



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

# ΑΝΑΓΝΩΡΙΣΗ ΝΟΡΜΩΝ ΣΕ ΠΟΛΥΠΡΑΚΤΟΡΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ

---

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

ΜΙΚΡΟΜΑΣΤΟΡΑ ΠΑΡΑΣΚΕΥΗ

**Επιβλέπουσα :** Δασκαλοπούλου Ασπασία, Επίκουρος Καθηγήτρια

**Μέλη επιτροπής :** Αντωνόπουλος Χρήστος, Επίκουρος Καθηγητής

Ακρίτας Αλκιβιάδης, Καθηγητής

Βόλος, Σεπτέμβριος 2015





UNIVERSITY OF THESSALY

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

# NORM IDENTIFICATION IN MULTI – AGENT ENVIRONMENTS

---

MASTER THESIS

**MIKROMASTORA PARASKEVI**

**Supervisor :** Daskalopulu Aspassia, Assistant Professor

**Committee Members :** Antonopoulos Christos, Assistant Professor

Akritas Alkiviadis, Professor

Volos, September 2015

2



## Ευχαριστίες

Ολοκληρώνοντας την παρούσα αυτή διατριβή θα ήθελα να ευχαριστήσω θερμά την επιβλέπουσα καθηγήτριά μου κ. Ασπασία Δασκαλοπούλου για την ευκαιρία που μου έδωσε να πραγματοποιήσω αυτή τη μελέτη, τη βοήθειά της στα πλαίσια αυτής αλλά κυρίως για τη βοήθεια, την καθοδήγηση και την ειλικρινή στήριξη που μου προσέφερε στα χρόνια των προπτυχιακών και μεταπτυχιακών σπουδών μου. Η υπομονή της, η αμέριστη συμπαράστασή της και οι εύστοχες υποδείξεις της βοήθησαν στην ολοκλήρωση αυτής της εργασίας.

Επιπρόσθετα, θα ήθελα να ευχαριστήσω τους δύο επιβλέποντες καθηγητές κ. Ακρίτα Αλκιβιάδη και κ. Αντωνόπουλο Χρήστο για τις συμβουλές και την καθοδήγησή τους καθ' όλη τη διάρκεια των σπουδών μου.

Θα ήθελα να πω ένα μεγάλο ευχαριστώ στην οικογένειά μου, το σύζυγό μου Βασίλη, τους γονείς μου, την αδερφή μου και τη φίλη μου Μαρί για την ειλικρινή στήριξή τους σε κάθε μου εγχείρημα.



*Στο Βασίλη και την πολύτιμή μας Πολυτίμη*



## Περίληψη

Είναι σύνηθες στα πολυπρακτορικά περιβάλλοντα να θεωρείται ότι ο πράκτορας ο οποίος δρα μέσα σε αυτά γνωρίζει εκ των προτέρων τη νόρμα που διέπει το περιβάλλον. Το ερώτημα που τίθεται σε αυτό το σημείο είναι πως μπορεί ένας πράκτορας που εισέρχεται σε ένα σύστημα και δεν γνωρίζει τις νόρμες του να τις αναγνωρίσει και όχι να τις μαθαίνει από π.χ. έναν κεντρικό αρχηγό του συστήματος. Αυτό μπορεί να γίνει εφικτό μέσω του *Μηχανισμού Αναγνώρισης Νορμών* ( Norm Identification Mechanism ).

Με τον όρο «Αναγνώριση Νόρμας» εννοούμε τη διαδικασία η οποία καθιστά ικανό ένα πράκτορα να γνωρίζει τι επιτρέπεται και τι όχι σε ένα περιβάλλον, ουσιαστικά δηλαδή να αναγνωρίζει τις νόρμες που το διέπουν. Ο Μηχανισμός Αναγνώρισης Νορμών ( Norm Identification Mechanism ) στοχεύει στο να απαντήσει στο πώς οι πράκτορες μπορούν να συμπεράνουν μία νόρμα σε ένα πολυπρακτορικό περιβάλλον, διαδικασία ιδιαίτερα σημαντική είτε για την αναγνώριση νορμών σε περιβάλλοντα που τροποποιούνται συχνά ( π.χ. σε ανοιχτά περιβάλλοντα ), είτε για πράκτορες οι οποίοι μετακινούνται από περιβάλλον σε περιβάλλον. Ο μηχανισμός αυτός στηρίζεται στην παρατήρηση των αλληλεπιδράσεων μεταξύ των πρακτόρων του περιβάλλοντος, δηλαδή στην παρατήρηση των πράξεων και των συνεπειών τους.

Η παρούσα μεταπτυχιακή εργασία ασχολείται με τον Μηχανισμό Αναγνώρισης Νορμών και ιδιαίτερα με τον αλγόριθμο Candidate Norm Inference. Γίνεται παρουσίαση του μηχανισμού αναγνώρισης νορμών και του αλγορίθμου Candidate Norm Inference, πάνω στον οποίο βασίζεται και η όλη διαδικασία, η υλοποίηση του αλγορίθμου καθώς επίσης και η αξιολόγησή του.



## Abstract

It is usual in multi-agent environments to consider that an agent who acts within them know *a-priori* the norm ruling the environment. The question at this point is how can an agent who enters a system and who is not aware of the norms, to recognize them and not to learn them, for example, by a central leader of the system. This may be possible through the Norm Identification Mechanism.

By the term "Norm Recognition" we mean the procedure which enables an agent to know what is allowed and what is not in an environment, that is to recognize the norms ruling it. The Norm Identification Mechanism aims to answer the question "how the agents can infer a norm in a multi-agent environment", a process particularly important both for recognition of norms in environments that are frequently modified (eg in open environments) and for agents who move from environment to environment. The mechanism is based on the observation of interactions between agents of the environment, that is the observation of acts and their consequences.

This thesis deals with the Norm Recognition Mechanism and especially with the algorithm Candidate Norm Inference. Here is presented the Norm Identification Mechanism, the algorithm of Candidate Norm Inference, on which the whole process is based, the implementation as well as the evaluation of the algorithm.



## Πίνακας Περιεχομένων

Ευχαριστίες .....	3
<i>Στο Βασίλη και την πολύτιμή μας Πολυτίμη</i> .....	4
Περίληψη .....	5
Abstract.....	6
Πίνακας Περιεχομένων Εικόνων.....	9
1Εισαγωγή.....	11
1.1 Εισαγωγή στην αναγνώριση νορμών σε πολυπρακτορικά συστήματα .....	11
2Πράκτορες και Νόρμες .....	12
2.1 Πράκτορες σε πολυπρακτορικά συστήματα (Multi – agent systems, MAS).....	12
2.1.1 Πράκτορες.....	12
2.1.2 Περιβάλλοντα .....	13
2.1.3 Πολυπρακτορικά Συστήματα (Multi-agent Systems - MAS).....	14
2.1.4 Επικοινωνία Πρακτόρων – Πρωτόκολλα .....	15
2.2 Τι είναι η Νόρμα. ....	17
2.2.1 Νόρμες στις ανθρώπινες κοινωνίες.....	17
2.2.2 Κύκλοι ζωής μιας Νόρμας.....	18
2.3 Πολυπρακτορικά συστήματα που διέπονται από Νόρμες (NorMAS).....	19
2.3.1 Διαφορετικές όψεις των νορμών σε Πολυπρακτορικά συστήματα που διέπονται από Νόρμες (NorMAS) .....	19
2.3.2 Επιβολή Νορμών.....	20
3Μηχανισμός για την εμφάνιση Νορμών ( Role Model ).....	21
3.1 Μηχανισμός Role Model.....	21
4Αρχιτεκτονική για την αναγνώριση νορμών.....	23
4.1 Η διαδικασία αναγνώρισης της νόρμας .....	23
4.2 Μηχανισμός Αναγνώρισης Νορμών. ....	27
4.2.1 Event storage component.....	27
4.2.2 Norm inference component .....	29
5Νόρμες Απαγόρευσης (Prohibition Norms).....	31
5.1 Παράμετροι του μηχανισμού αναγνώρισης νορμών .....	32
5.2 Candidate Norm Inference component .....	33

5.2.1	Event Episodes (EE) .....	33
5.2.2	Pruning Event Episodes List (PEEL).....	35
5.2.3	Υποψήφειες Νόρμες (Candidate Norms) .....	36
5.3	Επιβεβαίωση Νόρμας .....	38
6	Νόρμες Υποχρέωσης ( Obligation Norms ) .....	39
6.1	Το παράδειγμα του εστιατορίου .....	39
6.2	Αναγνώριση Νόρμας Υποχρέωσης ( Obligation Norm identification - ONI).....	40
6.2.1	Παράμετροι αλγορίθμου .....	40
6.2.2	Ο αλγόριθμος .....	40
6.2.3	Event Episodes .....	41
6.2.4	Special Event Episode List ( SEEL ) και την Normal Event Episode List ( NEEL ) .....	42
6.2.5	Norm – Related Event Set ( NREES ).....	42
6.2.6	Candidate Obligation Norm Set ( CONS ) .....	44
6.2.6	Επιβεβαίωση Νόρμας (Norm Verification) .....	45
7	Νόρμες Υπό προϋποθέσεις (Conditional Norms) .....	46
7.1	Αναγνωρίζοντας νόρμες υπό προϋποθέσεις (Conditional Norms) .....	46
8	Περιγραφή Υλοποίησης Αλγορίθμου .....	48
8.1	Υλοποίηση Αλγορίθμου .....	48
9	Συμπεράσματα & Επεκτάσεις.....	58
9.1	Συμπεράσματα & Επεκτάσεις της αρχιτεκτονικής για την αναγνώριση νορμών.....	58
9.2	Αρχιτεκτονικές Αναγνώρισης Νορμών .....	59
	Βιβλιογραφία – Πηγές .....	60
	Παράρτημα Α.....	61





## Πίνακας Περιεχομένων Εικόνων

Figure 1 .....	13
Figure 2 Αρχιτεκτονική της διαδικασίας αναγνώρισης νορμών ενός πράκτορα [5] .....	24
Figure 3 Γενική εικόνα μηχανισμού αναγνώρισης νορμών.....	30
Figure 4 Γεγονότα που παρατηρεί ένας πράκτορας .....	33
Figure 5 Γενική εικόνα μηχανισμού αναγνώρισης νορμών.....	50
Figure 6 Το Unique Event Set ( UES ) του παραδείγματος.....	54
Figure 7 OP των γεγονότων του EE και το τελικό RES .....	55
Figure 8 PEEL.....	56
Figure 9 Candidate Norm List.....	57





