μητρώο για λιγότερο από δέκα (10) χρόνια. Από την άλλη όμως έχουν περάσει περισσότερα από δέκα χρόνια, τότε του επιτρέπεται η απόκτηση Ασφάλισης Αυτοκινήτου.

rule(pr0(Person), prefer(r3(Person),r1(Person)), []).

rule(pr1(Person), prefer(r3(Person),r1(Person)), [less_than_10years(Person)]).

rule(pr2(Person), prefer(r1(Person),r3(Person)), [more_than_10years(Person)]).

Στους πιο κάτω κανόνες έχουμε τον κανόνα r2 να έχει πιο υψηλή προτεραιότητα από τον κανόνα r1, χωρίς όμως αυτό να είναι και απόλυτο. Σε περίπτωση που ένα άτομο δεν βρίσκεται εντός της Κυπριακής Επικράτειας, αλλά έχει στα χέρια του την πράσινη κάρτα που έχει εκδοθεί από την Εταιρεία, τότε αυτό κάνει τον κανόνα r2 να παύει να έχει μεγαλύτερη προτεραιότητα από τον κανόνα r1, και να αντιστρέφονται, με τον κανόνα r1 να προτιμάται από τον κανόνα r2, κάτω από αυτή την προϋπόθεση. Επιπρόσθετα, βλέπουμε την σημασία ύπαρξης άδειας οδήγησης για την απόκτηση Ασφάλισης Αυτοκινήτου, όπως φαίνεται από τον κανόνα προτίμησης pr6.

rule(pr3(Person), prefer(r2(Person),r1(Person)), []).

rule(pr4(Person), prefer(r2(Person),r1(Person)), [neg(have_green_card(Person))]).

rule(pr5(Person), prefer(r1(Person),r2(Person)), [have_green_card(Person)]).

rule(pr6(Person), prefer(r4(Person),r1(Person)), []).

4.3.2 Όροι Ακύρωσης Ασφαλιστικής Σύμβασης σε Κανόνες στο Γοργία

Στο διάγραμμα ροής 4.2 όπως έχουμε δει πιο πάνω, περιγράφονται οι περιπτώσεις στις οποίες επιτρέπεται στον Ασφαλισμένο ή στην Εταιρεία να ακυρώσει ή όχι μια Ασφαλιστική Σύμβαση.

(cancellationFromInsured(ConID), cancellationFromCompany(ConID)).

Πιο κάτω μετατρέπονται οι Πολιτικές της Εταιρείας σε Κανόνες με την βοήθεια του συστήματος του Γοργία.

Οι πρώτοι τρεις (3) κανόνες αφορούν την Ακύρωση μιας Ασφαλιστικής Σύμβασης για την οποία μπορεί να αιτηθεί ένας Ασφαλισμένος. Μπορεί να κάνει παράπονο είτε στην Εταιρεία είτε στον διαμεσολαβητή που τον έχει αναλάβει. Αφού σταλεί το

αίτημα/παράπονο και περάσουν τριάντα (30) ημέρες καθώς και τα αποτελέσματα είναι θετικά για την ακύρωση, τότε στέλλονται τα αποτελέσματα για την ακύρωση στον Ασφαλισμένο για την Ακύρωση της Ασφαλιστικής Σύμβασης.

```
rule(r1(Person,ConID), sendRequest(Person,ConID),  
[haveInsurance(Person,ConID),done_complaint_to_company(Person,ConID)]).
```

```
rule(r2(Person,ConID), sendRequest(Person,ConID),  
[haveInsurance(Person,ConID),done_complaint_to_agent(Person,ConID)]).
```

```
rule(r3(ConID,Person,Days,Answer), cancellationFromInsured(ConID),  
[sendRequest(Person,ConID),after30Days(Days),Days>30,sendResultsTo(Answer,Person,  
ConID),Answer=y]).
```

Οι επόμενοι κανόνες, αφορούν και περιγράφουν τη διαδικασία κατά την οποία μπορεί να γίνει ακύρωση Ασφαλιστικής Σύμβασης από την πλευρά της Εταιρείας. Παρουσιάζονται οι περιπτώσεις στις οποίες μπορεί να επέλθει η ακύρωση αλλά και οι περιπτώσεις οι οποίες δεν μπορεί να ισχύσει.

```
rule(r4(Person,ConID), sendNoticeTo(Person,ConID),  
[breach_of_essential_condition(Person),done_complaint_to_insured(Person,ConID)]).
```

```
rule(r5(Person,Days,ConID), cancellationFromCompany(ConID),  
[sendNoticeTo(Person,ConID),neg(compliance(Person)),after30Days(Days),Days>30]).
```

```
rule(r6(Person,ConID), cancellationFromCompany(ConID),  
[sendNoticeTo(Person,ConID),neg(receive_Notice(Person,ConID))]).
```

```
rule(r61(Person,ConID), cancellationFromCompany(ConID),  
[sendNoticeTo(Person,ConID),neg(find_address(Person))]).
```

```
rule(r62(Person,ConID), cancellationFromCompany(ConID),  
[sendNoticeTo(Person,ConID),neg(take_complaint_from_postOffice(Person))]).
```

```
rule(r7(Person,ConID), neg(cancellationFromCompany(ConID)),  
[prove_that_no_fault(Person,ConID),sendNoticeTo(Person,ConID),neg(receive_Notice(Person,ConID))]).
```

```
rule(r71(Person,ConID), neg(cancellationFromCompany(ConID)),  
[prove_that_no_fault(Person,ConID),sendNoticeTo(Person,ConID),neg(find_address(Person))]).
```

rule(r72(Person,ConID), neg(cancellationFromCompany(ConID)),
[prove_that_no_fault(Person,ConID),sendNoticeTo(Person,ConID),neg(take_complaint_from_postOffice(Person))]).

rule(r8(Person,ConID,Time),cancellationFromCompany(ConID),[sendNoticeTo(Person,ConID),neg(compliance(Person,Time)),Time=secondTime,after30Days(Days,Time),Days>30,Time=secondTime]).

Συνεχίζοντας, έχουμε τους πιο κάτω κανόνες να μας δείχνουν τις προτεραιότητες μεταξύ των κανόνων. Παρατηρούμε ότι οι κανόνες που προσδιορίζουν άρνηση στην Ακύρωση Ασφαλιστικής Σύμβασης (r7,r71,r72) έχουν υψηλότερη προτεραιότητα από τους αντίστοιχους κανόνες οι οποίοι προσδιορίζουν καταφατικά την ακύρωση (r5, r6, r61, r62).

rule(pr1(Person,ConID), prefer(r7(Person,ConID),r6(Person,ConID)), []).

rule(pr2(Person,ConID), prefer(r71(Person,ConID),r61(Person,ConID)), []).

rule(pr3(Person,ConID), prefer(r72(Person,ConID),r62(Person,ConID)), []).

rule(pr4(Person,Days,ConID), prefer(r7(Person,ConID),r5(Person,Days,ConID)), []).

rule(pr5(Person,Days,ConID), prefer(r71(Person,ConID),r5(Person,Days,ConID)), []).

rule(pr6(Person,Days,ConID), prefer(r72(Person,ConID),r5(Person,Days,ConID)), []).

rule(pr7(ConID,Person,Amount,Month1,Month2,M,CCost,ClaimId),
prefer(r31(ConID,Person,ClaimId,Amount),r32(ConID,Person,Amount,Month1,Month2,M,CCost)), []).

rule(pr8(ConID,Person,Amount,Month1,Month2,M,CCost,ClaimId),
prefer(r9(ConID,Person,ClaimId,Amount),r10(ConID,Person,Amount,Month1,Month2,M,CCost)), []).

rule(pr9(ConID,Person,Amount,ClaimId),
prefer(r11(ConID,Person,Amount),r9(ConID,Person,ClaimId,Amount)), []).

rule(pr10(ConID,Person,Amount,Month1,Month2,M,CCost),
prefer(r11(ConID,Person,Amount),r10(ConID,Person,Amount,Month1,Month2,M,CCost)), []).

4.3.2.1 Επιστροφή Ασφαλίσεων από Ακύρωση Ασφαλιστικής Σύμβασης σε Κανόνες στο Γοργία

Σε κάθε περίπτωση Ακύρωσης Ασφαλιστικής Σύμβασης, ακολουθείται η διαδικασία επιστροφής ασφαλίσεων στον δικαιούχο. Γίνεται έλεγχος αν ο Ασφαλισμένος έχει φάκελο απαίτησης, τότε όπως φαίνεται στον κανόνα r31 του επιστρέφονται μηδενικά ασφάλιστρα. Όμως, σε κάθε άλλη περίπτωση όπως βλέπουμε στον κανόνα r32, επιστρέφονται στον δικαιούχο τα μη δεδουλευμένα ασφάλιστρα χρησιμοποιώντας την πράξη υπολογισμού όπως φαίνεται πιο κάτω.

Τους ίδιους κανόνες χρησιμοποιούμε και σε περίπτωση ακύρωσης από την Εταιρεία. (οι κανόνες r9, r10). Βλ. Παράρτημα Β.

```
rule(r31(ConID,Person,Amount), neg(return_fees_to(Person,Amount)),  
[cancellationFromInsured(ConID), hasClaim(Person,ConID),write('Person has claim')]).
```

```
rule(r32(ConID,Person,Amount,Month1,Month2,M,CCost),return_fees_to(Person,Amount,  
[cancellationFromInsured(ConID),contractCost(CCost),monthStarted(Month1),monthCancelled(Month2),M is 12-(Month2-Month1),Amount is (CCost/12)*M, write(Amount),write('euros')]).
```

4.3.3 Αποζημίωση σε Περίπτωση Απαίτησης σε Κανόνες στο Γοργία

Στο διάγραμμα ροής 4.3 όπως φαίνεται πιο πάνω, περιγράφονται οι περιπτώσεις στις οποίες μπορεί ο Ασφαλισμένος να πάρει αποζημίωση σε περίπτωση απαίτησης καθώς και η μορφή της αποζημίωσης που δικαιούται ανάλογα με το ποσό της ζημιάς σε σύγκριση με το ποσό του συμβολαίου. Πιο κάτω θα δούμε πώς οι Πολιτικές της Εταιρείας μετατρέπονται σε Κανόνες με την βοήθεια του συστήματος του Γοργία.

Αρχικά, για να επιτραπεί σε ένα ασφαλισμένο να πάρει αποζημίωση υπάρχουν τρεις (3) βασικές προϋποθέσεις που πρέπει να τηρούνται. Αυτό φαίνεται από τον κανόνα r1 που αποτελεί τον βασικό κανόνα.

```
rule(r1(Person,AccId), allowedCompensation(Person,AccId),  
[haveCarInsurance(Person),haveAccident(Person,AccId),in_cyprus_limits(AccId)]).
```

Στους δύο πιο κάτω κανόνες(r2, r21), έχουμε τον διαχωρισμό των δύο ειδών αποζημίωσης, δηλαδή να φαίνεται σε ποια περίπτωση ο Ασφαλισμένος δικαιούται μέγιστη ή ολική αποζημίωση. Αυτό γίνεται με τις κατάλληλες συγκρίσεις.

```
rule(r2(Person,Damage,Cost,AccId), maxCompensation(Person,AccId),  
[allowedCompensation(Person,AccId),contract_cost(Cost),more_than_max_limit(Damage,  
Cost),Damage>=Cost,write(Cost),write(' euros'),nl,write('Max Compensation'),nl]).
```

```
rule(r21(Person,Damage,Cost,AccId), fullCompensation(Person,AccId),  
[allowedCompensation(Person,AccId),contract_cost(Cost),less_than_max_limit(Damage,C  
ost),Damage<Cost,write(Damage),write(' euros'), nl, write('Full Compensation'),nl]).
```

Όλοι οι πιο κάτω κανόνες, παρουσιάζουν τις περιπτώσεις όπου αποτρέπουν τον Ασφαλισμένο από το δικαίωμα είτε ολικής είτε μέγιστης αποζημίωσης.

```
rule(r3(Person,AccId), neg(maxCompensation(Person,AccId)),  
[expiryCarInsurance(Person)]).
```

```
rule(r31(Person,AccId), neg(fullCompensation(Person,AccId)),  
[expiryCarInsurance(Person)]).
```

```
rule(r4(Person), driven_illegally(Person), [neg(have_driver_license(Person))]).
```

```
rule(r41(Person), driven_illegally(Person), [expiryDriverLicense(Person)]).
```

```
rule(r42(Person), driven_illegally(Person), [neg(belongsInInsurance(Person))]).
```

```
rule(r43(Person,AccId), neg(maxCompensation(Person,AccId)),  
[driven_illegally(Person)]).
```

```
rule(r44(Person,AccId), neg(fullCompensation(Person,AccId)), [driven_illegally(Person)]).
```

```
rule(r5(Person,AccId), violationOfDrivingCode(Person ,AccId),  
[haveAccident(Person,AccId),confirmBythePolice_Alcohol(Person ,AccId)]).
```

rule(r51(Person,AccId), violationOfDrivingCode(Person ,AccId),
[haveAccident(Person,AccId),confirmBythePolice_Doping(Person ,AccId)]).

rule(r52(Person,AccId), neg(maxCompensation(Person,AccId)),
[violationOfDrivingCode(Person ,AccId)]).

rule(r53(Person,AccId), neg(fullCompensation(Person,AccId)),
[violationOfDrivingCode(Person ,AccId)]).

rule(r6(Person,AccId), neg(maxCompensation(Person,AccId)),
[confirmBythePolice_acted_fraudulently(Person,AccId)]).

rule(r61(Person,AccId), neg(fullCompensation(Person,AccId)),
[confirmBythePolice_acted_fraudulently(Person,AccId)]).