# 1 Εισαγωγή

## 1.1 Εισαγωγή στην αναγνώριση νομών σε πολυπρακτορικά συστήματα

Μέσα σε ένα πολυπρακτορικό περιβάλλον, οι νόρμες θεωρούνται ως μέσα ελέγχου της συμπεριφοράς των πρακτόρων μη επηρεάζοντας ταυτόχρονα και την αυτονομία τους. Οι νόρμες προσδιορίζονται ως υποχρεώσεις, απαγορεύσεις ή και άδειες κάτω από ορισμένες συνθήκες. Δρουν ως ένας ελαστικός περιορισμός που καθορίζει τη συμπεριφορά των πρακτόρων. Παρόλα αυτά οι πράκτορας μπορούν να παραβιάσουν αυτούς τους περιορισμούς για μια ποικιλία λόγων, μέσα στους οποίους περιλαμβάνεται η προσήλωσή τους σε ένα σημαντικό στόχο, η κακή συμπεριφορά ή ακόμα και η κακεντρεχής συμπεριφορά ενός πράκτορα. Όταν ένας πράκτορας παραβιάζει μία νόρμα τιμωρείται. Το να μελετηθεί το πώς μία νόρμα χρησιμοποιείται για τον περιορισμό της συμπεριφοράς ενός πράκτορα αποτελεί ένα σημαντικό θέμα προς μελέτη το οποίο κυρίως επικεντρώνεται στο πως επιδρούν οι νόρμες πάνω στον πράκτορα. Λιγότερο ενδιαφέρον μέχρι στιγμής έχει συγκεντρώσει ένα εξίσου σημαντικό κομμάτι, αυτό της αναγνώρισης των νομών μιας κοινωνίας από έναν πράκτορα που δεν τις γνωρίζει εξ αρχής. Το να θεωρείται ότι ο πράκτορας γνωρίζει ήδη τις νόρμες του περιβάλλοντος μέσα στο οποίο εισέρχεται αποτελεί γεγονός μη ρεαλιστικό ειδικά σε ανοιχτά πολυπρακτορικά περιβάλλοντα, περιβάλλοντα τα οποία προσομοιάζουν με κοινωνία ανθρώπων και στα οποία οι πράκτορες μπορούν να εισέρχονται και να εξέρχονται οποιαδήποτε χρονική στιγμή. Άλλες περιπτώσεις κατά τις οποίες χρειάζεται ο πράκτορας να έχει τη δυνατότητα να αναγνωρίζει μια νόρμα είναι αυτή των κακόβουλων πρακτόρων, δηλαδή πρακτόρων η συμπεριφορά των οποίων μπορεί να οδηγήσει στο να αποπροσανατολίσει τον πράκτορα που δεν έχει γνώση των νομών του περιβάλλοντος. Π.χ. μία λανθασμένη πληροφορία σχετικά με μία νόρμα του περιβάλλοντος μπορεί να έχει ως συνέπεια μη επιτρεπτή συμπεριφορά και άρα τιμωρία του πράκτορα που γίνεται δέκτης της λανθασμένης νόρμας. Επίσης, σε περιπτώσεις που υπάρχει έλλειψη κοινού πρωτοκόλλου επικοινωνίας μεταξύ των πρακτόρων, η δυνατότητα αναγνώρισης νομών που μπορεί να έχει ο πράκτορας οδηγεί στο να παρακαμφτεί αυτή η αδιέξοδη κατάσταση.

Στη συγκεκριμένη εργασία, ο πράκτορας παρατηρεί τους πράκτορες του περιβάλλοντός του, το πώς εκείνοι δρουν αλλά και την τιμωρία τους σε περίπτωση που παραβιάσουν μια νόρμα. Μαθαίνοντας και στοιχειοθετώντας τις ενέργειες που οδηγούν στην τιμωρία τους οδηγούνται στην αναγνώριση της νόρμας μέσω αλγορίθμων.



# 2 Πράκτορες και Νόρμες

## 2.1 Πράκτορες σε πολυπρακτορικά συστήματα (Multi - agent systems, MAS)

### 2.1.1 Πράκτορες

Τι είναι ένας πράκτορας;

Η έννοια «πράκτορας» έχει βρει κατά καιρούς πολλές ερμηνείες. Η αρχική δόθηκε από τον John McCarthy στα μέσα της δεκαετίας του πενήντα. Σύμφωνα με αυτή ένας πράκτορας είναι *η επιστήμη και η μηχανική του να φτιάξεις μία έξυπνη μηχανή* ( «*the science and engineering of making intelligent machines*», 1955, "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence"). Σύμφωνα με την οπτική αυτή, ο πράκτορας είναι ένα σύστημα που διεξάγει κάποιες διαδικασίες ώστε να πετύχει ένα στόχο και μπορεί να ρωτήσει και να πάρει συμβουλές, όποτε αυτό είναι απαραίτητο, είτε από τους ανθρώπους είτε από άλλους πράκτορες.

Σήμερα, ένας ευρέως αποδεκτός ορισμός είναι αυτός που έχει δοθεί από τον Wooldridge (2002, Εισαγωγή στα Πολυπρακτορικά συστήματα), σύμφωνα με τον οποίο **ένας πράκτορας είναι ένα υπολογιστικό σύστημα το οποίο βρίσκεται σε κάποιο περιβάλλον, και έχει τη δυνατότητα αυτόνομης δράσης μέσα σε αυτό το περιβάλλον ώστε να πετύχει τους στόχους οι οποίοι του έχουν ανατεθεί** ( "*an agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its delegated objectives*" ). Η παρακάτω εικόνα δείχνει μια αφαιρετική άποψη ενός πράκτορα ο οποίος έχει τη δυνατότητα να εκτελεί ένα ρεπερτόριο ενεργειών που μπορεί να τροποποιήσει το περιβάλλον στο οποίο βρίσκεται. Σε γενικές γραμμές πρέπει να σημειωθεί ότι δεν μπορούν να εκτελεστούν όλες οι ενέργειες σε όλες τις περιστάσεις καθώς επίσης ότι η εκτέλεση μιας ενέργειας πράκτορα δεν συνεπάγεται πάντα το ίδιο αποτέλεσμα στο περιβάλλον του.



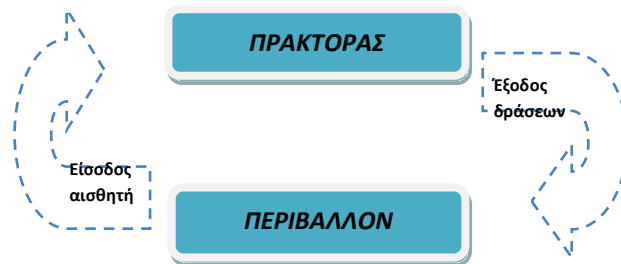


Figure 1

Πολλά είναι τα χαρακτηριστικά που έχουν προσδοθεί στους πράκτορες από τους επιστήμονες. Οι πράκτορες είναι *αυτόνομοι*, δηλαδή έχουν την ικανότητα να αποφασίζουν πότε θα ενεργήσουν χωρίς την παρεμβατικότητα ενός ανθρώπου. Είναι *αντιδραστικοί*, έχουν δηλαδή την ικανότητα να αντιλαμβάνονται το περιβάλλον τους και να ενεργούν αντίστοιχα. Δρουν επίσης *προληπτικά* παίρνοντας πρωτοβουλίες που θα τους οδηγήσουν στην επίτευξη ενός στόχου. Έχουν την ικανότητα να συνεργάζονται, να διαπραγματεύονται, να μαθαίνουν και να προσαρμόζονται είτε ακόμα και να μετακινούνται σε άλλα περιβάλλοντα (Wooldridge, 2002).

Συνοψίζοντας, οι πράκτορες είναι υπολογιστικά στοιχεία με δύο σημαντικές ικανότητες. Είναι ικανοί ως κάποιο βαθμό για αυτόνομη δράση, μπορούν δηλαδή να αποφασίζουν μόνοι τους για τη δράση τους ώστε να ικανοποιήσουν τους στόχους τους και μπορούν να αλληλεπιδρούν με άλλους πράκτορες που βρίσκονται στο περιβάλλον τους όπως ανάλογα αλληλεπιδρούν οι άνθρωποι στην καθημερινότητά τους.

### 2.1.2 Περιβάλλοντα

Τα περιβάλλοντα, μέσα στα οποία ενεργούν οι πράκτορες, σύμφωνα με τους Russell και Norvig μπορούν να ταξινομηθούν ως ακολούθως [6].

- i) **Προσιτό ή απρόσιτο.** Ένα περιβάλλον είναι προσιτό όταν ο πράκτορας μπορεί να αποκτήσει πλήρεις, ακριβείς, και ενημερωμένες πληροφορίες για την κατάσταση του περιβάλλοντος. Με αυτή την έννοια, τα περισσότερα περιβάλλοντα του πραγματικού κόσμου (συμπεριλαμβανομένων, για παράδειγμα, του καθημερινού φυσικού κόσμου και του Διαδικτύου) δεν είναι προσιτά.
- ii) **Αιτιοκρατικό ή μη αιτιοκρατικό.** Ένα περιβάλλον είναι αιτιοκρατικό όταν οποιαδήποτε ενέργεια σε αυτό έχει ένα μοναδικό εγγυημένο αποτέλεσμα – δεν υπάρχει αβεβαιότητα για την κατάσταση που θα προκύψει μετά την εκτέλεση μιας ενέργειας.
- iii) **Στατικό ή δυναμικό.** Ένα περιβάλλον είναι στατικό όταν μπορεί να υποθεθεί ότι παραμένει αμετάβλητο, εκτός και αν ο πράκτορας εκτελέσει κάποια ενέργεια. Αντίθετα, ένα δυναμικό

περιβάλλον έχει και άλλες διεργασίες σε λειτουργία και, επομένως, αλλάζει με τρόπους που δεν ελέγχει ο πράκτορας. Ο πραγματικός κόσμος είναι πολύ δυναμικό περιβάλλον, όπως επίσης και το Διαδίκτυο.

- iv) **Διακριτό ή συνεχές.** Ένα περιβάλλον είναι διακριτό αν υπάρχει σταθερός, πεπερασμένος αριθμός ενεργειών και αντιλήψεων σε αυτό. (Wooldridge, 2002)

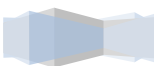
Το πώς δρα και αλληλεπιδρά ένας πράκτορας με το περιβάλλον του έχει να κάνει συνεπώς όχι μόνο με τον ίδιο τον πράκτορα αλλά και με τη φύση και τα χαρακτηριστικά του ίδιου του περιβάλλοντος. Όσο πιο προσιτό είναι το περιβάλλον τόσο πιο αποτελεσματικοί μπορούν να είναι οι πράκτορες μέσα σε αυτό. Όσο πιο αιτιοκρατικό είναι το περιβάλλον, τόση μεγαλύτερη είναι η επιρροή που μπορούν να έχουν οι πράκτορες πάνω σε αυτό λόγω της μειωμένης αβεβαιότητας των αποτελεσμάτων των πράξεών τους. Όταν ο πράκτορας καλείται να λειτουργήσει σε ένα δυναμικό περιβάλλον θα πρέπει πριν από κάθε του ενέργεια να συλλέξει πληροφορίες για να προσδιορίσει την κατάσταση του περιβάλλοντος του εκείνη τη χρονική στιγμή. Τέλος, είναι προφανές ότι στην περίπτωση που ο πράκτορας δρα μέσα σε ένα διακριτό περιβάλλον η δυνατότητα να απαριθμήσει τις διακριτές καταστάσεις που μπορεί να υπάρξουν συνεπάγεται και σε μία αυξημένη ακρίβεια των αποτελεσμάτων των ενεργειών του.

Βάσει των παραπάνω διαχωρισμών, ορίζονται τα **ανοιχτά περιβάλλοντα**, περιβάλλοντα τα οποία είναι απρόσιτα, μη αιτιοκρατικά, δυναμικά και συνεχή (Hewitt, 1986).

### 2.1.3 Πολυπρακτορικά Συστήματα (Multi-agent Systems - MAS)

Ένα πολυπρακτορικό σύστημα είναι ένα καταμεμημένο σύστημα το οποίο αποτελείται από διαφορετικού τύπου πράκτορες με ποικίλες δυνατότητες οι οποίοι αλληλεπιδρούν μεταξύ τους στέλνοντας ή/και λαμβάνοντας μηνύματα (Wooldridge, 2009). Ο Wooldridge επίσης αναφέρει ότι οι πράκτορες σε ένα πολυπρακτορικό σύστημα μπορούν να αντιπροσωπεύουν ή να δρουν στη θέση χρηστών με διαφορετικούς στόχους και κίνητρα. Επιπλέον, οι πράκτορες συνεργάζονται και διαπραγματεύονται μεταξύ τους ώστε να αλληλεπιδράσουν επιτυχώς.

Ένας πράκτορας σε ένα πολυπρακτορικό σύστημα δρα σε ένα περιβάλλον για το οποίο έχει *μερική εικόνα*, δηλαδή *ένα μόνο μέρος της πληροφορίας* είναι διαθέσιμο. Επιπλέον, οι κινήσεις του είναι *μη ντετερμινιστικές*, δηλαδή το αποτέλεσμά τους είναι πρακτικά μη προβλέψιμο. Το περιβάλλον είναι *δυναμικό* και μπορεί να έχει απεριόριστο αριθμό καταστάσεων. Βάσει και της ερμηνείας του Hewitt [16] που δόθηκε παραπάνω, μπορούμε να συμπεράνουμε ότι ο πράκτορας σε ένα πολυπρακτορικό περιβάλλον δρα μέσα σε ένα *ανοιχτό περιβάλλον*.



## Διαχωρισμός πολυπρακτορικού περιβάλλοντος σε επίπεδα

Ένα πολυπρακτορικό σύστημα μπορεί να θεωρηθεί ότι αποτελείται από τρία επίπεδα [14],

- i. το επίπεδο των *πρακτόρων*,
- ii. το επίπεδο της *αλληλεπίδρασης*, και το
- iii. επίπεδο της *κοινωνίας*.

Οι αυτόνομοι πράκτορες που βρίσκονται σε ένα περιβάλλον έχουν συγκεκριμένες δυνατότητες. Μπορούν να παρατηρούν το περιβάλλον, να μαθαίνουν βασισμένοι σε ό,τι παρατηρούν και να σχεδιάζουν τη δράση τους. Ένας πράκτορας μπορεί να μην δύναται να ανταποκριθεί από μόνος του σε ένα πολύπλοκο σενάριο. Σε αυτή την περίπτωση, χρειάζεται η συνεργασία με τους υπόλοιπους πράκτορες του περιβάλλοντος του ώστε το σενάριο αυτό να έρθει εις πέρας. Άρα οι πράκτορες πρέπει να **αλληλεπιδράσουν**. Οι πράκτορες αλληλεπιδρούν με άλλους πράκτορες μέσω της **επικοινωνίας**, στέλνοντας και λαμβάνοντας πληροφορία. Η επικοινωνία αυτή γίνεται βάσει πρωτοκόλλων τα οποία μπορούν να διαχωριστούν σε *πρωτόκολλα επικοινωνίας* και *πρωτόκολλα διαπραγμάτευσης*. Οι αλληλεπιδράσεις μεταξύ των πρακτόρων λαμβάνουν μέρος πάνω σε κοινωνικές και οργανωτικές δομές.

Άλλη μία κεντρική ιδέα στα πολυπρακτορικά συστήματα είναι το γεγονός ότι ο κάθε πράκτορας, για να καταλήξει στην τελική του απόφαση, λαμβάνει υπόψη του το **κοινωνικό πλαίσιο** μέσα στο οποίο δρα. Οι πράκτορες επηρεάζονται από την εξάρτησή τους από άλλους πράκτορες, από τον ρόλο τους μέσα στην κοινωνία, τις δεσμεύσεις τους έναντι στους άλλους πράκτορες μέσα στην κοινωνία, τους κοινωνικούς κανόνες και τους κανόνες που διέπουν τη συμπεριφορά τους. Για παράδειγμα, ένας πράκτορας μπορεί να εξαρτάται από τη δράση ενός άλλου πράκτορα για την ικανοποίηση των ατομικών του στόχων.

Γενικεύοντας, μπορούμε να πούμε ότι τα πολυπρακτορικά συστήματα είναι άρρηκτα συνδεδεμένα και επηρεασμένα τόσο από την καθημερινή ζωή των ανθρώπων όσο και από άλλες επιστήμες. Τεχνητή νοημοσύνη, καταναμημένα συστήματα, οικονομικά, φιλοσοφία και κοινωνικές επιστήμες αποτελούν μερικά παραδείγματα. Ταυτόχρονα, αντίστροφα τα πολυπρακτορικά συστήματα έχουν συμβάλει στην εξέλιξη των παραπάνω επιστημών. Στην παρακάτω υποενότητα δίνεται μία άλλη προσέγγιση στην έννοια της επικοινωνίας των πρακτόρων.

### 2.1.4 Επικοινωνία Πρακτόρων – Πρωτόκολλα

Οι πράκτορες που δρουν και αλληλεπιδρούν σε μία κοινωνία μπορεί να έχουν σχεδιαστεί από διαφορετικούς ανθρώπους. Το να χρησιμοποιούν διαφορετικές αναπαραστάσεις της γνώσης τους θα έχει σαν αποτέλεσμα να μην γίνει εφικτή η μεταξύ τους επικοινωνία. Οι πράκτορες ενός περιβάλλοντος πρέπει να κάνουν ορισμένες δεσμεύσεις για να καταστεί δυνατή η αποτελεσματική επικοινωνία μεταξύ τους. Οι παράγοντες που πρέπει τουλάχιστον να δεσμευτούν είναι οι ακόλουθοι:

- Ένα κοινό σύνολο αρχικών επικοινωνιακών πράξεων που προσδιορίζουν την πρόθεση του ομιλητή. Το συντακτικό και η σημασιολογία των επικοινωνιακών πράξεων πρέπει να ορίζονται εξαρχής της επικοινωνίας βάσει μιας συγκεκριμένης γλώσσας. Η γλώσσα συνήθως περιλαμβάνει *ισχυρισμούς*, όπως το *tell* και το *inform*, και *ερωτήσεις* όπως το *ask* και το *query*.
- Ένα κοινά δομημένο γλωσσάρι, που χρησιμοποιείται για να αναπαραστήσει τη γνώση που μεταδίδεται. Αυτό αναφέρεται συχνά ως μία κοινή οντολογία [15].

### *Η έννοια της ειλικρίνειας στην επικοινωνία των πρακτόρων*

Λέει ένας πράκτορας πάντα την αλήθεια στους άλλους πράκτορες του περιβάλλοντός του; Αυτό δεν είναι πάντα εφικτό να απαντηθεί με σιγουριά. Ένας πράκτορας που συμμετέχει σε μια κοινωνία μπορεί να επικοινωνεί λειτουργικά με τους άλλους πράκτορες μόνο αν είναι συνεργατικός και αν δίνει σωστές πληροφορίες. Παρόλα αυτά, κανένας κανόνας επικοινωνίας δε δεσμεύει τον πράκτορα να λέει την αλήθεια. Επίσης μία τέτοια δέσμευση ίσως απομάκρυνε μια κοινωνία πρακτόρων από το να προσομοιάζει μιας πραγματικής κοινωνίας ανθρώπων.

Ένας πράκτορας θεωρείται ειλικρινής αν και μόνο αν θέλει να πληροφορεί σωστά έναν άλλο πράκτορα, γι' αυτό και το κάνει αυτό μόνο όταν είναι σίγουρος για αυτά που πρόκειται να πει. Αυτό θα μπορούσε να θεωρηθεί ως ιδανική κατάσταση για ένα πολυπρακτορικό περιβάλλον. Από την άλλη θα μπορούσε ακόμα και να υπάρξει νόρμα που να απαιτεί την ειλικρινή πληροφόρηση μεταξύ των πρακτόρων.

Στη συγκεκριμένη εργασία, και στα πλαίσια της αναγνώρισης των νορμών, η έννοια της ειλικρίνειας είναι ζωτικής σημασίας γιατί, όπως σχολιάζεται και σε παρακάτω κεφάλαιο, σε αυτή στηρίζεται η επιβεβαίωση μιας νόρμας που έχει συμπεράνει ένας πράκτορας.