Λόγω του ότι έχουμε διαχωρίσει την αποζημίωση στις δύο (2) κατηγορίες που έχουμε προαναφέρει, έχουμε τη δυνατότητα μέσα από το σύστημα του Γοργία να δηλώσουμε το συμπλήρωμα δύο κατηγορημάτων. Αυτό υποδεικνύει ότι, το αποτέλεσμα της αποζημίωσης μπορεί να πάρει ολική μορφή ή μέγιστη, δεν μπορεί όμως ένας Ασφαλισμένος να δικαιούται να πάρει και από τις δύο κατηγορίες αποζημίωσης. Το αποτέλεσμα λοιπόν, θα πρέπει να είναι πάντοτε ένα από τα δύο.

complement(maxCompensation(Person,AccId),fullCompensation(Person,AccId)).

complement(fullCompensation(Person,AccId),maxCompensation(Person,AccId)).

Ακολούθως, πιο κάτω παρατηρούμε όλους τους κανόνες προτεραιότητας οι οποίοι είναι απαραίτητοι για την εξαγωγή των σωστών αποτελεσμάτων. Αρχικά έχουμε τον κανόνα με την μέγιστη αποζημίωση να έχει μεγαλύτερη προτεραιότητα από τον κανόνα με την ολική αποζημίωση, λόγω του ότι η μέγιστη αποζημίωση αποτελεί πιο περιοριστικό κανόνα όταν η ζημιά είναι μεγαλύτερη από το ποσό του συμβολαίου. Τότε ο δικαιούχος θα παραλάβει ως αποζημίωση το ποσό του συμβολαίου και όχι το ποσό της ζημιάς.

Σε κάθε περίπτωση βλέπουμε οποιοδήποτε κανόνα άρνησης παροχής αποζημίωσης να έχει υψηλότερη προτεραιότητα από τους βασικούς κανόνες r2 και r21. Οι κανόνες, δηλαδή, που αποτρέπουν τον Ασφαλισμένο να πάρει οποιαδήποτε είδους αποζημίωση, είτε από λήξη Ασφάλισης, είτε από παραβίαση Οδικού Κώδικα, είτε από παράνομη οδήγηση (όπως φαίνεται στους πιο πάνω κανόνες) αν ισχύουν παράλληλα με τους βασικούς κανόνες r1, r2, r21, τότε έχουν μεγαλύτερη προτεραιότητα, έχοντας ως αποτέλεσμα την μη ισχύ των βασικών κανόνων.

rule(pr1(Person,Damage,Cost,AccId),prefer(r2(Person,Damage,Cost,AccId),r21(Person,Damage,Cost,AccId)), []).

rule(pr2(Person,AccId), prefer(r31(Person,AccId),r1(Person,AccId)), []).

rule(pr3(Person,Damage,Cost,AccId),
prefer(r3(Person,AccId),r2(Person,Damage,Cost,AccId)), []).

rule(pr4(Person,Damage,Cost,AccId),
prefer(r31(Person,AccId),r21(Person,Damage,Cost,AccId)), []).

rule(pr5(Person,Damage,Cost,AccId),
prefer(r43(Person,AccId),r2(Person,Damage,Cost,AccId)), []).

rule(pr6(Person,AccId), prefer(r44(Person,AccId),r1(Person,AccId)), []).

rule(pr7(Person,Damage,Cost,AccId),
prefer(r44(Person,AccId),r21(Person,Damage,Cost,AccId)), []).

rule(pr8(Person,Damage,Cost,AccId),
prefer(r52(Person,AccId),r2(Person,Damage,Cost,AccId)), []).

rule(pr9(Person,Damage,Cost,AccId),
prefer(r53(Person,AccId),r21(Person,Damage,Cost,AccId)), []).

rule(pr10(Person,AccId), prefer(r53(Person,AccId),r1(Person,AccId)), []).

```
rule(pr11(Person,Damage,Cost,AccId),
prefer(r6(Person,AccId),r2(Person,Damage,Cost,AccId)), []).
```

```
rule(pr12(Person,Damage,Cost,AccId),
prefer(r61(Person,AccId),r21(Person,Damage,Cost,AccId)), []).
```

```
rule(pr13(Person,AccId), prefer(r61(Person,AccId),r1(Person,AccId)), []).
```

4.3.4 Όροι Ασφάλισης Υγείας σε Κανόνες στο Γοργία

Στο διάγραμμα ροής 4.4 όπως είδαμε πιο πάνω, περιγράφεται η διαδικασία κατά την οποία μπορεί ένα άτομο να αποκτήσει Ασφάλιση Υγείας. Τώρα θα δούμε πώς αυτές οι Πολιτικές της Εταιρείας μετατρέπονται σε Κανόνες με την βοήθεια του συστήματος του Γοργία. Οι δύο πιο κάτω κανόνες προσδιορίζουν την διαδικασία και τις προϋποθέσεις που απαιτούνται.

```
rule(r1(Person,ID), adoption_of_contract_MEI(Person,ID),
[member_in_EE(Person),person_age(ID,Age),Age>=18,Age=<60,write('Person has the
right age and he is member in EE'),nl]).
```

```
rule(r2(Person,ID), neg(adoption_of_contract_MEI(Person,ID)),
[criminal_record(ID,Person)]).
```

```
rule(r3(Person,Answer,ID,HConId), haveMedicalInsurance(Person,ID,HConId),
[adoption_of_contract_MEI(Person,ID),choose_MedicalExpensesInsurance(Person,ID),co
mplete_questionnaire(Person,ID),send_proposal_request_to_office(Person,ID),check_data(P
erson,ID,Answer),Answer=y,write('Person`s data are valid'),nl]).
```

```
rule(r4(Person,Amount,HConId,ID,Time), medicalInsuranceIsInForce(Person,ID,HConId),
[haveMedicalInsurance(Person,ID,HConId),installment(Time,Amount,HConId),Amount>=
100,Time=first]).
```

```
rule(r5(Person,Amount,HConId,ID,Time),neg(medicalInsuranceIsInForce(Person,ID,HCon
Id)),[haveMedicalInsurance(Person,ID,HConId),installment(Time,Amount,HConId),Amou
nt<100,Time=first,write('The first installment is less than 100 euros'),nl]).
```

rule(r6(Person,Amount2,HConId,ID,Time2),neg(medicalInsuranceIsInForce(Person,ID,HConId)), [installment2(Time2,Amount2,HConId),Amount2<100,Time2=second]).

Καταρχήν έχουμε ξανά τους κανόνες προτεραιότητα με το ποινικό μητρώο όπως είδαμε και στην Ασφάλιση Υγείας. Μπορούμε εύκολα επίσης να διακρίνουμε ότι οι κανόνες r5 και r6 έχουν σαφώς υψηλότερη προτεραιότητα από τον κανόνα r4. Αν η εξόφληση της πρώτης δόσης γίνει με το σωστό τρόπο, αλλά ακολούθως η δεύτερη δόση δεν εξοφληθεί σωστά, τότε δεν μπορεί το συμβόλαιο να βρίσκεται ακόμη σε ισχύ.

rule(pr1(Person,Amount,Amount2,HConId,ID,Time,Time2),prefer(r6(Person,Amount2,HConId,ID,Time2),r4(Person,Amount,HConId,ID,Time)), []).

rule(pr2(Person,Amount,HConId,ID,Time),prefer(r5(Person,Amount,HConId,ID,Time),r4(Person,Amount,HConId,ID,Time)), []).

rule(pr3(Person,ID), prefer(r2(Person,ID),r1(Person,ID)),[less_than_10years(ID,Person)]).

rule(pr4(Person,ID),prefer(r1(Person,ID),r2(Person,ID)),[more_than_10years(ID,Person)]).

Κεφάλαιο 5

Σύγκριση - Αξιολόγηση

5.1 Διαδικασία Αξιολόγησης	53
5.2 Σημασία Αξιολόγησης και Σύγκρισης	54
5.3 Αξιολόγηση Αποτελεσμάτων για Ασφάλιση Αυτοκινήτου	55
5.4 Αξιολόγηση Αποτελεσμάτων για Ακύρωση Ασφαλιστικής Σύμβασης	59
5.5 Αξιολόγηση Αποτελεσμάτων για Αποζημίωση σε Περίπτωση Απαίτησης	69
5.6 Αξιολόγηση Αποτελεσμάτων για Ασφάλιση Υγείας	83

Εισαγωγή

Στο κεφάλαιο αυτό θα ασχοληθούμε, θα παρουσιάσουμε και θα συγκρίνουμε τα αποτελέσματα που βγάζει το δικό μας πρόγραμμα σε σχέση με τα επιθυμητά αποτελέσματα από πραγματικά γεγονότα της Εταιρείας.

5.1 Διαδικασία Αξιολόγησης

Φτάνοντας σε αυτό το κεφάλαιο, αξίζει να σημειώσουμε ότι, σημαντικό μέρος της δουλειάς μας, έχει γίνει με την Ασφαλιστική Εταιρεία, όπου μεγάλο μέρος της έρευνας των πιο πάνω Πολιτικών Ασφάλειας έχει παρθεί από εκεί. Τα πιο κάτω σενάρια τα οποία θα μας βοηθήσουν στην αξιολόγηση και σύγκριση αποτελεσμάτων, τα έχουμε πάρει από αρμόδια άτομα που δουλεύουν στην Ασφαλιστική Εταιρεία, καθώς έχει γίνει μια πολύ όμορφη συνεργασία μεταξύ μας. Έχουμε συγκεντρώσει στο σύνολο τριάντα (30) περιπτώσεις, όμως τρέξαμε και συγκρίναμε σ' αυτό το κεφάλαιο τις δεκαεπτά από αυτές, γιατί θεωρήσαμε ότι θα ήταν καλύτερο να είχαμε για το καθένα διαφορετικές καταστάσεις

και σενάρια, χωρίς να επαναλαμβανόμαστε στα ίδια δεδομένα και περιστατικά. Με αυτό τον τρόπο θα καταφέρναμε να πάρουμε για κάθε σενάριο ξεχωριστά αποτελέσματα.

Έχει γίνει μια διαδικασία ανάλυσης των πολιτικών που δημιουργήθηκαν στο σύστημα, έτσι ώστε τα στοιχεία που πήραμε από την Εταιρεία να μπορούν να αντιστοιχιστούν και να αρμόζουν στα κατηγορήματα που δημιουργήσαμε. Στη συνέχεια έχουμε πάρει τα δεδομένα για κάθε ξεχωριστή περίπτωση, σε μορφή τέτοια που να αρμόζει να προσαρμοστούν κατάλληλα στις πολιτικές που έχουμε δημιουργήσει, για να μπορούν να βγάλουν αποτελέσματα και να συγκριθούν με τα αντίστοιχα έγκυρα που μας έχει δώσει η Ασφαλιστική Εταιρεία.

5.2 Σημασία Αξιολόγησης και Σύγκρισης

Αξιοσημείωτο κομμάτι μετά από κάθε έρευνα και εκπλήρωση ενός προγράμματος το οποίο δημιουργήσαμε για να επιλύσει κάποια ζητήματα, είναι η αξιολόγηση και η σύγκρισή του με πραγματικά γεγονότα. Έτσι λοιπόν, στόχος όλης αυτής της εργασίας είναι το δικό μας πρόγραμμα που έχουμε δημιουργήσει, να συμφωνεί με τα αποτελέσματα που μας έχουν δοθεί από την Εταιρεία.

Πρέπει να σημειώσουμε ότι, όλα τα δεδομένα τα οποία θα χρησιμοποιήσουμε για την σύγκριση και την αξιολόγηση σ' αυτό το κεφάλαιο, θα παραμείνουν ανώνυμα για λόγους εμπιστευτικότητας και ιδιωτικότητας προσωπικών δεδομένων των πελατών της Εταιρείας. Τα ονόματα που θα χρησιμοποιήσουμε δεν είναι τα πραγματικά ονόματα και δεν σχετίζονται με υπαρκτά πρόσωπα. Για σκοπούς ορθότητας του προγράμματος θα χρησιμοποιούμε το όνομα John, ταυτότητες και αναγνωριστικά στοιχεία τα οποία δεν είναι έγκυρα. Όμως τα συμβάντα και οι περιπτώσεις (profiles/cases) που θα χρησιμοποιήσουμε και θα εξετάσουμε είναι πραγματικά, ως εκ τούτου θα επιφέρουν πραγματικά αποτελέσματα. Αυτά τα αποτελέσματα καλούμαστε να συγκρίνουμε για να αξιολογήσουμε με απόλυτη ακρίβεια και συνέπεια.

5.3 Αξιολόγηση Αποτελεσμάτων για Ασφάλιση Αυτοκινήτου

Case 1: Στην περίπτωση αυτή, το συγκεκριμένο άτομο είναι μέλος της Ευρωπαϊκής Ένωσης, έχει στην κατοχή του αυτοκίνητο με κυπριακές πινακίδες, βρίσκεται εντός της Κυπριακής Επικράτειας και έχει άδεια οδηγού. Όμως έχει βεβαρυσμένο ποινικό μητρώο και συγκεκριμένα για 7 χρόνια. Με βάση αυτά τα δεδομένα που μας έχουν δοθεί από την Εταιρεία, η απάντηση για το αν αυτό το άτομο μπορεί να κάνει Ασφάλιση Αυτοκινήτου είναι **αρνητική**.