### *Η έννοια της δέσμευσης στην επικοινωνία των πρακτόρων*

Η έννοια της δέσμευσης σε ένα πολυπρακτορικό περιβάλλον έχει γενικά να κάνει με μία ενέργεια ή ένα στόχο που ο πράκτορας πρέπει να ικανοποιήσει, μία πρόθεση. Μπορούμε να ορίσουμε την πρόθεση ενός ατόμου ως επίμονο στόχο, μία κατάσταση κατά την οποία ο πράκτορας θα συνεχίσει να έχει ως κίνητρο μέχρι είτε την επίτευξη του στόχου του, είτε μέχρι να κατανοήσει ότι ο στόχος του δεν μπορεί να επιτευχθεί. Οι κοινωνικές δεσμεύσεις λειτουργούν βασισμένες στις νόρμες της κοινωνίας. Οι κοινωνικές δεσμεύσεις επιπλέον δημιουργούν δικαιώματα στα άτομα που δρουν μέσα σε μια κοινωνία. Όπως θα δούμε και παρακάτω, ειδικά σε ό,τι αφορά καταστάσεις κατά τις οποίες πράκτορες παραβαίνουν τις νόρμες μιας κοινωνίας, ο πράκτορας έχει το δικαίωμα να διαμαρτυρηθεί ή και ακόμα να τιμωρήσει τον πράκτορα που δεν ικανοποίησε τη δέσμευση.





## 2.2 Τι είναι η Νόρμα.

Οι νόρμες μπορεί να ερμηνευτούν διαφορετικά ανάλογα με την οπτική γωνία από την οποία προσπαθεί ο καθένας να προσδώσει έναν ορισμό. Βλέποντας τη νόρμα από την οπτική της Τεχνητής Νοημοσύνης και πιο συγκεκριμένα μέσα στο πλαίσιο ενός πολυπρακτορικού περιβάλλοντος, μπορούμε να ορίσουμε τη νόρμα ως ένα κανόνα ο οποίος οδηγεί τους πράκτορες στο να έχουν μία κοινή συμπεριφορά διευκολύνοντάς τους έτσι στη λήψη αποφάσεων, στο συντονισμό και την οργάνωσή τους.

Η ανθρώπινη κοινωνία ακολουθεί νόρμες. Οι διαφορετικές κουλτούρες, οι διαφορετικές χώρες διέπονται από διαφορετικές νόρμες και αποτελούν αντικείμενο έρευνας για πολλές επιστήμες όπως η Κοινωνιολογία, τα Οικονομικά, η Ψυχολογία όπως και την επιστήμη των Υπολογιστών.

### 2.2.1 Νόρμες στις ανθρώπινες κοινωνίες

Η νόρμα που διέπει μια κοινωνία είναι το πρωταρχικό μέσο κοινωνικής ένταξης στη σύγχρονη κοινωνία, και είναι η εξουσία που επιβάλλει υπακοή από τους υπηκόους της κοινωνίας [16]. Ως δύναμη από μόνη της δεν μπορεί να χορηγήσει τη νομιμότητά της στη σύγχρονη κοινωνία, η νόρμα αντλεί την ισχύ της από τη συναίνεση των κυβερνωμένων.

Οι οργανωμένες από νόρμες ενέργειες προσδιορίζονται ως ένα από τα τέσσερα υποδείγματα πράξεων στην ανθρώπινη συμπεριφορά. Μία νόρμα σημαίνει να εκπληρώνεις μία γενικευμένη προσδοκία συμπεριφοράς η οποία είναι κοινώς αποδεκτή. Μια προσδοκία συμπεριφοράς γενικεύεται αν κάθε μέλος της κοινωνίας περιμένει πως όλα τα υπόλοιπα μέλη θα συμπεριφερθούν με ένα συγκεκριμένο τρόπο σε μια δεδομένη κατάσταση.

Αυτό που είναι κοινό σε παρόμοιες ερμηνείες είναι ότι πάντα ο «πράκτορας» ή το άτομο αναμένεται να συμπεριφερθεί με *συγκεκριμένο τρόπο* και η «ορθή συμπεριφορά» υπαγορεύεται από την κοινωνία. Δεν υπάρχει παρόλα αυτά σε αυτές του είδους τις ερμηνείες αναφορά στην παραβίαση κανόνα βάσει του κοινωνικού ελέγχου. Παραδείγματος χάριν, η παραπάνω ερμηνεία [16] δεν αναφέρεται στην παραβίαση των νορμών.

Οι νόρμες θα μπορούσαν να διαχωριστούν, σε μια πρώτη ανάγνωση, σε δύο κατηγορίες. *Κοινωνικές νόρμες* και *προσωπικές νόρμες*. Οι νόρμες της κοινωνίας προσδιορίζουν τη συμπεριφορά της ομάδας και συνδέονται και με κυρώσεις για τον πράκτορα που τις παραβαίνει ή τιμωρία. Οι προσωπικές νόρμες βασίζονται στις προσωπικές πεποιθήσεις του ατόμου. Οι προσωπικές νόρμες είναι εν δυνάμει κοινωνικές νόρμες. Οι προσωπικές νόρμες θα μπορούσαν να αποτελέσουν κοινωνικές νόρμες αν παρακολουθούνταν από άλλους πράκτορες και αν το να μην ακολουθεί κάποιος τη νόρμα συνδεόταν με κυρώσεις.

Οι *κοινωνικές νόρμες* ταξινομούνται περαιτέρω σε *r-norms* (rule norms, νόρμες κανόνα) και σε *s-norms* (social norms, κοινωνικές νόρμες). Οι *προσωπικές νόρμες* ταξινομούνται περαιτέρω σε *m-norms* (moral norms, νόρμες ηθικής) και σε *p-norms* (prudential norms, προληπτικές νόρμες). Οι *νόρμες κανόνα* επιβάλλονται στις συμφωνίες μεταξύ των μελών μιας κοινωνίας από μία «εξουσιαστική αρχή». Οι *κοινωνικές νόρμες* εφαρμόζονται σε μεγάλες ομάδες, όπως π.χ. μία ολόκληρη κοινωνία, και βασίζονται στο ποια συμπεριφορά θα πρέπει να έχουν τα μέλη της ομάδας στην καθημερινότητά τους (π.χ. δεν πρέπει να πετάμε τα σκουπίδια μας εκτός κάδων). Τα μέλη της κοινωνίας αναμένουν ότι η κοινωνική νόρμα ακολουθείται και από τα υπόλοιπα μέλη της κοινωνίας. Οι *νόρμες ηθικής* επικαλούνται την συνείδηση του πράκτορα ή του ατόμου. Οι *προληπτικές νόρμες* βασίζονται στη λογική. Όταν ένα μέλος της κοινωνίας παραβαίνει κάποια από τις κοινωνικές νόρμες τιμωρείται ή, σε μερικές περιπτώσεις, εξοστρακίζεται από την κοινωνία.

Όσον αφορά τα πολυπρακτορικά συστήματα και την παρούσα εργασία θα ασχοληθούμε κυρίως με τις κοινωνικές νόρμες. Θα μπορούσαμε να σημειώσουμε ότι η ιδέα της *κοινωνικής νόρμας* αποτελείται από τα τρία παρακάτω στοιχεία.

- **Προσδοκία νόρμας για τακτικές συμπεριφορές:** Η κοινωνία διέπεται από μία γενικευμένη συμφωνία βάσει της οποίας αναμένεται μία συγκεκριμένη συμπεριφορά από τον πράκτορα κάτω από συγκεκριμένες συνθήκες.
- **Μηχανισμός επιβολής νόρμας :** Όταν ένας πράκτορας δεν ακολουθεί τη νόρμα τότε υπόκειται σε κυρώσεις/τιμωρία. Οι κυρώσεις μπορεί να περιλαμβάνουν χρηματική ή φυσική τιμωρία στον πραγματικό κόσμο και μπορεί να προκαλέσουν συναισθήματα στον πράκτορα (ντροπή, ενοχή κλπ.) ή άμεση απώλεια των παροχών. Επίσης, οι υπόλοιποι πράκτορες μπορεί να δείξουν απροθυμία να έρθουν σε επικοινωνία με ένα πράκτορα που έχει δεχθεί κυρώσεις επειδή είχε παραβεί τη νόρμα της κοινωνίας. Οι πράκτορες που τηρούν τη νόρμα επιβραβεύονται.
- **Μηχανισμός διάδοσης νόρμας:** Μία νόρμα μπορεί να διαδοθεί είτε μέσω ενός ηγέτη προς την κοινωνία είτε ακόμα και μέσω πολιτισμικών επιρροών. Για έναν εξωτερικό παρατηρητή, το γεγονός ότι οι πράκτορες μπορούν να προσδιορίσουν και να υιοθετήσουν μια νόρμα μέσω μηχανισμών μάθησης, όπως αυτόν της μίμησης, φαίνεται σαν τρόπος διάδοσης νορμών σε μια κοινωνία πρακτόρων.

### 2.2.2 Κύκλοι ζωής μιας Νόρμας

Μέχρι στιγμής δεν υπάρχει μία κοινά αποδεκτή άποψη σχετικά με το ποιος είναι ο κύκλος ζωής μιας νόρμας. Παρόλα αυτά η ζωή μιας νόρμας μπορεί να διαχωριστεί σε τρία στάδια [7]. Το πρώτο στάδιο είναι η **εμφάνιση της νόρμας**, το οποίο χαρακτηρίζεται από το να πεισθούν οι πράκτορες να ακολουθήσουν τη νόρμα. Το δεύτερο στάδιο είναι το στάδιο της **αλληλοεπικάλυψης**, το οποίο χαρακτηρίζεται από τη δυναμική της μίμησης. Κατά το στάδιο αυτό, τα άτομα που δρουν ως «αρχηγοί» σχετικά με ότι έχει να κάνει με τη νόρμα, προσπαθούν να έρθουν σε επικοινωνία και να επηρεάσουν άλλους πράκτορες ώστε να τους ακολουθήσουν. Οι πράκτορες που ακολουθούν μπορεί να

υιοθετήσουν την εκάστοτε νόρμα που τους προτείνεται. Το να ακολουθήσουν τη νόρμα μπορεί να αυξήσει τη φήμη και την αυτοεκτίμησή τους μέσα στην ομάδα. Μπορεί επίσης το να ακολουθήσουν τη νόρμα να οφείλεται σε πίεση που ενδεχομένως έχουν δεχθεί από άλλα μέλη της ομάδας τα οποία ακολουθούν ήδη τη νόρμα. Το τρίτο στάδιο είναι η **εσωτερίκευση της νόρμας**. Κατά το στάδιο αυτό, η νόρμα γίνεται ευρέως αποδεκτή από τους πράκτορες της κοινωνίας. Η νόρμα πλέον θεωρείται ως κάτι το δεδομένο για την κοινωνία. Ένα ζήτημα σχετικά με την εσωτερίκευση της νόρμας είναι ότι μετά το πέρας αυτού του σταδίου η αποδοχή της νόρμας είναι τέτοια που δύσκολα τίθεται θέμα αμφισβήτησης της νόρμας και του κατά πόσο πρέπει αυτή να ακολουθείται.