Πιο κάτω έχουμε ενσωματώσει τα πιο πάνω στοιχεία ως δεδομένα στο πρόγραμμα για το συγκεκριμένο profile για να συγκρίνουμε αν το δικό μας αποτέλεσμα συμφωνεί με αυτό της Εταιρείας.

```
rule(f1, member_in_EE(john), []).
rule(f2, cyprus_signs(john), []).
rule(f3, in_cyprus_limits(john), []).
rule(f4, criminal_record(john), []).
rule(f5, less_than_10years(john), []).
rule(f6, have_driver_license(john), []).
```

Στο ερώτημα που θέσαμε:

```
?- prove([haveCarInsurance(john)],Delta).
```

πήραμε την εξής απάντηση:

The person has criminal record under 10 years

```
Attack:[r3(john),f4,pr1(john),f5]
```

The person has criminal record under 10 years

```
Defence:[r1(john),f2,f1,ass(neg(prefer(r3(john),r1(john))))]
```

The person has criminal record under 10 years

```
Attack:[r3(john),f4,pr1(john),f5]
```

The person has criminal record under 10 years

Defence:[r1(john),f2,f1,ass(neg(prefer(r3(john),r1(john))))]

Attack:[pr1(john),f5]

The person has criminal record under 10 years

false.

Επομένως, παρατηρούμε ότι παίρνουμε την σωστή απάντηση. Λόγω του ότι ο john έχει βεβαρυσμένο ποινικό μητρώο λιγότερα από δέκα (10) χρόνια, τότε δεν του επιτρέπεται η απόκτηση Ασφάλισης Αυτοκινήτου, σύμφωνα με τον κανόνα r3 ο οποίος υπερισχύει του κανόνα r1 σε τέτοια περίπτωση. Αυτό φαίνεται από τον κανόνα προτεραιότητας pr1, βλ. παράρτημα Α .

Case 2: Συνεχίζοντας, στο παράδειγμα αυτό έχουμε το συγκεκριμένο άτομο να είναι μέλος της Ευρωπαϊκής Ένωσης, να έχει στην κατοχή του αυτοκίνητο με κυπριακές πινακίδες, να έχει άδεια οδηγού, καθώς όμως δεν βρίσκεται εντός της Κυπριακής Επικράτειας. Έχει στα χέρια του την «Πράσινη Κάρτα». Με βάση τα δεδομένα αυτά που μας έχουν δοθεί, η απάντηση για το αν αυτό το άτομο μπορεί να κάνει Ασφάλιση Αυτοκινήτου πρέπει να είναι **θετική**.

Πιο κάτω έχουμε προσαρμόσει τα πιο πάνω στοιχεία ως δεδομένα στο πρόγραμμα για το συγκεκριμένο profile για να συγκρίνουμε αν το δικό μας αποτέλεσμα συμφωνεί με αυτό της Εταιρείας.

rule(f1, member_in_EE(john), []).

rule(f2, cyprus_signs(john), []).

rule(f3, neg(in_cyprus_limits(john)), []).

rule(f4, have_green_card(john), []).

rule(f5, have_driver_license(john), []).

Στο ερώτημα που θέσαμε:

?- prove([neg(haveCarInsurance(john))],Delta).

πήραμε την εξής απάντηση:

The person can obtain CarInsurance because of green card

Attack:[r1(john),f2,f1,pr5(john),f4]

The person can obtain CarInsurance because of green card

Defence:[r2(john),f3,ass(neg(prefer(r1(john),r2(john)))))]

The person can obtain CarInsurance because of green card

Attack:[r1(john),f2,f1,pr5(john),f4]

The person can obtain CarInsurance because of green card

Defence:[r2(john),f3,ass(neg(prefer(r1(john),r2(john)))))]

Attack:[pr5(john),f4]

The person can obtain CarInsurance because of green card

false.

Παρατηρούμε ότι η απάντηση που παίρνουμε είναι αρνητική στην αρνητική ερώτηση που έχουμε θέσει (άρα θετική), άρα στο συγκεκριμένο παράδειγμα ο «john» μπορεί να αποκτήσει Ασφάλιση Αυτοκινήτου, λόγω του ότι έχει την πράσινη κάρτα που του το επιτρέπει αν και βρίσκεται εκτός των Κυπριακής Επικράτειας. Βλ. Παράρτημα Α.

Case 3: Στο ακόλουθο παράδειγμα, το συγκεκριμένο άτομο είναι μέλος της Ευρωπαϊκής Ένωσης, έχει αυτοκίνητο με κυπριακές πινακίδες και βρίσκεται εντός της Κυπριακής Επικράτειας, όμως δεν έχει άδεια οδήγησης.

Με βάση αυτά τα δεδομένα, η απάντηση για το αν αυτό το άτομο μπορεί να κάνει Ασφάλιση Αυτοκινήτου είναι πρέπει να είναι **αρνητική**.

Τα στοιχεία του φαίνονται ως εξής:

rule(f1, member_in_EE(john), []).

rule(f2, cyprus_signs(john), []).

rule(f3, in_cyprus_limits(john), []).

rule(f4, neg(have_driver_license(john)), []).

Στο ερώτημα που θέσαμε:

?- prove([haveCarInsurance(john)],Delta).

πήραμε την απάντηση:

The Person can not obtain CarInsurance because has not car license

Attack:[r4(john),f4,pr6(john)]

The Person can not obtain CarInsurance because has not car license

Defence:[r1(john),f2,f1,ass(neg(prefer(r4(john),r1(john))))]

The Person can not obtain CarInsurance because has not car license

Attack:[r4(john),f4,pr6(john)]

The Person can not obtain CarInsurance because has not car license

Defence:[r1(john),f2,f1,ass(neg(prefer(r4(john),r1(john))))]

Attack:[pr6(john)]

The Person can not obtain CarInsurance because has not car license

false.

Παρατηρούμε λοιπόν ότι η απάντηση είναι η αναμενόμενη λόγω του ότι ο «john» δεν έχει άδεια οδήγησης και σύμφωνα με τον κανόνα r4 δεν του επιτρέπεται η απόκτηση Ασφάλισης, όπως φαίνεται και στον κανόνα προτεραιότητας pr6. Βλ. Παράρτημα Α.

5.3.1 Πίνακας Αξιολόγησης Ασφάλισης Αυτοκινήτου

	Member inEE	Cyprus signs	in_cyprus limits	Criminal record	Have Driver license	Query: haveCar Insurance	Message	CompareResults
Case 1	✓	✓	✓	<10years	✓	false	The person has criminal record under 10 years	Same as Company's
Case 2	✓	✓	× havegreencard	×	✓	true	The person can obtain CarInsurance because of green card	Same as Company's
Case 3	✓	✓	✓	×	×	false.	The Person can not obtain CarInsurance because has not car license	Same as Company's

5.4 Αξιολόγηση Αποτελεσμάτων για Ακύρωση Ασφαλιστικής Σύμβασης και Επιστροφής Ασφαλίσεων

Για κάθε περίπτωση Ακύρωσης θα θέτουμε δύο ερωτήματα, το πρώτο θα αφορά αν μπορεί να γίνει ή όχι η ακύρωση και το δεύτερο το ποσό επιστροφής ασφαλίσεων.

Case 1: Σχετικά με την αυτή την ακύρωση, ο Ασφαλισμένος έχει θελήσει να κάνει παράπονο στην Εταιρεία και να στείλει το αίτημά του. Μετά από τριάντα (30) ημέρες και συγκεκριμένα σαράντα δύο (42), η Εταιρεία απαντά θετικά στο αίτημα του Ασφαλισμένου για ακύρωση. Το συμβόλαιό του έχει ξεκινήσει τον Φεβρουάριο και η ακύρωση έγινε τον Ιούνιο και ποσό του συγκεκριμένου συμβολαίου ανέρχεται στα €420. Επομένως πρέπει να γίνει έλεγχος και για επιστροφή ασφαλίσεων.

Με βάση τα πιο πάνω δεδομένα, η απάντηση που πρέπει να πάρουμε πρέπει να είναι θετική για την ακύρωση και να εμφανίζει και το ποσό των €280 ως επιστρεφόμενο.

Εδώ έχουμε ενσωματώσει τα πιο πάνω στοιχεία ως δεδομένα στο πρόγραμμα για τη συγκεκριμένη περίπτωση, για να συγκρίνουμε αν το δικό μας αποτέλεσμα συμφωνεί με αυτό της Εταιρείας.

Τα στοιχεία φαίνονται ως εξής:

rule(f1, haveInsurance(john,id1), []).

rule(f2, done_complaint_to_company(john,id1), []).

rule(f3, after30Days(42), []).

rule(f4, sendResultsTo(y,john,id1), []).

rule(f5, monthStarted(2), []).

rule(f6, monthCancelled(6), []).

rule(f7, contractCost(420), []).

Στο ερώτημα που θέσαμε:

1 ?- prove([cancellationFromInsured(id1)],Delta).

Data is ok, cancellation from insured is done

Delta = [f4, f3, f2, f1, r1(john, id1), r3(id1, john, 42, y)] ;

false.

2 ?- prove([return_fees_to(john,X)],Delta).

280 euros

X = 280,

Delta = [f6, f5, f7, f4, f3, f2, f1, r1(john, id1), r3(..., ..., ..., ...)|...];

false.

Εύκολα μπορούμε να παρατηρήσουμε ότι, τα δικά μας αποτελέσματα είναι τα αναμενόμενα και συμφωνούν με αυτά της Εταιρείας. Οι κανόνες όπως αναγράφονται στα πιο πάνω αποτελέσματα φαίνονται καλύτερα στο Παράρτημα Β.

Case 2: Στην παρούσα ακύρωση έχουμε τον Ασφαλισμένο αυτή τη φορά να κάνει παράπονο και να στέλλει το αίτημά του στον διαμεσολαβητή (agent) με τον οποίο

επικοινωνεί. Αφού έχουν περάσει 38 ημέρες, στέλλονται πίσω στον Ασφαλισμένο με θετικό αποτέλεσμα. Το συμβολαίο του έχει ημερομηνία έναρξης τον Απρίλιο και η ακύρωση έγινε τον Σεπτέμβριο. Το ποσό του συγκεκριμένου συμβολαίου ανέρχεται στα €375, όμως πρέπει να σημειωθεί ότι ο Ασφαλισμένος έχει φάκελο απαίτησης στην Εταιρεία.

Με βάση τα πιο πάνω δεδομένα, η απάντηση που πρέπει να πάρουμε πρέπει να είναι θετική για την ακύρωση και να εμφανίζει False λόγω του ότι έχει φάκελο απαίτησης που του αποτρέπει το δικαίωμα επιστροφής ασφαλίσεων.

Πιο κάτω ενσωματώσαμε τα στοιχεία ως δεδομένα στο πρόγραμμα για τη συγκεκριμένη περίπτωση, για να συγκρίνουμε αν το δικό μας αποτέλεσμα συμφωνεί με αυτό της Εταιρείας.

Τα στοιχεία φαίνονται ως εξής:

rule(f1, haveInsurance(john,id2), []).

rule(f2, done_complaint_to_agent(john,id2), []).

rule(f3, after30Days(38), []).

rule(f4, sendResultsTo(y,john,id2), []).

rule(f5, monthStarted(4), []).

rule(f6, monthCancelled(9), []).

rule(f7, contractCost(375), []).

rule(f8, hasClaim(john,id2), []).

Στο ερώτημα :

1 ?- prove([cancellationFromInsured(id2)],Delta).

Data is ok, cancellation from insured is done

Delta = [f4, f3, f2, f1, r2(john, id2), r3(id2, john, 38, y)] ;

false.

2?- prove([neg(return_fees_to(john,X))],Delta).

Person has claim

Delta = [f8, f4, f3, f2, f1, r2(john, id2), r3(id2, john, 38, y), r31(id2, john, X)] ;

false.

Τρέχοντας λοιπόν το πιο πάνω παράδειγμα έχουμε πάρει θετική απάντηση για την ακύρωση ασφαλιστικής σύμβασης, καθώς επίσης το Delta δείχνει για ποιο λόγο δεν θα επιστραφούν ασφαλίστρα, όπως και έπρεπε. Ο κανόνας ο οποίος προϋποθέτει ότι ο Ασφαλισμένος έχει Claim έχει υψηλότερη προτεραιότητα (r31) από τον κανόνα που επιστρέφει το ποσό ασφαλίστρων. Βλ παράρτημα Β.

5.4.1 Πίνακας Αξιολόγησης Ακύρωσης από Ασφαλισμένο

	have Insurance	done Complaint To	After 30 Days	Send Results	month Started	Month Cancelled	Contract Cost €	hasClaim	Query1: Cancellation FromInsured	Message1	Query2: returnFees	Message2	Compare Results
Case1 (id1)	v	v(Company)	42	y	2	6	420	x	true	Data is ok, cancellati on from insured is done	true	X = 280 euros	Same as Company's
Case2 (id2)	v	v(Agent)	38	y	4	9	375	v	true	Data is ok, cancellati on from insured is done	false	Person has claim	Same as Company's

Πιο κάτω θα εξετάσουμε τις περιπτώσεις Ακύρωσης που υποβάλλονται από την Εταιρεία.

Case 3: Αρχικά, η Εταιρεία στέλλει το δικό της αίτημα/παράπονο στον Ασφαλισμένο, μετά από παραβίαση κάποιου ουσιώδους κανόνα από τον ίδιο. Αφού έχουν περάσει τριάντα τέσσερις (34) ημέρες χωρίς ο Ασφαλισμένος να έχει συμμορφωθεί, τότε με βάση τους κανονισμούς η Εταιρεία μπορεί να εκδώσει την Ακύρωση της συγκεκριμένης Ασφαλιστικής Σύμβασης. Στη συνέχεια όμως έχει αποδείξει ότι δεν έχει φτάσει ποτέ στα χέρια του η αίτηση με αποτέλεσμα να μην μπορεί να ισχύσει η ακύρωση, και να του δίνεται το δικαίωμα για δεύτερη φορά μέτρησης των 30 ημερών, αφού τώρα πια έχει στα χέρια του το αίτημα. Και για τη δεύτερη διορία που του δόθηκε, δεν έχει συμμορφωθεί με τους κανόνες της Εταιρείας. Το συμβόλαιο έχει ξεκινήσει τον Μάιο και το ποσό του ανέρχεται στα €490. Η ακύρωση σημειώθηκε τον μήνα Οκτώβριο, με το πέρας των 42 ημερών.

Η απάντηση που πρέπει να πάρουμε πρέπει να είναι θετική για την ακύρωση και να εμφανίζει το ποσό των €285,83 επιστροφής ασφαλίσεων.

Πιο κάτω ενσωματώσαμε τα στοιχεία ως δεδομένα στο πρόγραμμά μας:

```
rule(f1, breach_of_essential_condition(john), []).
```

```
rule(f2, done_complaint_to_insured(john,cid3), []).
```

```
rule(f3, neg(compliance(john)), []).
```

```
rule(f4, after30Days(34), []).
```

```
rule(f5, neg(find_address(john)), []).
```

```
rule(f6, prove_that_no_fault(john,cid3), []).
```

```
rule(f7, after30Days(42,secondTime), []).
```

```
rule(f8, neg(compliance(john,secondTime)), []).
```

```
rule(f9, contractCost(490), []).
```

```
rule(f10, monthStarted(5), []).
```

```
rule(f11, monthCancelled(10), []).
```

Θέτουμε τα πιο κάτω ερωτήματα:

1 ?- prove([neg(cancellationFromCompany(cid3))],Delta).

No Cancellation, address not found

Delta = [f5, f2, f1, r4(john, cid3), f6, r71(john, cid3)] ;

false.

Χωρίς να προσθέσουμε τα δεδομένα f7,f8, πιο πάνω παίρνουμε την σωστή απάντηση, ότι πράγματι δεν μπορεί να γίνει η ακύρωση λόγω του κανόνα r71. Στη συνέχεια αφού προσθέσουμε τα συγκεκριμένα δεδομένα, παρατηρούμε ότι το σύστημά μας, αφού βγάζει όλα τα attacks και defences, βγάζει το πιο κάτω αποτέλεσμα. Το ίδιο πρόβλημα αντιμετωπίζουμε πιο κάτω και στον υπολογισμό επιστροφής ασφαλιστρών, βγάζοντας όμως το σωστό ποσό.

2?- prove([cancellationFromCompany(cid3)],Delta).

...

Delta = [r8(john, cid3, secondTime), f7, f8, ass(neg(prefer(r71(john, cid3), r8(john, cid3, secondTime))))), f5, f2, f1, r4(john, cid3), r61(..., ...)] ;

The cancellation is valid,the insured not compliance.

Delta = [f7, f8, f2, f1, r4(john, cid3), r8(john, cid3, secondTime)] ;

false.

3?- prove([return_fees_to(john,X)],Delta).

...

X = 285.83333333333337,

Delta = [ass(neg(prefer(r71(john, cid3), r8(john, cid3, secondTime))))), f11, f10, f9, f7, f8, f2, f1, r4(..., ...)|...] ;

false.

Στο συγκεκριμένο παράδειγμα παρατηρούμε την αδυναμία του προγράμματός μας να βγάζει την ακριβή και μοναδική απάντηση εξ' αρχής.

Case 4: Έχοντας αρχικά ίδια σχετικά δεδομένα με την προηγούμενη περίπτωση, με τη διαφορά ότι ο Ασφαλισμένος έχει αποδείξει ότι δεν έχει παραλάβει το αίτημα παραπόνου από την Εταιρεία άρα έχει τη δυνατότητα για δεύτερη ευκαιρία συμμόρφωσης εντός 30 ημερών, όπως και έχει πράξει. Το συμβόλαιό του έχει ξεκινήσει τον Απρίλιο με ποσό €340 και η ακύρωση έχει σταλεί τον Δεκέμβριο.

Σύμφωνα με τα πιο πάνω, η ακύρωση του συγκεκριμένου Ασφαλισμένου δεν πρέπει να είναι σε ισχύ τελικά, και το πρόγραμμα δεν μπορεί να βγάλει κανένα ποσό επιστροφής ασφαλιστρών.

Έχουμε τα πιο κάτω δεδομένα:

rule(f1, breach_of_essential_condition(john), []).

rule(f2, done_complaint_to_insured(john,cid4), []).

rule(f3, neg(compliance(john)), []).

rule(f4, after30Days(52), []).

rule(f5, neg(receive_Notice(john,cid4)), []).

rule(f6, prove_that_no_fault(john,cid4), []).

rule(f7, after30Days(41,secondTime), []).

rule(f8, compliance(john,secondTime), []).

rule(f9, contractCost(340), []).

rule(f10, monthStarted(4), []).

rule(f11, monthCancelled(12), []).

Για τα πιο κάτω ερωτήματα παίρνουμε τις εξής απαντήσεις:

1 ?- prove([neg(cancellationFromCompany(cid4))],Delta).

No Cancellation,the insured has not received the notice

Delta = [f5, f2, f1, r4(john, cid4), f6, r7(john, cid4)] ;

false.

2 ?- prove([neg(return_fees_to(john,X))],Delta).

No Cancellation,the insured has not received the notice

No Cancellation,No return fees

Delta = [f5, f2, f1, r4(john, cid4), f6, r7(john, cid4), r11(cid4, john, X)] ;

false.

Τρέχοντας τα πιο πάνω ερωτήματα παίρνουμε τις σωστές απαντήσεις, συγκεκριμένα όμως μόνο με την αρνητική μορφή. Αντίθετα όμως, αν τρέξουμε τα ίδια ερωτήματα, αποφεύγοντας την άρνηση, τα αποτελέσματά μας εμφανίζονται με μια μεγάλη λίστα με όλα τα attacks και defences που δημιουργούνται, χωρίς να καταλαβαίνουμε ξεκάθαρα την απάντηση.

Case 5: Ο Ασφαλισμένος στη συγκεκριμένη περίπτωση ενώ έχει γίνει το αίτημα από την Εταιρεία, απέδειξε ότι δεν το έχει παραλάβει. Δεν έχει παρατηρηθεί συμμόρφωσή του και έτσι μπορεί να εκδοθεί η ακύρωση τελειωτικά. Το συμβόλαιο του έχει ξεκινήσει τον Ιούλιο και έχει γίνει αίτημα για παράπονο τον Νοέμβριο. Το ποσό του συμβολαίου ανέρχεται στα €505. Πρέπει να σημειωθεί ότι η Εταιρεία έχει στο όνομα του, φάκελο απαίτησης που του απαγορεύει το δικαίωμα επιστροφής ασφαλιστρών.

Έτσι λοιπόν, σύμφωνα με τα πιο πάνω, η ακύρωση πρέπει να ισχύσει, αλλά δεν μπορεί να του επιστραφούν τα μη δεδουλευμένα ασφάλιστρα.

Έχουμε προσαρμόσει τα δεδομένα πιο κάτω:

rule(f1, breach_of_essential_condition(john), []).

rule(f2, done_complaint_to_insured(john,cid5), []).

rule(f3, neg(compliance(john)), []).

rule(f4, after30Days(45), []).

rule(f5, neg(receive_Notice(john,cid5)), []).

rule(f6, prove_that_no_fault(john,cid5), []).

rule(f7, after30Days(40,secondTime), []).

rule(f8, neg(compliance(john,secondTime)), []).

rule(f9, hasClaim(john,c5), []).

rule(f10, contractCost(505), []).

rule(f11, monthStarted(7), []).

rule(f12, monthCancelled(11), []).

Θέτουμε τα πιο κάτω ερωτήματα:

1 ?- prove([cancellationFromCompany(cid5)],Delta).

...

Delta = [r8(john, cid5, secondTime), f7, f8, ass(neg(prefer(r7(john, cid5), r8(john, cid5, secondTime))))), f5, f2, f1, r4(john, cid5), r6(..., ...)] ;

The cancellation is valid,the insured not compliance

Delta = [f7, f8, f2, f1, r4(john, cid5), r8(john, cid5, secondTime)] ;

false.

2 ?- prove([neg(return_fees_to(john,X))],Delta).

...

The cancellation is valid,the insured not compliance

Person has claim,no return fees

Delta = [f9, f7, f8, f2, f1, r4(john, cid5), r8(john, cid5, secondTime), r9(cid5, john, X)] ;

false.

Μας δίνεται η σωστή απάντηση, όμως μετά από την εμφάνιση όλων των attacks και defences που δημιουργούνται. Πιο πάνω παρουσιάζουμε τα τελευταία δύο αποτελέσματα για την απάντηση η οποία δεν είναι η μοναδική. Λόγω αυτού του προβλήματος μη εύρεσης της μοναδικής λύσης, το πρόγραμμά μας αδυνατεί να απαντήσει μοναδικά και στο δεύτερο ερώτημα, prove([return_fees_to(john,X)],Delta).

5.5 Αξιολόγηση Αποτελεσμάτων για Αποζημίωση σε Περίπτωση Απαίτησης

Case 1: Σε αυτό το παράδειγμα εξετάζεται η περίπτωση ενός Ασφαλισμένου, ο οποίος έχει κάνει δυστύχημα εντός της Κυπριακής Επικράτειας. Βάση αυτών των γεγονότων έχει το δικαίωμα για αποζημίωση. Σύμφωνα με το συμβόλαιό του έχει συμφωνηθεί το ποσό των €520 και η ζημιά έχει εκτιμηθεί στα €700.

Σχετικά με τα πιο πάνω, το σύστημα πρέπει να βγάλει ότι ο Ασφαλισμένος δικαιούται αρχικά να πάρει αποζημίωση (θετική απάντηση), η οποία θα είναι και η μέγιστη, λόγω του ότι η ζημιά είναι μεγαλύτερη από το ποσό συμβολαίου. Δηλαδή θα πρέπει να βγάλει αποτέλεσμα ίσο με το ποσό του συμβολαίου, €520.

Ενσωματώνουμε τα στοιχεία ως δεδομένα στο πρόγραμμα για τη συγκεκριμένη περίπτωση, για να συγκρίνουμε αν το δικό μας αποτέλεσμα συμφωνεί με το αναμενόμενο.

Τα στοιχεία φαίνονται ως εξής:

```
rule(f1, haveCarInsurance(john), []).
```

```
rule(f2, in_cyprus_limits(acc_id1), []).
```

```
rule(f3, haveAccident(john,acc_id1), []).
```

```
rule(f4, contract_cost(520), []).
```

```
rule(f5, more_than_max_limit(700,520), []).
```

Τρέχουμε τα πιο κάτω ερωτήματα και παίρνουμε τα εξής αποτελέσματα:

```
1 ?- prove([allowedCompensation(john,acc_id1)],Delta).
```

The insured allow to get compensation

Delta = [f2, f3, f1, r1(john, acc_id1)] ;

false.

2 ?- prove([maxCompensation(john,acc_id1)],Delta).

The insured allow to get compensation

520 euros

Max Compensation

Delta = [f5, f4, f2, f3, f1, r1(john, acc_id1), r2(john, 700, 520, acc_id1)] ;

false.

Βλέπουμε λοιπόν, ότι η απάντηση στην πρώτη ερώτηση με το αν επιτρέπεται στον Ασφαλισμένο «john» να πάρει αποζημίωση είναι θετική και φαίνεται από τους κανόνες και τα γεγονότα στο Delta. Στην δεύτερη ερώτηση εμφανίζεται σωστά το αποτέλεσμα για το ποσό της αποζημίωσης και ισούται με €520. Αυτό φαίνεται από τους κανόνες και τα συγκεκριμένα γεγονότα που δόθηκαν ως δεδομένα. Βλ παράρτημα Γ.

Case 2: Στο δεύτερο παράδειγμα της συγκεκριμένης πολιτικής, έχουμε τον Ασφαλισμένο να δικαιούται αποζημίωση αφού τηρεί τις τρεις απαραίτητες προϋποθέσεις. Έχει καταγεγραμμένο ποσό συμβολαίου €440 και εκτιμημένη ζημιά στα €355.

Επομένως, με βάση τα πιο πάνω, στο συγκεκριμένο παράδειγμα έχουμε το Ασφαλισμένο να δικαιούται την ολική αποζημίωση, €355 , αφού η ζημιά είναι μικρότερη από το ποσό του συμβολαίου.