## 2.3 Πολυπρακτορικά συστήματα που διέπονται από Νόρμες (NorMAS)

### 2.3.1 Διαφορετικές όψεις των νορμών σε Πολυπρακτορικά συστήματα που διέπονται από Νόρμες (NorMAS)

Οι νόρμες, όπως αναφέρθηκε και παραπάνω, βοηθούν έτσι ώστε να διατηρηθεί η κοινωνική τάξη, να διευκολυνθεί η συνεργασία και να υπάρχει συντονισμός ενεργειών σε ένα πολυπρακτορικό σύστημα. Ορισμός ενός πολυπρακτορικού συστήματος που διέπεται από νόρμα είναι ο εξής: «Ένα πολυπρακτορικό σύστημα που διέπεται από νόρμα είναι ένα πολυπρακτορικό σύστημα οργανωμένο μέσω μηχανισμών ώστε να αντιπροσωπεύει, επικοινωνεί, διανέμει, εντοπίζει, δημιουργεί, τροποποιεί και ενισχύει τη νόρμα και επιπλέον μπορεί να εντοπίζει την παραβίαση και την εκπλήρωσή της. »

#### *Αυτόνομος Πράκτορας*

Ένα βασικό χαρακτηριστικό πράκτορα που δρα σε ένα NorMAS είναι η **αυτονομία** του σχετικά με τη νόρμα του συστήματος [17]. Ένας *αυτόνομος πράκτορας* :

- Είναι ικανός να αναγνωρίσει ότι υπάρχει νόρμα στην κοινωνία και ότι οι συμπεριφορές των υπόλοιπων δεν είναι απλά μια συνήθεια των πρακτόρων της κοινωνίας, ή ένα προσωπικό αίτημα, εντολή ή προσδοκία ενός ή περισσότερων πρακτόρων.
- Είναι σε θέση να υιοθετεί νόρμα η οποία προσκρούει στη δική του συμπεριφορά ή απόφαση
- Είναι ικανός να ακολουθεί ελεύθερα αυτή τη νόρμα
- Μπορεί ελεύθερα να παραβιάζει τη νόρμα σε περίπτωση που αυτή έρχεται σε σύγκρουση με άλλες νόρμες.

Οι αυτόνομοι πράκτορες είναι πράκτορες που επιλέγουν ποιους στόχους τους είναι θεμιτό να επιδιώξουν, βασισμένοι σε ένα δοθέν σύστημα νορμών. Ο πράκτορας έχει την αυτονομία να «παράγει» τους στόχους του και να επιλέγει ποιους από αυτούς τελικά θα επιδιώξει να φέρει εις πέρας. Εκτός

αυτού, ο πράκτορας μπορεί να κρίνει τη νομιμότητα τόσο των δικών του στόχων όσο και των υπολοίπων πρακτόρων που «ζουν» στην ίδια με αυτόν κοινωνία. Όταν υπάρχει σύγκρουση ενός στόχου με τις νόρμες της κοινωνίας, ο πράκτορας μπορεί να αλλάξει το σύστημα νομών του, δηλαδή να αλλάξει τις προτεραιότητες των στόχων του, να εγκαταλείψει έναν στόχο, να αλλάξει έναν στόχο, ή να δημιουργήσει έναν καινούριο. Οι νόρμες σε ένα τέτοιο περιβάλλον μπορούν να αντιμετωπιστούν ως περιορισμοί πάνω στη συμπεριφορά των πρακτόρων, ως στόχοι που πρέπει να επιτευχθούν, ως γενικές υποχρεώσεις, ως κοινωνικοί νόμοι, δικαιώματα, συμβάσεις, πρωτόκολλα ή δεσμεύσεις.

### 2.3.2 Επιβολή Νομών

Κατά καιρούς έχουν προταθεί από τους επιστήμονες διαφορετικοί μηχανισμοί που συμβάλλουν στο να τηρηθούν οι νόρμες σε ένα πολυπρακτορικό σύστημα. Οι μηχανισμοί αυτοί στηρίζονται εκάστοτε είτε στον έλεγχο παραβίασης νομών είτε στην «επίκληση στο συναίσθημα» του πράκτορα όταν εκείνος παραβιάζει μια νόρμα.

Οι πράκτορες ενός περιβάλλοντος ενδέχεται να αποφύγουν την επικοινωνία με ένα πράκτορα που έχει παραβιάσει μια νόρμα και συνεπώς να τον απομονώσουν. Αυτή η αντιμετώπιση αποτελεί άλλο ένα παράδειγμα μηχανισμού για την επιβολή μιας νόρμας κατά το οποίο ο «παραβάτης» εξοστρακίζεται από την κοινωνία. Ένας ακόμα τρόπος επιβολής νόρμας είναι η δημιουργία πράκτορα «επιτηρητή», ο οποίος έχει ως στόχο να ελέγχει αν και κατά πόσο οι υπόλοιποι πράκτορες του περιβάλλοντος συμμορφώνονται με τις νόρμες της κοινωνίας. Τέλος, έναν ακόμα μηχανισμό επιβολής νόρμας θα μπορούσε να αποτελέσει και η επιβράβευση των πρακτόρων που τηρούν τη νόρμα.



# 3 Μηχανισμός για την εμφάνιση Νορμών ( Role Model )

Μέσα σε μια κοινωνία, ένα άτομο μπορεί να επηρεάσει τις πεποιθήσεις των υπολοίπων. Οι κανόνες, οι συνήθειες, οι πεποιθήσεις μπορούν να διαδοθούν από τους γονείς προς τα παιδιά, από ένα αρχηγό προς τους ακολούθους του είτε ακόμα και με αλληλεπίδραση μεταξύ «ομότιμων» ατόμων.

Η *δομή μιας κοινωνίας* είναι ένα στοιχείο που εξαρτάται αποκλειστικά από τη μορφή της αλληλεπίδρασης μεταξύ των ατόμων της κοινωνίας. Ο τρόπος που επηρεάζεται μια ομάδα ατόμων καταδεικνύει και το πώς διαδίδεται η πληροφορία μέσα στην ομάδα αυτή. Το παραπάνω σχήμα βρίσκει εφαρμογή και στις κοινωνίες των πρακτόρων. Ως εκ τούτου, ένα πολύ σημαντικό κομμάτι έρευνας για τα πολυπρακτορικά συστήματα αποτελεί το πώς «εμφανίζεται» μια νόρμα και το πώς διαδίδεται αυτή μέσα σε μία κοινωνία πρακτόρων.

Στη συνέχεια αυτού του κεφαλαίου παρατίθεται ο μηχανισμός για την *εμφάνιση* και την *διάδοση* μιας νόρμας (norm emergence & norm spreading) βασισμένος στους ρόλους-πρότυπα (role model ) που αναλαμβάνουν οι πράκτορες.

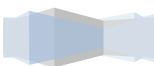
## 3.1 Μηχανισμός Role Model

Βασισμένοι στο τι μπορούμε να αποκομίσουμε παρατηρώντας την πραγματική κοινωνία των ανθρώπων, θα μπορούσαμε να πούμε ότι ένα πρότυπο για μια κοινωνία βρίσκει πολλούς μιμητές. Συνεπώς θα μπορούσαμε να θεωρήσουμε ως άτομα – πρότυπα εκείνα που πολλά άλλα μέλη μιας κοινωνίας θα ήθελαν να ακολουθήσουν και να μιμηθούν. Αντίστοιχα με την ανθρώπινη κοινωνία, σε ένα πολυπρακτορικό περιβάλλον, το ρόλο – πρότυπο καλείται να παίξει είτε ο πράκτορας που έχει τη μεγαλύτερη επιρροή έναντι των άλλων είτε αυτός που έχει τη δυνατότητα να συμβουλευτεί τους άλλους πράκτορες.

Ένας πράκτορας αναζητά το πρότυπό του στη γειτονιά του, δηλαδή ανάμεσα σε πράκτορες που βρίσκονται κοντά σε αυτόν και έχουν άμεση σύνδεση με αυτόν. Μετά τον εντοπισμό του μπορεί να ζητήσει πληροφορίες ή συμβουλές από αυτόν. Ουσιαστικά δηλαδή ο πράκτορας-πρότυπο μεταφέρει μία νόρμα προς τον πράκτορα που τον συμβουλευέται. Ο πράκτορας που λαμβάνει αυτές τις

πληροφορίες έχει επίσης την επιλογή του να υιοθετήσει αυτές τις πληροφορίες ή όχι, γεγονός που καταδεικνύει την *αυτονομία* των πρακτόρων.

Οι πράκτορες που εξετάζουμε, όπως αναφέρεται και παραπάνω, είναι **αυτόνομοι**. Όπως ο πράκτορας που λαμβάνει μια πληροφορία-νόρμα έχει την επιλογή να τη δεχτεί ή όχι, έτσι και ο πράκτορας που ζητείται μία πληροφορία έχει την επιλογή να μην τη δώσει. Σε αυτήν την περίπτωση, ο πράκτορας που ζητάει την πληροφορία έχει τη δυνατότητα να απευθυνθεί σε άλλον πράκτορα-πρότυπο στο εγγύς του περιβάλλον. Η επιλογή του επόμενου πράκτορα – προτύπου γίνεται βάσει της επίδοσης του μέσα στο περιβάλλον.



# 4 Αρχιτεκτονική για την αναγνώριση νορμών

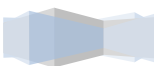
## 4.1 Η διαδικασία αναγνώρισης της νόρμας

Σε αυτή την ενότητα παρατίθεται μια επισκόπηση του μηχανισμού ο οποίος προτείνεται από τους Bastin Tony Roy Savarimuthu, Stephen Cranefield, Maryam A. Purvis και Martin K. Purvis στην επιστημονική εργασία «Norm Identification in Multi – agent Societies» (University of Otago, Department of Information Science), με τη βοήθεια του οποίου ένας πράκτορας συμπεραίνει μία νόρμα του περιβάλλοντος στο οποίο εκείνη τη στιγμή βρίσκεται .

Η θεωρία της κοινωνικής μάθησης υποθέτει ότι το άτομο μπορεί να μάθει μία νέα συμπεριφορά μέσα από την παρατήρηση της τιμωρίας και της επικρότησης. Έτσι, παρόμοια με μία κοινωνία ανθρώπων, η παρατήρηση των πράξεων ενός πράκτορα μέσα από τη διαδικασία της παρακολούθησης και μάθησης μπορεί να χρησιμοποιηθεί για να εξακριβωθούν οι νόρμες μέσα στην κοινωνία των πρακτόρων.

Στο παρακάτω διάγραμμα ροής (Εικόνα 4.1) παρουσιάζεται η διαδικασία για την αναγνώριση νορμών.