Τα δεδομένα έχουν ως εξής:

rule(f1, haveCarInsurance(john), []).

rule(f2, in_cyprus_limits(acc_id2), []).

rule(f3, haveAccident(john,acc_id2), []).

rule(f4, contract_cost(440), []).

rule(f5, less_than_max_limit(355,440), []).

Τρέχουμε τα ερωτήματα και παίρνουμε τα πιο κάτω αποτελέσματα:

1 ?- prove([allowedCompensation(john,acc_id2)],Delta).

The insured allow to get compensation

Delta = [f2, f3, f1, r1(john, acc_id2)] ;

false.

2 ?- prove([fullCompensation(john,acc_id2)],Delta).

The insured allow to get compensation

355 euros

Full Compensation

Delta = [f5, f4, f2, f3, f1, r1(john, acc_id2), r21(john, 355, 440, acc_id2)] ;

false.

Εύκολα διαπιστώνουμε ότι έχουμε τα σωστά αποτελέσματα σχετικά με τα πιο πάνω ερωτήματα. Πράγματι, επιτρέπεται στον john να πάρει αποζημίωση, όπως φαίνεται στο πρώτο ερώτημα και εμφανίζεται το σωστό ποσό για την αποζημίωση σύμφωνα με το δεύτερο ερώτημα. Οι απαντήσεις δικαιολογούνται βλέποντας τα δεδομένα που δώσαμε και τους κατάλληλους κανόνες όπως φαίνεται στο παράρτημα Γ.

Case 3: Εδώ έχουμε τον Ασφαλισμένο αφού έχει εμπλακεί σε ατύχημα στα Κυπριακά όρια, να μπορεί να πάρει αποζημίωση. Έχει στο συμβόλαιό του το συμφωνημένο ποσό των €495 και η ζημιά έχει εκτιμηθεί στα €460. Εκτός αυτών όμως, παρατηρείται ότι έχει λήξει η Ασφάλιση Αυτοκινήτου του.

Με βάση τα πιο πάνω, το σύστημα πρέπει να βγάλει ότι ο Ασφαλισμένος δικαιούται αρχικά να πάρει αποζημίωση (θετική απάντηση), αλλά στην συνέχεια δεν θα του επιτραπεί να πάρει την ολική αποζημίωση λόγω του ότι έχει λήξει το συμβόλαιο Ασφάλισης.

Τα δεδομένα μας έχουν ως εξής:

rule(f1, haveCarInsurance(john), []).
rule(f2, in_cyprus_limits(acc_id3), []).
rule(f3, haveAccident(john,acc_id3), []).
rule(f4, contract_cost(495), []).
rule(f5, less_than_max_limit(460,495), []).
rule(f6, expiryCarInsurance(john), []).

Αφού τρέξουμε τα πιο κάτω ερωτήματα, έχουμε τις εξής απαντήσεις:

1 ?- prove([allowedCompensation(john,acc_id3)],Delta).

The insured allow to get compensation

Delta = [f2, f3, f1, r1(john, acc_id3)] ;

false.

2 ?- prove([fullCompensation(john,acc_id3)],Delta).

The insured allow to get compensation

460 euros

Full Compensation

No Compensation-ExpiryCarInsurance

Attack:[r31(john,acc_id3),f6,pr4(john,acc_id3,460,495)]

The insured allow to get compensation

460 euros

Full Compensation

No Compensation-ExpiryCarInsurance

Defence:[r21(john,460,495,acc_id3),f5,f4,f2,f3,f1,r1(john,acc_id3),ass(neg(prefer(r31(john,acc_id3),r21(john,460,495,acc_id3)))))]

No Compensation-ExpiryCarInsurance

Attack:[r31(john,acc_id3),f6,pr4(john,acc_id3,460,495)]

The insured allow to get compensation

460 euros

Full Compensation

No Compensation-ExpiryCarInsurance

Defence:[r21(john,460,495,acc_id3),f5,f4,f2,f3,f1,r1(john,acc_id3),ass(neg(prefer(r31(john,acc_id3),r21(john,460,495,acc_id3)))))]

Attack:[pr4(john,acc_id3,460,495)]

No Compensation-ExpiryCarInsurance

false.

3 ?- prove([neg(fullCompensation(john,acc_id3))],Delta).

Delta = [f6, r31(john, acc_id3)] ;

false.

Βλέπουμε ότι έχουμε πάρει θετική απάντηση στην πρώτη ερώτηση όπως και έπρεπε. Στη συνέχεια τρέξαμε και τα δύο είδους ερωτήματα σχετικά με την ολική αποζημίωση για να μπορέσουμε να δούμε πώς το σύστημα αντιδρά σε θετικές ή αρνητικές ερωτήσεις. Στο δεύτερο ερώτημα βλέπουμε ότι δεν επιτρέπεται η ολική αποζημίωση (αρνητική απάντηση) και τεκμηριώνεται με τα attacks των κανόνων προτεραιότητας, όμως παρουσιάζεται το ποσό της πιθανής αποζημίωσης. Στο τελευταίο ερώτημα, παίρνουμε απάντηση στο Delta, σύμφωνα με το κανόνα r31 που αναιρεί το δικαίωμα αποζημίωσης όταν έχει λήξει η Ασφάλισή του. Βλ. παράρτημα Γ.

Case 4: Έχουμε στοιχεία για τον συγκεκριμένο Ασφαλισμένο τα οποία αρχικά του επιτρέπουν να πάρει αποζημίωση. Το ποσό του συμβολαίου ανέρχεται στα €640 και η ζημιά έχει εκτιμηθεί στα €1020. Έχει διαπιστωθεί ότι στο ατύχημα, ο Ασφαλισμένος οδηγούσε παράνομα, δηλαδή χωρίς άδεια οδήγησης.

Αφού τηρούνται οι αρχικές βασικές προϋποθέσεις κανονικά επιτρέπεται στον Ασφαλισμένο να πάρει αποζημίωση. Σύμφωνα με τα ποσά του συμβολαίου και της ζημιάς συμπεραίνεται η μέγιστη αποζημίωση, όμως δεν πρέπει να του επιτραπεί αφού οδηγούσε παράνομα κι αυτό του αποτρέπει το δικαίωμα αποζημίωσης.

Τα δεδομένα μας έχουν ως εξής:

```
rule(f1, haveCarInsurance(john), []).
rule(f2, in_cyprus_limits(acc_id4), []).
rule(f3, haveAccident(john,acc_id4), []).
rule(f4, contract_cost(640), []).
rule(f5, more_than_max_limit(1020,640), []).
rule(f6, neg(have_driver_license(john)), []).
```

Τρέχουμε τα πιο ερωτήματα για να δούμε αν πράγματι συμφωνούν με αυτά της Εταιρείας και παίρνουμε τα πιο κάτω αποτελέσματα:

```
1 ?- prove([allowedCompensation(john,acc_id4)],Delta).
```

The insured allow to get compensation

```
Delta = [f2, f3, f1, r1(john, acc_id4)] ;
```

false.

```
2 ?- prove([maxCompensation(john,acc_id4)],Delta).
```

The insured allow to get compensation

640 euros

Max Compensation

No Compensation, driven illegally(without driver license)

Attack:[r43(john,acc_id4),f6,r4(john),pr5(john,1020,640,acc_id4)]

The insured allow to get compensation

640 euros

Max Compensation

No Compensation, driven illegally(without driver license)

Defence:[r2(john,1020,640,acc_id4),f5,f4,f2,f3,f1,r1(john,acc_id4),ass(neg(prefer(r43(john,acc_id4),r2(john,1020,640,acc_id4)))))]

No Compensation, driven illegally(without driver license)

Attack:[r43(john,acc_id4),f6,r4(john),pr5(john,1020,640,acc_id4)]

The insured allow to get compensation

640 euros

Max Compensation

No Compensation, driven illegally(without driver license)

Defence:[r2(john,1020,640,acc_id4),f5,f4,f2,f3,f1,r1(john,acc_id4),ass(neg(prefer(r43(john,acc_id4),r2(john,1020,640,acc_id4)))))]

Attack:[pr5(john,1020,640,acc_id4)]

No Compensation, driven illegally(without driver license)

false.

3 ?- prove([neg(maxCompensation(john,acc_id4))],Delta).

Delta = [f6, r4(john), r43(john, acc_id4)] ;

false.

Στο πρώτο ερώτημα, με βάση τα γεγονότα που δώσαμε και τον αρχικό βασικό κανόνα έχουμε θετική απάντηση που φαίνεται μέσα από το Delta. Στο δεύτερο ερώτημα, αν και φαίνεται ο λόγος για τον οποίο δεν μπορεί να πραγματοποιηθεί η αποζημίωση, εμφανίζεται όμως και το ποσό το οποίο θα επιτρεπόταν να πάρει ο Ασφαλισμένος. Έπειτα, στο τρίτο ερώτημα βλέπουμε ότι δεν επιτρέπεται στον Ασφαλισμένο να πάρει την μέγιστη αποζημίωση όπως και θα έπρεπε, που φαίνεται από τους κανόνες προτεραιότητας και τη σύγκρουση που δημιουργούν βλ παράρτημα Γ.

Case 5: Μια άλλη περίπτωση για έλεγχο αποζημίωσης είναι ο συγκεκριμένος Ασφαλισμένος να έχει το δικαίωμα αποζημίωσης αφού το ατύχημα έχει γίνει εντός της Κυπριακής Επικράτειας. Έχει συμβόλαιο με ποσό €725 και η ζημιά για το συγκεκριμένο ατύχημα ανέρχεται στα €982. Η αστυνομία έχει αποδείξει ότι στο συγκεκριμένο ατύχημα ο Ασφαλισμένος οδηγούσε υπό την επήρεια αλκοόλης, όπου αυτό συνεπάγεται ως παραβίαση του οδικού κώδικα.

Με βάση τους κανόνες της Εταιρείας, ο βασικός κανόνας τηρείται αλλά έχουμε την απόδειξη της Αστυνομίας που είναι πιο ισχυρή, έτσι ώστε δεν μπορεί ο Ασφαλισμένος να πάρει την μέγιστη αποζημίωση όπως και θα του επιτρεπόταν σύμφωνα με τα ποσά του συμβολαίου και της ζημιάς.

Έχουμε τα πιο κάτω δεδομένα:

```
rule(f1, haveCarInsurance(john), []).
rule(f2, in_cyprus_limits(acc_id5), []).
rule(f3, haveAccident(john,acc_id5), []).
rule(f4, contract_cost(725), []).
rule(f5, more_than_max_limit(982,725), []).
rule(f6, confirmBythePolice_Alcohol(john), []).
```

Για να αποδείξουμε την εγκυρότητα των αποτελεσμάτων μας, τρέχουμε τα συνήθη ερωτήματα και παρατηρούμε τις απαντήσεις για να μπορέσουμε να συγκρίνουμε:

1 ?- prove([allowedCompensation(john,acc_id5)],Delta).

The insured allow to get compensation

Delta = [f2, f3, f1, r1(john, acc_id5)] ;

false.

2 ?- prove([neg(maxCompensation(john,acc_id5))],Delta).

Delta = [f6, f3, r5(john, acc_id5), r52(john, acc_id5)] ;

false.

3 ?- prove([maxCompensation(john,acc_id5)],Delta).

The insured allow to get compensation

725 euros

Max Compensation

No Compensation, confirmBythePolice_Alcohol

Attack:[r52(john,acc_id5),f6,f3,r5(john,acc_id5),pr8(john,982,725,acc_id5)]

The insured allow to get compensation

725 euros

Max Compensation

No Compensation, confirmBythePolice_Alcohol

Defence:[r2(john,982,725,acc_id5),f5,f4,f2,f3,f1,r1(john,acc_id5),ass(neg(prefer(r52(john,acc_id5),r2(john,982,725,acc_id5)))))]

No Compensation, confirmBythePolice_Alcohol

Attack:[r52(john,acc_id5),f6,f3,r5(john,acc_id5),pr8(john,982,725,acc_id5)]

The insured allow to get compensation

725 euros

Max Compensation

No Compensation, confirmBythePolice_Alcohol

Defence:[r2(john,982,725,acc_id5),f5,f4,f2,f3,f1,r1(john,acc_id5),ass(neg(prefer(r52(john,acc_id5),r2(john,982,725,acc_id5)))))]

Attack:[pr8(john,982,725,acc_id5)]

**No Compensation, confirmBythePolice_Alcohol
false.**

Και στις τρεις απαντήσεις έχουμε πάρει τα σωστά αποτελέσματα. Ενώ αρχικά επιτρέπεται στον john να πάρει αποζημίωση, ο κανόνας που αποδεικνύει ότι οδηγούσε με την επήρεια αλκοόλ, έχει υψηλότερη προτεραιότητα από τον κανόνα που του δίνει τη μέγιστη αποζημίωση. Αυτό φαίνεται στα αποτελέσματα από τα attacks και τεκμηριώνεται κατάλληλα από τους κανόνες προτεραιότητας. Βλ παράρτημα Γ.

Case 6: Στο συγκεκριμένο παράδειγμα που μας δόθηκε, για το ατύχημα, ο Ασφαλισμένος μπορεί να πάρει αποζημίωση αφού τηρείται ο αρχικός βασικός κανόνας. Έχει στο συμβόλαιο του το καταγεγραμμένο ποσό των €345 και η ζημιά του ατυχήματος εκτιμήθηκε στα €190. Επίσης η Ασφάλιση Αυτοκινήτου του είναι σε ισχύ και ο Ασφαλισμένος οδηγούσε χωρίς την επήρεια αλκοόλης και οποιασδήποτε παράνομης ουσίας. Η αστυνομία έχει αποδείξει όμως ότι ο Ασφαλισμένος στο συγκεκριμένο ατύχημα έχει ενεργήσει με δόλο.

Αφού λοιπόν ο βασικός κανόνας τηρείται, αρχικά μπορεί να πάρει αποζημίωση. Στη συνέχεια όμως ο Ασφαλισμένος δεν μπορεί να πάρει την ολική αποζημίωση όπως και θα του επιτρεπόταν σύμφωνα με τα ποσά του συμβολαίου και της ζημιάς, αφού έχει ενεργήσει με δόλο, που του στερεί το δικαίωμα.

Έχουμε τα πιο κάτω δεδομένα:

```
rule(f1, haveCarInsurance(john), []).
rule(f2, in_cyprus_limits(acc_id6), []).
rule(f3, haveAccident(john,acc_id6), []).
rule(f4, contract_cost(345), []).
rule(f5, less_than_max_limit(190,345), []).
rule(f6, neg(expiryCarInsurance(john)), []).
rule(f7, neg(confirmBythePolice_Alcohol(john)), []).
rule(f8, neg(confirmBythePolice_Doping(john)), []).
```

rule(f9, confirmBythePolice_acted_fraudulently(john,acc_id6), []).

Για να αποδείξουμε την ορθότητα των αποτελεσμάτων μας, τρέχουμε τα πιο κάτω ερωτήματα και παρατηρούμε τις απαντήσεις για να μπορέσουμε να συγκρίνουμε:

1 ?- prove([allowedCompensation(john,acc_id6)],Delta).

The insured allow to get compensation

Delta = [f2, f3, f1, r1(john, acc_id6)] ;

false.

2 ?- prove([fullCompensation(john,acc_id6)],Delta).

The insured allow to get compensation

190 euros

Full Compensation

No Compensation,Person acted fraudulently

The insured allow to get compensation

Attack:[r61(john,acc_id6),f9,pr12(john,190,345,acc_id6)]

The insured allow to get compensation

190 euros

Full Compensation

No Compensation,Person acted fraudulently

Defence:[r21(john,190,345,acc_id6),f5,f4,f2,f3,f1,r1(john,acc_id6),ass(neg(prefer(r61(john,acc_id6),r21(john,190,345,acc_id6)))))]

No Compensation,Person acted fraudulently

The insured allow to get compensation

No Compensation,Person acted fraudulently

Attack:[r61(john,acc_id6),f9,pr12(john,190,345,acc_id6)]

The insured allow to get compensation

190 euros

Full Compensation

No Compensation,Person acted fraudulently

Defence:[r21(john,190,345,acc_id6),f5,f4,f2,f3,f1,r1(john,acc_id6),ass(neg(prefer(r61(john,acc_id6),r21(john,190,345,acc_id6)))))]

Attack:[pr12(john,190,345,acc_id6)]

No Compensation,Person acted fraudulently

false.

3 ?- prove([neg(fullCompensation(john,acc_id6))],Delta).

Delta = [f9, r61(john, acc_id6)] ;

false.

Έχουμε ακριβώς ότι περιμέναμε ως αποτέλεσμα Αρχικά παίρνουμε θετική απάντηση και στη συνέχεια λόγω του ότι ο κανόνας που υποδηλώνει ότι ένα άτομο ενεργησε με δόλο σε ένα δυστύχημα, υπερισχύει του βασικού κανόνα για παροχή ολικής αποζημίωσης. Αυτό φαίνεται καλύτερα από τους κανόνες προτεραιότητας, βλ παράρτημα Γ.

5.5.1 Πίνακας Αξιολόγησης για Αποζημίωση σε Περίπτωση Απαίτησης

	haveCar Insuranc e	in_cyprusli mits	Have Accident	Contract cost	Damage	Expiry Car Insurance	violation OfDrivin gCode	driven illegally	acted_fra udulently	Less Than Max Limit	More Than Max Limit	Query1: Allowed Compensation Message1	Query2: Max/Full Compensa tion Message2	Compare Results
Case1 (acc_id1)	√	√	√	520	700	-	-	-	-	-	√	True The insured allow to get compensation	True 520 euros Max Compensati on	Same as Company's
Case2 (acc_id2)	√	√	√	440	355	-	-	-	-	√	-	True The insured allow to get compensation	True 355 euros Full Compensati on	Same as Company's
Case3 (acc_id3)	√	√	√	495	460	√	-	-	-	√	-	True The insured allow to get compensation	False No Compensati onExpiryC arInsurance	Same as Company's (presents alla the messages about the compensati on)
Case4 (acc_id4)	√	√	√	640	1020	-	-	√ (NoDriver License)	-	-	√	True The insured allow to get compensation	False NoCompen sation, driven illegally	Same as Company's (presents alla the messages about the compensati on)
Case5 (acc_id5)	√	√	√	725	982	-	√ (Alcohol)	-	-	-	√	True The insured allow to get compensation	False NoCompen sation,confi rmBythePo liceAlcohol	Same as Company's (presents alla the messages about the compensati on)

Case6 (acc_id6)	√	√	√	345	190	-	-	-	√	√	-	True The insured allow to get compensation	False NoCompen sation,Pers onactedfrau dulently	Same as Company's (presents alla the messages about the compensati on)
--------------------	---	---	---	-----	-----	---	---	---	---	---	---	---	---	---

5.6 Αξιολόγηση Αποτελεσμάτων για Ασφάλιση Υγείας

Case 1: Άτομο ηλικίας 29 ετών μέλος της Ευρωπαϊκής Ένωσης, με ποινικό μητρώο που ξεπερνά τα δέκα (10) έτη, αφού έχει επιλέξει το κατάλληλο πρόγραμμα υγείας που τον ενδιαφέρει και έχει συμπληρώσει το ερωτηματολόγιο υγείας το έχει στείλει για έλεγχο στην Εταιρεία. Τα δεδομένα έχουν ελεγχθεί με επιτυχία και το συγκεκριμένο άτομο έχει δώσει το ποσό των €140 ως πρώτη προκαταβολή για να μπορέσει το συμβόλαιο να είναι σε ισχύ. Κατά τη δεύτερη δόση έχει εξασφαλίσει το ποσό των €80.

Βάση των πιο πάνω, ο πελάτης έχει το δικαίωμα να αποκτήσει Ασφάλιση Υγείας. Ως αποτέλεσμα όμως, το συμβόλαιό του δεν θα είναι σε ισχύ αφού δεν έχει δοθεί το σωστό ποσό για την δεύτερη δόση.

Έχουμε προσαρμόσει τα δεδομένα ως εξής:

rule(f1, member_in_EE(john), []).

rule(f2, person_age(id1,29), []).

rule(f3, criminal_record(john), []).

rule(f4, more_than_10years(john), []).

rule(f5, choose_MedicalExpensesInsurance(john,id1), []).

rule(f6, complete_questionnaire(john,id1), []).

rule(f7, send_proposal_request_to_office(john,id1), []).

rule(f8, check_data(john,id1,y), []).

rule(f9, installment(first,140,mId1), []).

rule(f10, installment2(second,80,mId1), []).

Τρέχουμε τα τρία πιο κάτω ερωτήματα για να μπορέσουμε να συγκρίνουμε τα αποτελέσματα:

1 ?- prove([adoption_of_contract_MEI(john,id1)],Delta).

Person has the right age and he is member in EE

Delta = [f2, f1, r1(john, id1)] ;

false.

2 ?- prove([haveMedicalInsurance(john,id1,mId1)],Delta).

Person has the right age and he is member in EE

Person`s data are valid

Delta = [f8, f7, f6, f5, f2, f1, r1(john, id1), r3(john, y, id1, mId1)] ;

false.

3 ?- prove([medicalInsuranceIsInForce(john,id1,mId1)],Delta).

Person has the right age and he is member in EE

Person`s data are valid

Person has the right age and he is member in EE

Person`s data are valid

The contract is not valid,the second installment is less than 100 euros

Attack:[r6(john,80,mId1,id1,second),f10,pr1(john,140,80,mId1,id1,first,second)]

Person has the right age and he is member in EE

Person`s data are valid

The contract is not valid,the second installment is less than 100 euros

Defence:[r4(john,140,mId1,id1,first),f9,f8,f7,f6,f5,f2,f1,r1(john,id1),r3(john,y,id1,mId1),as
s(neg(prefer(r6(john,80,mId1,id1,second),r4(john,140,mId1,id1,first))))]

Person has the right age and he is member in EE

Person`s data are valid

The contract is not valid,the second installment is less than 100 euros

Attack:[r6(john,80,mId1,id1,second),f10,pr1(john,140,80,mId1,id1,first,second)]

Person has the right age and he is member in EE

Person`s data are valid

The contract is not valid ,the second installment is less than 100 euros

Defence:[r4(john,140,mId1,id1,first),f9,f8,f7,f6,f5,f2,f1,r1(john,id1),r3(john,y,id1,mId1),as
s(neg(prefer(r6(john,80,mId1,id1,second),r4(john,140,mId1,id1,first))))]

Attack:[pr1(john,140,80,mId1,id1,first,second)]

The contract is not valid,the second installment is less than 100 euros

false.

Παρατηρούμε ότι έχουμε τα αναμενόμενα αποτελέσματα. Αν και η πρώτη δόση έχει πληρωθεί με βάση τους όρους της Εταιρείας, λόγω του ότι η δεύτερη δόση δεν ξεπερνούσε τα €100 τότε δεν μπορεί το συμβόλαιο του πελάτη να είναι πλέον σε ισχύ και αυτό φαίνεται από τη δύναμη του κανόνα r5 και το attack που προκαλεί ο κανόνας στους υπόλοιπους κανόνες όπως φαίνεται στο παράρτημα Δ.

Case 2: Εδώ εξετάζουμε την περίπτωση στην οποία άτομο μέλους της Ευρωπαϊκής Ένωσης ηλικίας 42 ετών με φάκελο ποινικού μητρώου για έξι (6) χρόνια, έχει επιλέξει το κατάλληλο πρόγραμμα υγείας που αρμόζει στο δικό του ιστορικό, έχει συμπληρώσει το ερωτηματολόγιο υγείας και το έχει στείλει στην Εταιρεία. Έχουν ελέγξει τα δεδομένα με επιτυχία. Ως αρχική προκαταβολή έδωσε το ποσό των €100. Κατά τη δεύτερη δόση έχει εξασφαλίσει το ποσό των €150.

Σχετικά με τα πιο πάνω, ο πελάτης δεν να αποκτήσει Ασφάλιση Υγείας λόγω του ότι ο φάκελος για το ποινικό του μητρώο είναι κάτω από δέκα χρόνια, ακόμα κι αν οι δύο δόσεις έχουν πληρωθεί ορθά.

Πιο κάτω φαίνονται τα δεδομένα της συγκεκριμένης περίπτωσης που περιγράψαμε:

rule(f1, member_in_EE(john), []).

rule(f2, person_age(id2,42), []).

rule(f3, criminal_record(id2,john), []).

rule(f4, less_than_10years(id2, john), []).
 rule(f5, choose_MedicalExpensesInsurance(john, id2), []).
 rule(f6, complete_questionnaire(john, id2), []).
 rule(f7, send_proposal_request_to_office(john, id2), []).
 rule(f8, check_data(john, id2, y), []).
 rule(f9, installment(first, 100, mId2), []).
 rule(f10, installment2(second, 150, mId2), []).

Τρέχουμε τα πιο κάτω ερωτήματα παίρνοντας τις πιο κάτω απαντήσεις:

1 ?- prove([adoption_of_contract_MEI(john, id2)], Delta).

Person has the right age and he is member in EE

Person has criminal record under 10 years

Attack:[r2(john, id2), f3, pr3(john, id2), f4]

Person has the right age and he is member in EE

Person has criminal record under 10 years

Defence:[r1(john, id2), f2, f1, ass(neg(prefer(r2(john, id2), r1(john, id2)))))]

Person has criminal record under 10 years

Attack:[r2(john, id2), f3, pr3(john, id2), f4]

Person has the right age and he is member in EE

Person has criminal record under 10 years

Defence:[r1(john, id2), f2, f1, ass(neg(prefer(r2(john, id2), r1(john, id2)))))]

Attack:[pr3(john, id2), f4]

Person has criminal record under 10 years

false.

2 ?- prove([medicalInsuranceIsInForce(john,id2,mId2)],Delta).

false.

Παρατηρούμε ότι αφού λόγω του κανόνα r2 ο πελάτης δεν μπορεί καν να αποκτήσει συμβόλαιο Ασφάλισης Υγείας όπως και φαίνεται από τα attacks στο αποτέλεσμα και στο παράρτημα Δ.

Case 3: Στο συγκεκριμένο παράδειγμα, πελάτης μέλους της Ευρωπαϊκής Ένωσης ηλικίας 34 ετών, χωρίς φάκελο που να επιβαρύνει το ποινικό του μητρώο, έχει επιλέξει το κατάλληλο πρόγραμμα υγείας χωρίς να υπάρξει κάποια αρνητική απάντηση για τα στοιχεία του από την Εταιρεία. Στη συνέχεια έχει δώσει το ποσό των €75 ως αρχική προκαταβολή.