- Οι κύκλοι αναπαριστούν τα στοιχεία αποθήκευσης της πληροφορίας.
- Τα παραλληλόγραμμα με τις στρογγυλεμένες γωνίες αναπαριστούν τα στοιχεία επεξεργασίας της πληροφορίας.
- Οι ρόμβοι αναπαριστούν τα στοιχεία απόφασης.
- Οι γραμμές αναπαριστούν τη ροή της πληροφορίας μεταξύ των στοιχείων.



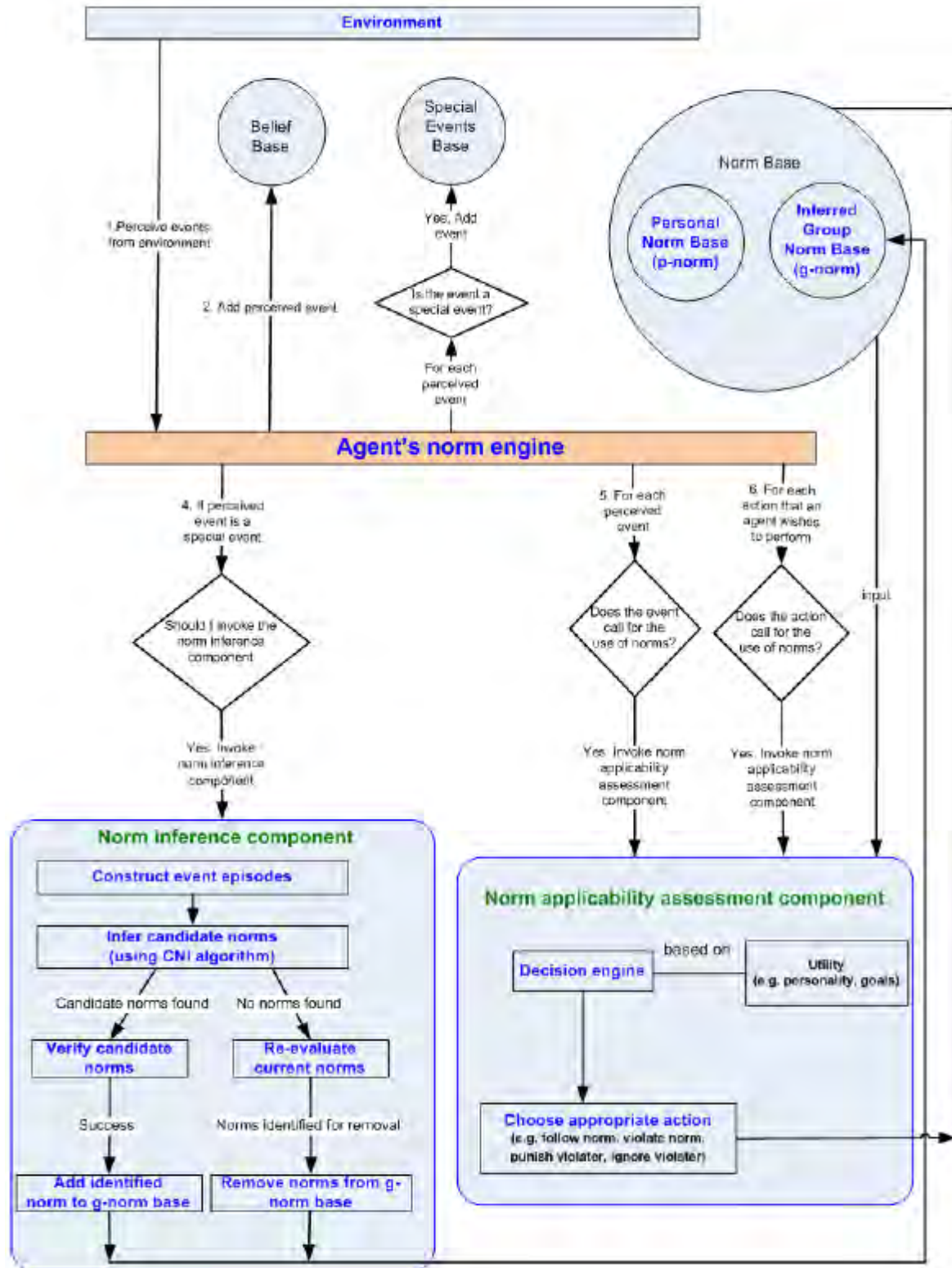


Figure 2 Αρχιτεκτονική της διαδικασίας αναγνώρισης νομών ενός πράκτορα [5]





Ο πράκτορας που εφαρμόζει αυτή την αρχιτεκτονική ακολουθεί τα παρακάτω στάδια:

- 1 Ο πράκτορας αντιλαμβάνεται τα γεγονότα που παρατηρεί στο περιβάλλον του.
- 2 Όταν ο πράκτορας αντιλαμβάνεται ένα γεγονός το αποθηκεύει στη *βάση δεδομένων των πεποιθήσεών (belief base)* του. Η βάση δεδομένων πεποιθήσεων περιέχει τις πεποιθήσεις του πράκτορα. Ο πράκτορας έχει πλήρη έλεγχο της βάσης αυτής και μπορεί να προσθέτει, να τροποποιεί και να διαγράφει πεποιθήσεις. Τα γεγονότα που παρατηρεί ένας πράκτορας μπορούν να διαχωριστούν σε δύο κατηγορίες :
  - i. **Regular events:** Γεγονότα που θεωρούνται κανονικά και αναμενόμενα από ένα πράκτορα
  - ii. **Signaling events (special events):** Γεγονότα που οι πράκτορες θεωρούν είτε ενθαρρυντικά είτε αποθαρρυντικά, προκαλούν δηλαδή είτε *ανταμοιβή (rewards)* είτε *κυρώσεις (sanctions)*, και είναι απόρροια πράξης που είχε προηγηθεί.

Ένα παράδειγμα regular event είναι αυτό ενός πράκτορα να μετακινείται μέσα στο πάρκο ή να κάθεται σε ένα παγκάκι. Αντίστοιχα παράδειγμα ενός signaling event είναι αυτό ενός πράκτορα ο οποίος επικρίνει έναν πράκτορα που προηγουμένως έχει ρυπάνει το πάρκο. Υποθέτουμε ότι οι πράκτορες έχουν τη δυνατότητα να ξεχωρίζουν τα signaling events βασισμένοι στην προηγούμενη γνώση και εμπειρία τους.

- 3 Όταν συμβαίνει ένα special event, ο πράκτορας αποθηκεύει το special event στην **special events base**. Πρέπει να σημειωθεί ότι όλα τα γεγονότα αποθηκεύονται στην belief base του πράκτορα αλλά μόνο τα special events αποθηκεύονται στην special events base.
- 4 Αν το γεγονός που έχει αντιληφθεί ο πράκτορας είναι ένα special event, ο πράκτορας ελέγχει αν αυτό υπάρχει στην **personal norm (p-norm) base** ή στην **group norm (g-norm) base**. Ένας πράκτορας κατέχει κάποιες p-norms βασισμένες στην παλαιότερη εμπειρία του ή προτίμησή του. Μία p-norm «υποδεικνύει» στον πράκτορα το τι είναι ορθό κατά αυτόν να πράξει. Παραδείγματος χάριν, ένας πράκτορας μπορεί να περιμένει ότι οι πράκτορες γενικά δεν ρυπαίνουν το πάρκο. Η πεποιθήσεις αυτές είναι *προσωπικές* και δεν είναι απαραίτητο αυτό να ισχύει και για τους υπόλοιπους πράκτορες του περιβάλλοντός του. Κάτι τέτοιο σημαίνει ότι η p-norm του κάθε πράκτορα είναι κάτι το προσωπικό και μπορεί να διαφέρει από πράκτορα σε πράκτορα. Ένας πράκτορας που τιμωρεί, μπορεί να αποθαρρύνει έναν άλλον πράκτορα που παραβιάζει την p-norm του με το να τον τιμωρήσει. Παραδείγματος χάριν, η p-norm ενός πράκτορα A μπορεί να είναι ότι κανείς δεν πρέπει να ρυπαίνει το πάρκο. Αν ένας πράκτορας B παραβιάσει αυτή την p-norm, τότε ο A μπορεί να τιμωρήσει τον B. Ο πράκτορας A μπορεί να θεωρηθεί σαν τον πράκτορα που προτείνει τη νόρμα και θεωρούμε ότι ο πράκτορας A έχει τη δυνατότητα να τιμωρήσει τον πράκτορα B. Ο πράκτορας συμπεραίνει τη g-norm μέσω είτε των αλληλεπιδράσεων που έχει ο ίδιος με τους υπόλοιπους πράκτορες είτε μέσω των αλληλεπιδράσεων των υπόλοιπων πρακτόρων τις οποίες παρατηρεί στην κοινωνία στην οποία βρίσκεται. Ο πράκτορας συμπεραίνει τις g-norms χρησιμοποιώντας το **norm inference component**. Η παρούσα διπλωματική έχει εστιαστεί στο πως συμπεραίνεται η g-norm. Όταν συμβαίνει ένα special event, ο πράκτορας μπορεί περιστασιακά να επικαλεστεί το norm inference component για να διαπιστώσει κατά πόσο αυτό είναι αποτέλεσμα κάποιας νόρμας που μέχρι τότε ήταν άγνωστη στον πράκτορα. Αυτή η διαδικασία μπορεί να οδηγήσει στην

αναγνώριση μιας νέας g-norm η οποία και προστίθεται στην αντίστοιχη g-norm base. Οι δύο συνιστώσες του norm inference component είναι αυτή που κάνει αναγνώριση της υποψήφιας νόρμας (**candidate norm identification component**) και αυτή που κάνει επαλήθευση της νόρμας (**norm verification component**).

- 5 Όταν παρατηρείται ένα γεγονός, ο πράκτορας ελέγχει αν κάποια από τις νόρμες που διαθέτει στη βάση για p-norm ή για g-norm εφαρμόζεται στο γεγονός αυτό ώστε να αποφασίσει σε ποια πράξη θα πρέπει να προβεί.
- 6 Όταν ο πράκτορας έχει αποφασίσει ότι θα προβεί σε κάποια συγκεκριμένη πράξη επικαλείται και το Norm Applicability Assessment component ώστε να ελέγξει αν η πράξη αυτή έρχεται σε αντίθεση με τις νόρμες του.

Η υπόλοιπη μεταπτυχιακή διατριβή επικεντρώνεται στον μηχανισμό αναγνώρισης νορμών (Norm Identification Component), την επεξήγηση και την υλοποίησή του.