Συμπερασματικά, για την ισχύ του συμβολαίου Ασφάλισης Υγείας το αποτέλεσμα που έδωσε η Εταιρεία ως απάντηση είναι αρνητικό, ενώ για το adoption_of_contract_MEI και haveMedicalInsurance τα αποτελέσματα σύμφωνα με τα στοιχεία είναι θετικά.

Τα στοιχεία έχουν ως εξής:

rule(f1, member_in_EE(john), []).

rule(f2, person_age(id3,34), []).

rule(f3, choose_MedicalExpensesInsurance(john,id3), []).

rule(f5, complete_questionnaire(john,id3), []).

rule(f6, send_proposal_request_to_office(john,id3), []).

rule(f7, check_data(john,id3,y), []).

rule(f8, installment(first,75,mId3), []).

Τρέχουμε τα πιο κάτω ερωτήματα για να συγκρίνουμε τα αποτελέσματα:

1 ?- prove([adoption_of_contract_MEI(john,id3)],Delta).

Person has the right age and he is member in EE

Delta = [f2, f1, r1(john, id3)] ;

false.

2 ?- prove([haveMedicalInsurance(john,id3,mId3)],Delta).

Person has the right age and he is member in EE

Person`s data are valid

Delta = [f7, f6, f5, f3, f2, f1, r1(john, id3), r3(john, y, id3, mId3)] ;

false.

3 ?- prove([neg(medicalInsuranceIsInForce(john,id3,mId3))],Delta)

Person has the right age and he is member in EE

Person`s data are valid

The first installment is less than 100 euros

Delta = [f8, f7, f6, f5, f3, f2, f1, r1(john, id3), r3(..., ..., ..., ...)|...] ;

false.

Εμφανίζονται τα αποτελέσματα ορθά, αν και στο τελευταίο ερώτημα για το αν το συμβόλαιο μπορεί αν είναι σε ισχύ, παρουσιάζονται και τα μηνύματα των πιο πάνω περιπτώσεων που αφορούν τα προηγούμενα ερωτήματα. Δεν πληρούνται οι προϋποθέσεις του κανόνα r4 (βλ. παράρτημα Δ) με αποτέλεσμα να παίρνουμε αρνητικό αποτέλεσμα και να συμφωνούμε με αυτό της Εταιρείας.

5.6.1 Πίνακας Αξιολόγησης Ασφάλισης Υγείας

	Member EE	person_age	Criminal record	chooseMedicalExpensesInsurance	completequestionnaire	sendproposalrequest	Check Data(y/n)	Installment,first	Installment,second	Query1: adoptionofcontract Message1	Query2: haveMedicalInsurance Message2	Query3: medicalInsuranceIsInForce Message3	Compare Results
Case1 (id1,mId1)	✓	29	✓ (>10years)	✓	✓	✓	y	140	80	True Person has the right age and he is member in EE	True Person's data are valid	False The contract is not valid. The second installment is less than 100euros	Same as Company's
Case2 (id2,mId2)	✓	42	✓ (<10years)	✓	✓	✓	y	100	150	False Person has criminal record under 10 years	False Noadoption of contract	False Noadoption of contract	Same as Company's
Case3 (id3,mId3)	✓	34	×	✓	✓	✓	y	75	×	True Person has the right age and he is member in EE	True Person's data are valid	False The first installment is less than 100 euros	Same as Company's

Κεφάλαιο 6

Συμπεράσματα και Μελλοντική Εργασία

6.1 Συμπεράσματα	90
6.2 Μελλοντική Εργασία	92

6.1 Συμπεράσματα

Σ' αυτή τη διπλωματική εργασία έχουμε παρουσιάσει Ασφαλιστικές Πολιτικές μετατρέποντάς τες από τη φυσική τους γλώσσα. Έχουμε δείξει πώς μια πολιτική μπορεί να έχει πιο υψηλή προτεραιότητα από κάποια άλλη και πώς αντιδρά το σύστημα σε τέτοιες περιπτώσεις όταν χρειάζεται να αποφασίσει. Το σύστημα μας υλοποιήθηκε χρησιμοποιώντας τον Γοργία, ένα εργαλείο που τρέχει σε ένα prolog περιβάλλον [7]. Με μια απλή περιγραφή των Πολιτικών Ασφάλειας, το πρόγραμμά μας είναι σε θέση να παίρνει κατάλληλες αποφάσεις σε συγκεκριμένα σενάρια και περιπτώσεις. Επίσης οι πολιτικές ασφάλειας μπορούν να επεκταθούν πολύ εύκολα προσθέτοντας νέους κανόνες ή αφαιρώντας παλιούς λαμβάνοντας υπόψη κάθε φορά τις υπάρχουσες ανάγκες και απαιτήσεις της Εταιρείας.

Στο προηγούμενο κεφάλαιο (Κεφ.5), όπου έχει γίνει η σύγκριση και η αξιολόγηση με πραγματικά γεγονότα και περιπτώσεις που πάρθηκαν από την Εταιρεία, έχουμε καταλήξει στο συμπέρασμα ότι το πρόγραμμά μας σε συνδυασμό με τις Πολιτικές Ασφάλειας που έχουμε δημιουργήσει, αποφασίζει και δουλεύει τις πλείστες φορές δίνοντας τα σωστά αποτελέσματα και ποσά με αυτά που αναμέναμε να βγάξει. Μπορούμε να πούμε πώς ένας

από τους κύριους στόχους από την αρχή της εκπόνησης της διπλωματικής εργασίας, έχει εν μέρει επιτευχθεί, οι Πολιτικές που θα εφαρμόζαμε να μπορούσαν να ανταποκριθούν σε οποιεσδήποτε ξεχωριστές περιπτώσεις και ερωτήματα.

Στη συνέχεια, με την εκπόνηση της παρούσας διπλωματικής εργασίας, μπορούμε να εκφράσουμε εκτός από τα καθορισμένα συμπεράσματα και αξιολογήσεις, τις αδυναμίες σχετικά με τη δουλειά που έχουμε κάνει, βλέποντας την ολοκληρωμένη. Το σύστημά μας, αφού το είδαμε και στην φάση της εκτέλεσης στο προηγούμενο κεφάλαιο, μπορεί να θεωρηθεί αδύναμο ως προς την παρουσίαση των αποτελεσμάτων. Όπως μπορούμε να παρατηρήσουμε, τα μηνύματα σχετικά με τις επεξηγήσεις των κανόνων εμφανίζονται περισσότερες από μια φορές το καθένα.

Επιπλέον, κάποιες φορές είδαμε ότι παρουσιάζει αδυναμία στην εμφάνιση μοναδικών απαντήσεων. Εμφανίζονται δηλαδή περισσότερες από μία απαντήσεις έχοντας μια μεγάλη λίστα με όλα τα attacks, defences και prefers που δημιουργούνται.

Θα ήταν πολύ πιο όμορφο και ευπαρουσίαστο, αν τα αποτελέσματα εμφανίζονταν πιο επεξηγηματικά, με περισσότερα μηνύματα, για εύκολη κατανόηση από τους χρήστες.

Όσον αφορά τον χρόνο εκτέλεσης των Ασφαλιστικών παραδειγμάτων δεν σημειώθηκε κάποιο πρόβλημα, αντίθετα όμως, μπορούμε να πούμε πώς όλα τα παραδείγματα έτρεχαν χωρίς καμιά καθυστέρηση. Αυτό οφείλεται και στο γεγονός ότι οι Ασφαλιστικές Πολιτικές δεν είναι ιδιαίτερα μεγάλες και ούτε αποτελούνται από τεράστιο αριθμών γραμμών κανόνων και προτεραιοτήτων

Κλείνοντας με το κεφάλαιο των συμπερασμάτων καλό θα ήταν να κάνουμε μια αυτοκριτική για όλη την εργασία. Μπορούμε λοιπόν, να καταλήξουμε στο συμπέρασμα ότι ίσως αν είχαμε την δυνατότητα να το κάνουμε πάλι από την αρχή, να αλλάζαμε κάποια σημεία στα οποία θα ήμασταν πιο προσεκτικοί και ακριβείς, έχοντας όσο λιγότερα σφάλματα έως και καθόλου.

6.2 Μελλοντική Εργασία

Στο μέλλον το σύστημα μπορεί να συνεχιστεί και να εξελιχτεί. Θα μπορούσε να υλοποιηθεί ένα εργαλείο διαπροσωπείας (GUI) το οποίο θα διευκόλυνε τους διαχειριστές του συστήματος στον ορισμό των Πολιτικών Ασφάλειας. Ο χρήστης να μπορεί να έχει τη δυνατότητα να εισάγει τα στοιχεία και τα δεδομένα κάθε πελάτη, μέσα από ένα ολοκληρωμένο περιβάλλον όπου θα εμφανίζονται τα κατάλληλα labels, buttons και μηνύματα, βοηθώντας στην εύκολη καταχώρησή τους.

Έτσι, μετά τη μετατροπή του σ' ένα όμορφο σύστημα, θα μπορούσε να αποτελέσει ένα χρήσιμο εργαλείο και εγχειρίδιο καθημερινής χρήσης για την Εταιρεία, για την αυτοματοποίηση και λήψη αποφάσεων, καθώς θα μπορεί να απαλλάξει τους χρήστες του από ένα μεγάλο φόρτο εργασίας κατά τη διαδικασία της λήψης αποφάσεων .

Επιπλέον, θα μπορούσε να αυτοματοποιηθεί η διαδικασία εισαγωγής των κανόνων ασφάλειας που επιλύουν διάφορα νέα ζητήματα και απαιτήσεις της Εταιρείας χωρίς να χρειάζεται η συμβολή του διαχειριστή.

Τέλος, μια επιπρόσθετη μελλοντική επέκταση, είναι η πρόσθεση και η ολοκλήρωση κι άλλων Πολιτικών που ασχολείται η Ασφαλιστική Εταιρεία, όπως για παράδειγμα σχετικά με την Ασφάλιση Κατοικίας, Επιχείρησης, Ταξιδιού, Περιουσίας κλπ, με παρόμοιο τρόπο όπως πιο πάνω.

Βιβλιογραφία

- [1] "Security Policies
<http://www.comptechdoc.org/independent/security/policies/> "
- [2] Gorgias. "Argumentation and Abduction,
<http://www2.cs.ucy.ac.cy/~nkd/gorgias/>".
- [3] "Πολιτικές Ασφάλειας
<http://webcache.googleusercontent.com/search?q=cache:R8Be7ThOHJYJ:www.dpa.gr/url/IT/y>"
- [4] "Διαγράμματα Ροής
http://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%AC%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B1_%CF%81%CE%BF%CE%AE%CF%82"
- [5] Yannis Dimopoulos, Antonis Kakas, "Logic Programming Without Negation as Failure", 1995
- [6] Victor Noel and Antonis Kakas, "Gorgias-C: Extending Argumentation with Constraint Solving", Universite de Toulouse, France, University of Cyprus, Cyprus, 2009, retrieved from "Logic Programming and Nonmonotonic Reasoning", p.515-540
- [7] Prolog. "SWIProlog, <http://www.swi-prolog.org/>".

Παράρτημα Α

Πιο κάτω παρουσιάζονται οι κανόνες που μετατράπηκαν για την **Ασφάλιση Αυτοκινήτου**.

% CAR INSURANCE CONTRACT

% This example is taken from the article 'Car Insurance Terms' Insurance Company.

% It is used here to decide based on car insurance policies if a client can have Car Insurance.

:- compile('./lib/gorgias').

:- compile('./ext/lpwnf').

%the insurance liability vehicles applies to the boundaries of the Cypriot Territory, for the Member States of the European Union

rule(r1(Person),haveCarInsurance(Person),[member_in_EE(Person),cyprus_signs(Person)]).

rule(r2(Person), neg(haveCarInsurance(Person)), [neg(in_cyprus_limits(Person))]).

%the legal representatives of the legal person insured or the Company has not acquired legal personality-criminal record

rule(r3(Person), neg(haveCarInsurance(Person)), [criminal_record(Person)]).

rule(r4(Person), neg(haveCarInsurance(Person)), [neg(have_driver_license(Person))]).

rule(pr1(Person), prefer(r3(Person),r1(Person)), [less_than_10years(Person)]):-write('The person has criminal record under 10 years'),nl.

rule(pr2(Person), prefer(r1(Person),r3(Person)), [more_than_10years(Person)]).

rule(pr3(Person), prefer(r2(Person),r1(Person)), []).

%Motor vehicle having the normally based in the Cypriot territory andleaving the boundaries thereof, shall be provided with international insurance certificate (greencard)

```
rule(pr4(Person), prefer(r2(Person),r1(Person)), [neg(have_green_card(Person))]).:-  
write('The person can not obtain CarInsurance because has not green card'),nl.
```

```
rule(pr5(Person), prefer(r1(Person),r2(Person)), [have_green_card(Person)]):-write('The  
person can obtain CarInsurance because of green card'),nl.
```

```
rule(pr6(Person), prefer(r4(Person),r1(Person)), []).%:-write('The Person can not obtain  
CarInsurance because has not car license'),nl.
```

Παράρτημα Β

Εδώ έχουμε τους κανόνες σχετικά με την **Ακύρωση Ασφαλιστικής Σύμβασης**.