*Τι μπορεί να αναγνωρίσει μηχανισμός αναγνώρισης νορμών.*

Ο μηχανισμός αναγνώρισης νορμών μπορεί να αναγνωρίσει δύο τύπους νορμών, τις *νόρμες απαγόρευσης (prohibition norms)* και τις *νόρμες υποχρέωσης (obligation norms)*. Οι νόρμες απαγόρευσης συμπεραίνονται από παρόμοια γεγονότα που έχουν συμβεί στο παρελθόν (π.χ. ένας πράκτορας αναγνωρίζει ότι το να ρυπάνει το περιβάλλον του συνεπάγεται κυρώσεις). Για να αναγνωρίσει ένας πράκτορας μία νόρμα υποχρέωσης παρατηρεί μετά από ποια γεγονότα που δεν συνέβησαν επήλθε η κύρωση (π.χ. η μη πληρωμή του λογαριασμού του εστιατορίου).

*Πότε ο πράκτορας επικαλείται το μηχανισμό αναγνώρισης νορμών.*

Ο πράκτορας, σαν αυτόνομη οντότητα, μπορεί να αποφασίσει ότι δε χρειάζεται να επικαλεστεί το μηχανισμό αναγνώρισης νορμών κάθε φορά που αντιλαμβάνεται ένα special event. Αντίθετα, μπορεί αυτό να γίνεται με κάποια περιοδικότητα. Μία καινούρια g-norm μπορεί να προκύψει όταν ένα γεγονός αξιολογείται από το μηχανισμό αναγνώρισης νορμών. Η νέα αυτή g-norm προστίθεται στην g-norm base. Αν ο πράκτορας δεν καταφέρει να βρει κάποια g-norm μπορεί να τροποποιήσει κάποιες από τις παραμέτρους του μηχανισμού αναγνώρισης νορμών και να επαναλάβει τη διαδικασία εύρεσης νόρμας ή μπορεί να αναμείνει ώστε να συλλέξει περισσότερες πληροφορίες.

Σε τακτά χρονικά διαστήματα, ο πράκτορας επανεκτιμά με τη βοήθεια του norm inference component τις g-norms που ήδη έχει ώστε να διαπιστώσει κατά πόσο αυτές ισχύουν ακόμα. Όταν μία g-norm που κατέχει ισχύει ακόμα, τη διατηρεί. Διαφορετικά, την αφαιρεί από την g-norm base.



*Πότε δεν υπάρχει τιμωρία για έναν πράκτορα παραβάτη;*

Υπάρχουν δύο λόγοι για τους οποίους μπορεί να μην υπάρχει τιμωρία για τους πράκτορες-παραβάτες στην κοινωνία. Ο πρώτος είναι το γεγονός ότι μπορεί να μην υπάρχουν πράκτορες –τιμωροί στην κοινωνία. Σε αυτή την περίπτωση η νόρμα διαγράφεται. Ο δεύτερος είναι ότι υπάρχει η πιθανότητα ο πράκτορας να έχει αποδεχτεί τη νόρμα και να την εφαρμόζει απόλυτα οπότε δεν εγείρει και κυρώσεις. Οι πράκτορες που εγκαταλείπουν μία κοινωνία μπορούν να σβήσουν τις νόρμες που αφορούν την κοινωνία αυτή.

## 4.2 Μηχανισμός Αναγνώρισης Νορμών.

Διαχωρίζουμε τα μέρη του μηχανισμού αναγνώρισης νορμών σε δύο κατηγορίες. Η πρώτη είναι αυτή των μερών όπου αποθηκεύονται τα γεγονότα (**event storage components**) και η δεύτερη είναι ο ίδιος ο μηχανισμός αναγνώρισης νορμών (**norm inference component**).

### 4.2.1 Event storage component

Ας υποθέσουμε ότι ένας πράκτορας βρίσκεται σε ένα πάρκο. Ο πράκτορας γνωρίζει ότι βρίσκεται στο πάρκο και ότι υπάρχουν αλληλεπιδράσεις μέσα στα πλαίσια αυτού. Επιπλέον, κάθε πράκτορας γνωρίζει πληροφορίες σχετικές με το περιβάλλον, όπως το πού βρίσκεται ο κάδος απορριμμάτων. Ας υποθέσουμε ότι αρχικά δεν υπάρχει καμία νόρμα σχετική με τη ρύπανση του πάρκου, αλλά μερικοί πράκτορες γνωρίζουν τι πρέπει να κάνουν σε συγκεκριμένες καταστάσεις (βασισμένοι στην προσωπική τους *p-norm*). Σε αυτή την αρχιτεκτονική ο πράκτορας αρχικά παρατηρεί τις αλληλεπιδράσεις μεταξύ των πρακτόρων στο περιβάλλον – πάρκο. Οι αλληλεπιδράσεις είναι δύο τύπων. Ο πρώτος τύπος ονομάζεται *προσωπική αλληλεπίδραση* (**personnel interaction**) και εμπλέκεται ο ίδιος ο πράκτορας σε αυτή, π.χ. μία πράξη στην οποία προβαίνει ο πράκτορας μέσα στο περιβάλλον ή ένα μήνυμα που ανταλλάσσει με κάποιον άλλον πράκτορα. Ο δεύτερος τύπος ονομάζεται *παρατηρούμενη αλληλεπίδραση* (**observed interaction**) και είναι η αλληλεπίδραση μεταξύ άλλων πρακτόρων που παρατηρεί ο πράκτορας. Ο πράκτορας αποθηκεύει όλα τα παραπάνω γεγονότα στην βάση δεδομένων των πεποιθήσεών του (**belief base**).

Ένας πράκτορας σε μία κοινωνία μπορεί να έχει έναν ή περισσότερους από τους παρακάτω ρόλους:

- **Μετέχων (Participant (P))** : ο πράκτορας που συμμετέχει προσωπικά σε μια αλληλεπίδραση.



- Παρατηρητής (**Observer (O)**) : ο πράκτορας που παρατηρεί τις δραστηριότητες των άλλων πρακτόρων.
- Ο πράκτορας που αποδίδει είτε τις κυρώσεις είτε τις ανταμοιβές (**Signaler (S)**). Οι κυρώσεις αποδίδονται σε πράκτορες που δρουν αντίθετα με τις p-norms ή τις g-norms του πράκτορα-signaler.

Συνεχίζοντας στο παράδειγμά μας, ένας πράκτορας μέσα σε ένα πάρκο μπορεί π.χ. να τρώει (litter), να ρυπαίνει (litter), να μετακινείται (move) κ.α. Ορισμένοι πράκτορες μπορεί να θεωρούν το να ρυπαίνει κανείς το πάρκο, βασισμένοι στις p-norms τους και σε παλαιότερες εμπειρίες τους), σαν μία πράξη που πρέπει να αποθαρρύνεται. Έτσι επιλέγουν να προχωρούν στο να ειδοποιήσουν τους «παραβάτες» είτε φωνάζοντάς τους είτε κουνώντας το κεφάλι τους ως ένδειξη δυσαρέσκειας. Υποθέτουμε ότι ένας πράκτορας έχει μηχανισμούς που διαχωρίζει ενδείξεις σαν τις παραπάνω ως **signaling actions**. Τα **signaling events** μπορεί να είναι είτε *θετικά* (ανταμοιβές) είτε *αρνητικά* (κυρώσεις). Τα signaling events αποθηκεύονται στην **special events base**. Ένας signaler μπορεί να θεωρηθεί και ως ο πράκτορας που ουσιαστικά προτείνει νόρμες (**proposer**). Ένας πράκτορας proposer καταλήγει σε μία νόρμα και την προτείνει και στους υπόλοιπους. Δεν είναι απαραίτητο πως ενός τέτοιου είδους πράκτορας πρέπει να είναι αναγκαστικά και signaler, δηλαδή δεν είναι αυτός που αποδίδει κυρώσεις σε πράκτορες που δεν ακολουθούν τις νόρμες. Αυτή είναι δουλειά του πράκτορα που έχει το ρόλο της «εξουσίας» στην κοινωνία (**authoritative leader**). Το ίδιο το σύστημα μπορεί επίσης να αποδίδει κυρώσεις στους πράκτορες – παραβάτες των νορμών.

Υποθέτουμε ότι οι πράκτορες έχουν ένα συγκεκριμένο, περιορισμένο πεδίο παρατήρησης. Μπορούν δηλαδή να δουν ό,τι συμβαίνει στη γειτονιά τους. Ένας παρατηρητής καταγράφει τις κινήσεις του άλλου πράκτορα μέχρι αυτός να εξαφανιστεί από το οπτικό του πεδίο. Όταν ένας πράκτορας παρατηρήσει ένα συμβάν το οποίο παραβιάζει τις νόρμες του προχωρεί σε κυρώσεις. Ο πράκτορας που παρατηρεί την όλη αλληλεπίδραση μπορεί να συμπεράνει ότι κάποιος που είχε εμπλακεί σε μια αλληλεπίδραση παρέβη κάποια νόρμα. Ακόμα και αν ο παρατηρητής πράκτορας γνωρίζει ότι κάποιος πράκτορας έχει υποστεί κάποια κύρωση είναι δυνατό να μη γνωρίζει το λόγο για τον οποίο έχει επιβληθεί η συγκεκριμένη κύρωση.

*Τι αποθηκεύει ένας πράκτορας.*

Ένα γεγονός που γίνεται αντιληπτό από έναν πράκτορα αποτελείται από ένα δείκτη του γεγονότος (*event index*), την αντιληπτή δράση (*observed action*) και τον πράκτορα που λαμβάνει μέρος σε αυτό το γεγονός. Έτσι η παραπάνω αναπαράσταση της δραστηριότητας ενός πράκτορα μπορεί να γίνει ως :  $happen(1, eat(A))$ . Σύμφωνα με αυτή, στη χρονική στιγμή 1, ο πράκτορας A εκτελεί την πράξη eat. Ένα παράδειγμα αναπαράστασης των γεγονότων που ένας πράκτορας έχει παρατηρήσει μπορεί να αναπαρασταθεί ως :

$do(1, eat(A))$

$do(2, litter(A))$



do( 3, move( B ) )

do( 4, move( A ) )

do( 5, sanction( B, A ) )

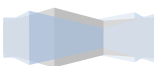
Ένας πράκτορας καταγράφει τα παραπάνω γεγονότα στην belief base του. Το γεγονός 5 είναι ένα sanctioning event κατά το οποίο ο πράκτορας B αποδοκιμάζει τον πράκτορα A.