% CANCELLATION INSURANCE CONTRACT

% This example is taken from the article 'Vehicle Insurance Terms' by Insurance Company.

% It is used here to decide based on cancellation policies if a client can cancel his/her Car Insurance.

:- compile('./lib/gorgias').

:- compile('./ext/lpwnf').

% The insured may terminate the insurance contract at any time with a denunciation served to the Company or an authorized this mediator(agent). The results of the complaint made after thirty (30) days on receipt there of to the Company.

rule(r1(Person,ConID),sendRequest(Person,ConID),[haveInsurance(Person,ConID),done_complaint_to_company(Person,ConID)]).

rule(r2(Person,ConID),sendRequest(Person,ConID),[haveInsurance(Person,ConID),done_complaint_to_agent(Person,ConID)]).

rule(r3(ConID,Person,Days,Answer), cancellationFromInsured(ConID),
[sendRequest(Person,ConID),after30Days(Days),Days>30,sendResultsTo(Answer,Person,ConID),Answer=y,write('Data is ok, cancellation from insured is done')]).

rule(r31(ConID,Person,Amount), neg(return_fees_to(Person,Amount)),
[cancellationFromInsured(ConID), hasClaim(Person,ConID),write('Person has claim')]).

rule(r32(ConID,Person,Amount,Month1,Month2,M,CCost),return_fees_to(Person,Amount,
[cancellationFromInsured(ConID),contractCost(CCost),monthStarted(Month1),monthCancelled(Month2),M is 12-(Month2-Month1),Amount is (CCost/12)*M, write(Amount),write('euros')]).

%The Company may, by written declaration, terminate the insurance contract only breach of an essential condition of the Insured. Disclosed that the non Accord educated with the violated essential condition within thirty (30) days after service of complaint, causes the termination of an insurance contract.

```
rule(r4(Person,ConID),sendNoticeTo(Person,ConID),[breach_of_essential_condition(Person),done_complaint_to_insured(Person,ConID)]).
```

```
rule(r5(Person,Days,ConID),cancellationFromCompany(ConID),[sendNoticeTo(Person,ConID),neg(compliance(Person)),after30Days(Days),Days>30]).
```

%The termination of the insurance made notwithstanding the refusal of the recipient of insurance and the Insured to receive it or have not been found in the home addresses not presenting themselves in the post Office for the production handle

```
rule(r6(Person,ConID),cancellationFromCompany(ConID),[sendNoticeTo(Person,ConID),neg(receive_Notice(Person,ConID))]).
```

```
rule(r61(Person,ConID),cancellationFromCompany(ConID),[sendNoticeTo(Person,ConID),neg(find_address(Person))]).
```

```
rule(r62(Person,ConID),cancellationFromCompany(ConID),[sendNoticeTo(Person,ConID),neg(take_complaint_from_postOffice(Person))]).
```

```
rule(r7(Person,ConID), neg(cancellationFromCompany(ConID)),  
[prove_that_no_fault(Person,ConID),sendNoticeTo(Person,ConID),neg(receive_Notice(Person,ConID)), write('No Cancellation,the insured has not received')]).
```

```
rule(r71(Person,ConID), neg(cancellationFromCompany(ConID)),  
[prove_that_no_fault(Person,ConID),sendNoticeTo(Person,ConID),neg(find_address(Person))],write('No Cancellation, address not found')]).
```

```
rule(r72(Person,ConID), neg(cancellationFromCompany(ConID)),  
[prove_that_no_fault(Person,ConID),sendNoticeTo(Person,ConID),neg(take_complaint_from_postOffice(Person))], write('No Cancellation,the insured has not take the complaint')]).
```


rule(r8(Person,ConID,Time), cancellationFromCompany(ConID),
[neg(cancellationFromCompany(ConID)),sendNoticeTo(Person,ConID),after30Days(Days,
Time),Days>30, Time=secondTime, write('The cancellation is valid,the insured not
compliance')]).

rule(r9(ConID,Person,Amount), neg(return_fees_to(Person,Amount)),
[cancellationFromCompany(ConID), hasClaim(Person,ConID),nl,write(' Person has
claim,no return fees'),nl]).

rule(r10(ConID,Person,Amount,Month1,Month2,M,CCost),return_fees_to(Person,Amount,
[cancellationFromCompany(ConID),contractCost(CCost),monthStarted(Month1),monthCa
ncelled(Month2),M is 12-(Month2-Month1),Amount is (CCost/12)*M,
write(Amount),write(' euros'),nl]).

rule(pr1(Person,ConID), prefer(r7(Person,ConID),r6(Person,ConID)), []).

rule(pr2(Person,ConID), prefer(r71(Person,ConID),r61(Person,ConID)), []).

rule(pr3(Person,ConID), prefer(r72(Person,ConID),r62(Person,ConID)), []).

rule(pr4(Person,Days,ConID), prefer(r7(Person,ConID),r5(Person,Days,ConID)), []).

rule(pr5(Person,Days,ConID), prefer(r71(Person,ConID),r5(Person,Days,ConID)), []).

rule(pr6(Person,Days,ConID), prefer(r72(Person,ConID),r5(Person,Days,ConID)), []).

rule(pr7(ConID,Person,Amount,Month1,Month2,M,CCost,ClaimId),prefer(r31(ConID,Pers
on,ClaimId,Amount),r32(ConID,Person,Amount,Month1,Month2,M,CCost)), []).

rule(pr8(ConID,Person,Amount,Month1,Month2,M,CCost,ClaimId),prefer(r9(ConID,Perso
n,ClaimId,Amount),r10(ConID,Person,Amount,Month1,Month2,M,CCost)), []).

Παράρτημα Γ

Κανόνες σχετικά με την Αποζημίωση σε Περίπτωση Απαιτήσης.

% COMPENSATION FOR CAR INSURANCE POLICIES

% This example is taken from the article 'Vehicle Insurance Terms' by Insurance Company. It is used here to decide based on policies if a client can get compensation from an accident.

:- compile('./lib/gorgias').

:- compile('./ext/lpwnf').

% This insurance applies only within the limits of the Cypriot Territory

rule(r1(Person,AccId),allowedCompensation(Person,AccId),[haveCarInsurance(Person),haveAccident(Person,AccId),in_cyprus_limits(AccId), write('The insured allow to get compensation'),nl]).

% Company, compensating the Insured acquires up to the amount of compensation

rule(r2(Person,Damage,Cost,AccId),maxCompensation(Person,AccId),[allowedCompensation(Person,AccId),contract_cost(Cost),more_than_max_limit(Damage,Cost),Damage>=Cost,write(Cost),write(' euros'),nl,write('Max Compensation'),nl]).

rule(r21(Person,Damage,Cost,AccId),fullCompensation(Person,AccId),[allowedCompensation(Person,AccId),contract_cost(Cost),less_than_max_limit(Damage,Cost),Damage<Cost,write(Damage),write(' euros'), nl, write('Full Compensation'),nl]).

rule(r3(Person,AccId),neg(maxCompensation(Person,AccId)),[expiryCarInsurance(Person)]).

rule(r31(Person,AccId),neg(fullCompensation(Person,AccId)),[expiryCarInsurance(Person)]).

% If the Insured Vehicle is driven illegally, as without authorization

rule(r4(Person), driven_illegally(Person), [neg(have_driver_license(Person))]).

rule(r41(Person), driven_illegally(Person), [expiryDriverLicense(Person)]).

rule(r42(Person), driven_illegally(Person), [neg(belongsInInsurance(Person))]).

rule(r43(Person,AccId), neg(maxCompensation(Person,AccId)),[driven_illegally(Person)]).

rule(r44(Person,AccId), neg(fullCompensation(Person,AccId)), [driven_illegally(Person)]).

%by a driver who at the time of the accident is under the influence of alcohol or in-xikon substances in violation of the Driving Code

rule(r5(Person,AccId),violationOfDrivingCode(Person,AccId),[haveAccident(Person,AccId),confirmBythePolice_Alcohol(Person,AccId)]).

rule(r51(Person,AccId),violationOfDrivingCode(Person,AccId),[haveAccident(Person,AccId),confirmBythePolice_Doping(Person,AccId)]).

rule(r52(Person,AccId),neg(maxCompensation(Person,AccId)),[violationOfDrivingCode(Person,AccId)]).

rule(r53(Person,AccId),neg(fullCompensation(Person,AccId)),[violationOfDrivingCode(Person,AccId)]).

%If the insured has acted fraudulently or attempted intentional action

rule(r6(Person,AccId),neg(maxCompensation(Person,AccId)),[confirmBythePolice_acted_fraudulently(Person,AccId)]).

rule(r61(Person,AccId),neg(fullCompensation(Person,AccId)),[confirmBythePolice_acted_fraudulently(Person,AccId)]).

complement(maxCompensation(Person,AccId),fullCompensation(Person,AccId)).

complement(fullCompensation(Person,AccId),maxCompensation(Person,AccId)).

rule(pr1(Person,Damage,Cost,AccId),prefer(r2(Person,Damage,Cost,AccId),r21(Person,Damage,Cost,AccId)), []).

rule(pr2(Person,AccId), prefer(r31(Person,AccId),r1(Person,AccId)), []).

rule(pr3(Person,Damage,Cost,AccId),prefer(r3(Person,AccId),r2(Person,Damage,Cost,AccId)), []).

rule(pr4(Person,Damage,Cost,AccId),prefer(r31(Person,AccId),r21(Person,Damage,Cost,AccId)), []).

rule(pr5(Person,Damage,Cost,AccId),prefer(r43(Person,AccId),r2(Person,Damage,Cost,AccId)), []):-write('No Compensation, driven illegally(without driver license)'),nl.

rule(pr6(Person,AccId), prefer(r44(Person,AccId),r1(Person,AccId)), []).

rule(pr7(Person,Damage,Cost,AccId),prefer(r44(Person,AccId),r21(Person,Damage,Cost,AccId)), []).

rule(pr8(Person,Damage,Cost,AccId),prefer(r52(Person,AccId),r2(Person,Damage,Cost,AccId)), []):-write('No Compensation, confirmBythePolice_Alcohol'),nl.

rule(pr9(Person,Damage,Cost,AccId),prefer(r53(Person,AccId),r21(Person,Damage,Cost,AccId)), []).

rule(pr10(Person,AccId), prefer(r53(Person,AccId),r1(Person,AccId)), []).

rule(pr11(Person,Damage,Cost,AccId),prefer(r6(Person,AccId),r2(Person,Damage,Cost,AccId)), []).

rule(pr12(Person,Damage,Cost,AccId),prefer(r61(Person,AccId),r21(Person,Damage,Cost,AccId)), []):-write('No Compensation,Person acted fraudulently'),nl.

rule(pr13(Person,AccId), prefer(r61(Person,AccId),r1(Person,AccId)), []).

Παράρτημα Δ

Κανόνες για την Ασφάλιση Υγείας.

% MEDICAL INSURANCE CONTRACT

% This example is taken from the article 'Medical Insurance Terms' Insurance Company.

% It is used here to decide based on health policies if a client can have Health Insurance.

:- compile('./lib/gorgias').

:- compile('./ext/lpwnf').

rule(r1(Person, ID), adoption_of_contract_MEI(Person, ID), [member_in_EE(Person), person_age(ID, Age), Age >= 18, Age <= 60, write('Person has the right age and he is member in EE'), nl]).

rule(r2(Person, ID), neg(adoption_of_contract_MEI(Person, ID)), [criminal_record(ID, Person)]).

rule(r3(Person, Answer, ID, HConId), haveMedicalInsurance(Person, ID, HConId), [adoption_of_contract_MEI(Person, ID), choose_MedicalExpensesInsurance(Person, ID), complete_questionnaire(Person, ID), send_proposal_request_to_office(Person, ID), check_data(Person, ID, Answer), Answer=y, write('Person`s data are valid'), nl]).

rule(r4(Person, Amount, HConId, ID, Time), medicalInsuranceIsInForce(Person, ID, HConId), [haveMedicalInsurance(Person, ID, HConId), installment(Time, Amount, HConId), Amount >= 100, Time=first]).

rule(r5(Person, Amount, HConId, ID, Time), neg(medicalInsuranceIsInForce(Person, ID, HConId)), [haveMedicalInsurance(Person, ID, HConId), installment(Time, Amount, HConId), Amount < 100, Time=first, write('The first installment is less than 100 euros'), nl]).

rule(r6(Person, Amount2, HConId, ID, Time2), neg(medicalInsuranceIsInForce(Person, ID, HConId)), [installment2(Time2, Amount2, HConId), Amount2 < 100, Time2=second]).

```
rule(pr1(Person,Amount,Amount2,HConId,ID,Time,Time2),prefer(r6(Person,Amount2,HConId,ID,Time2),r4(Person,Amount,HConId,ID,Time)), []):-write('The contract is not valid,the second installment is less than 100 euros'),nl.
```

```
rule(pr2(Person,Amount,HConId,ID,Time),prefer(r5(Person,Amount,HConId,ID,Time),r4(Person,Amount,HConId,ID,Time)), []).
```

```
rule(pr3(Person,ID), prefer(r2(Person,ID),r1(Person,ID)),[less_than_10years(ID,Person)]):-write('Person has criminal record under 10 years'),nl.
```

```
rule(pr4(Person,ID),prefer(r1(Person,ID),r2(Person,ID)),[more_than_10years(ID,Person)]):-write('Person has criminal record more 10 years'),nl.
```