## 4.2.2 Norm inference component

Ο μηχανισμός αναγνώρισης νομών αποτελείται από δύο επιμέρους μέρη. Το πρώτο μέρος κάνει χρήση του **Candidate Norm Inference (CNI) αλγορίθμου** για να παραγάγει υποψήφιας νόμους. Ο CNI αλγόριθμος αποτελείται με τη σειρά του από δύο επιπλέον μέρη τα οποία αναγνωρίζουν αντίστοιχα τις νόμους απαγόρευσης και υποχρέωσης. Το δεύτερο μέρος είναι υπεύθυνο για την επαλήθευση της υποψήφιας νόμους ώστε αυτή να αποτελέσει και νόμα της κοινωνίας.

### 4.2.2.1 Candidate Norm Identification component

Υποψήφιας ονομάζονται οι νόμους οι οποίες είναι ενδεχόμενες νόμους μιας κοινωνίας. Ο μηχανισμός αναγνώρισης υποψήφιας νομών έχει τη δυνατότητα να αναγνωρίζει δύο τύπους νόμους, τις *νόμους απαγόρευσης* και τις *νόμους υποχρέωσης*. Η αναγνώριση των υποψήφιας νομών αποτελείται από τρία βήματα. Αρχικά, βασισμένος στην παρατήρηση, ο πράκτορας φτιάχνει ακολουθίες γεγονότων (**event sequences**). Αυτές είναι ακολουθίες γεγονότων που συμβαίνουν μεταξύ των πρακτόρων που αλληλεπιδρούν. Στο δεύτερο βήμα, ο πράκτορας διαχωρίζει και ξεχωρίζει τις ακολουθίες γεγονότων βασισμένος στο γεγονός ότι τον ενδιαφέρουν μόνο τα γεγονότα που προηγούνται είτε μιας τιμωρίας είτε μιας επιβράβευσης γιατί τελικά αυτά μόνο καθορίζουν μια νόμα. Στην τρίτο βήμα γίνεται η αναγνώριση της νόμους. Οι νόμους απαγόρευσης και υποχρέωσης που αναγνωρίζονται αποθηκεύονται στο **candidate prohibition norm set (CPNS)** και στο **candidate obligation norms sets (CONS)** αντίστοιχα. Στο παρακάτω σχήμα φαίνεται η προαναφερθείσα διάταξη. Στην πρώτη στήλη αναπαρίστανται τα βήματα της διαδικασίας με ορθογώνια με στρογγυλεμένες γωνίες. Στη δεύτερη και τρίτη στήλη αναπαρίστανται η διαδικασία εύρεσης της νόμους απαγόρευσης και της νόμους υποχρέωσης αντίστοιχα. Τα αποτελέσματα αυτής της διαδικασίας χρησιμοποιούνται για τις επόμενες.



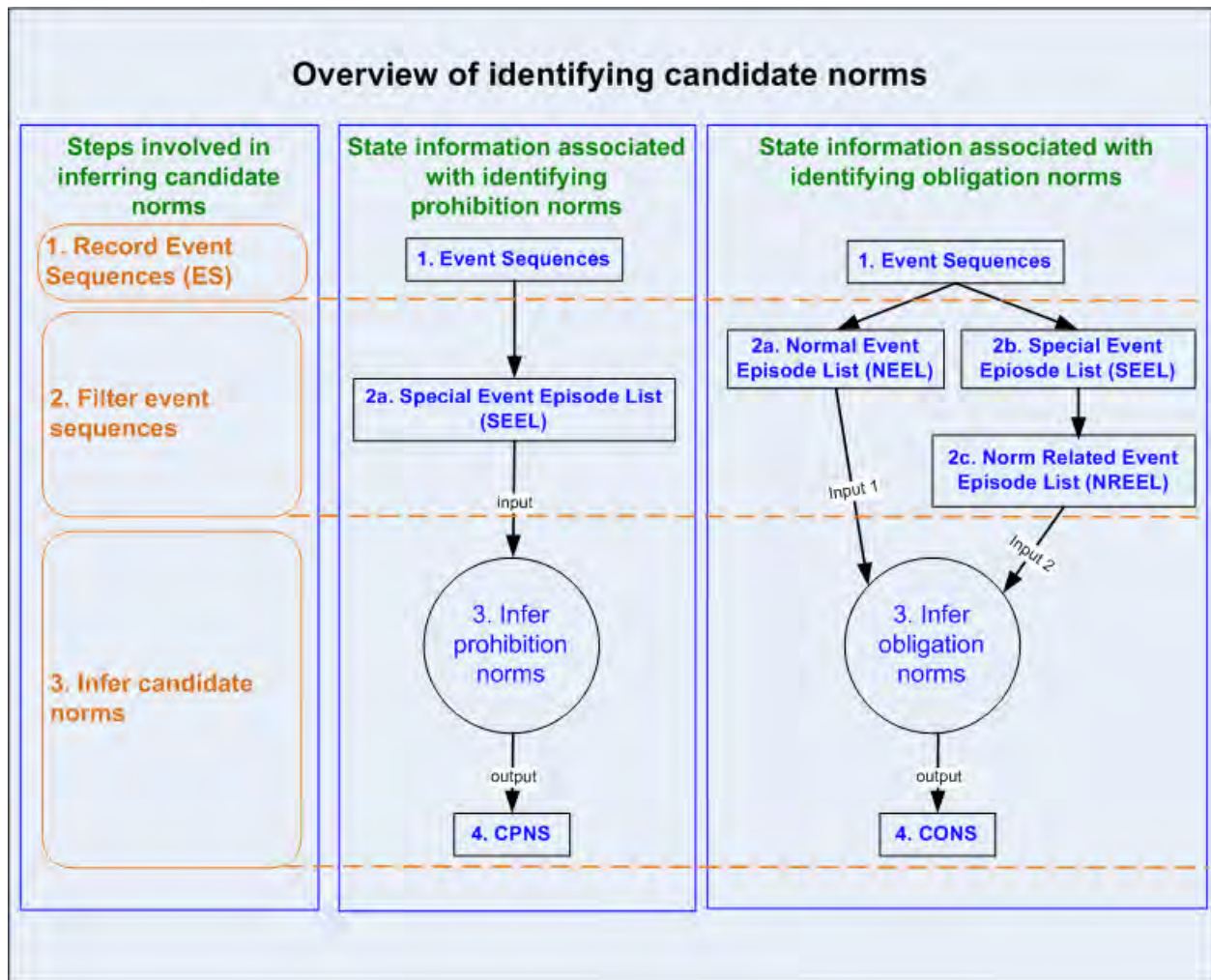


Figure 3 Γενική εικόνα μηχανισμού αναγνώρισης νορμών

#### 4.2.2.2 Norm Verification Component

Κατά τη διαδικασία αυτή η κάθε υποψήφια νόρμα αξιολογείται για το κατά πόσο μπορεί να αποτελέσει και νόρμα της κοινωνίας. Ο πράκτορας ρωτά άλλους πράκτορες στη γειτονιά του κατά πόσο μία νόρμα ισχύει στην κοινωνία. Αν η απάντηση είναι θετική τότε προσθέτει τη νόρμα αυτή στην g-norm base του. Διαφορετικά, αν η απάντηση είναι αρνητική ή ο γειτονικός πράκτορας δε γνωρίζει κατά πόσο αυτή η νόρμα ισχύει στο συγκεκριμένο περιβάλλον, ο πράκτορας είτε ρωτάει άλλο γειτονικό πράκτορα ή μαζεύει περισσότερη πληροφορία.

*Γιατί ο πράκτορας ρωτάει γειτονικούς πράκτορες για την επιβεβαίωση μιας νόρμας*

Ένας πράκτορας όταν εισέρχεται σε ένα περιβάλλον μπορεί να μην ενδιαφέρεται στο να μάθει όλες τις νόρμες που διέπουν το περιβάλλον αλλά μόνο νόρμες που αφορούν σε ένα πολύ συγκεκριμένο αντικείμενο. Ο πράκτορας σε αυτή την περίπτωση πρέπει να συμπεράνει τις σχετικές με το αντικείμενο νόρμες βασισμένος σε ό,τι εκείνος παρατηρεί μέσα από τις αλληλεπιδράσεις των άλλων πρακτόρων και σε επόμενη φάση να ζητήσει την επιβεβαίωση της νόρμας που συμπεράνε από γειτονικούς πράκτορες.

Επίσης, σε ένα πιο ρεαλιστικό περιβάλλον, ένας πράκτορας μπορεί να μην εμπιστεύεται άλλους πράκτορες ως προς το αν, σε περίπτωση που εκείνος ζητήσει μια νόρμα που διέπει το περιβάλλον, εκείνοι θα τον πληροφορήσουν σωστά. Για αυτό ένας πράκτορας θέλει να είναι σίγουρος για την ορθότητα της νόρμας που αργότερα θα χρησιμοποιήσει και δε θα διατρέχει τον κίνδυνο να παρασυρθεί κακόβουλα από άλλους πράκτορες της κοινωνίας. Πρέπει να σημειωθεί ότι αυτό δε λύνει απαραίτητα το πρόβλημα του ψεύδους ακόμα και στη φάση που ο πράκτορας ζητάει την επιβεβαίωση μιας νόρμας από γειτονικούς του πράκτορες (οι οποίοι και μπορούν να τον πληροφορήσουν λανθασμένα), αλλά τουλάχιστον ο πράκτορας έτσι έχει τη δυνατότητα να σχηματίσει κάποιες υποψήφιες νόρμες μόνος του.

## 5 Νόρμες Απαγόρευσης (Prohibition Norms)

Σε αυτό το κεφάλαιο παρουσιάζεται πιο αναλυτικά η αρχιτεκτονική με βάση την οποία μπορούμε και συμπεραίνουμε *νόρμες απαγόρευσης*. Ο μηχανισμός αυτός βασίζεται ουσιαστικά σε έναν αλγόριθμο που βοηθά τον πράκτορα να εντοπίσει τη νόρμα απαγόρευσης και θα αναφέρεται ως **Prohibition Norm Identification algorithm (PNI)**. Παρακάτω παρουσιάζονται αναλυτικά οι παράμετροι του αλγορίθμου καθώς και τα μέρη του μηχανισμού κατά τα οποία γίνεται η αναγνώριση των υποψήφιων νορμών (candidate norm identification component) και η επιβεβαίωση των νορμών (norm verification component).



## 5.1 Παράμετροι του μηχανισμού αναγνώρισης νομών

**History Length ( HL )** : Ένας πράκτορας κρατά ένα ιστορικό των αλληλεπιδράσεων που παρακολουθεί κατά τη διάρκεια ενός χρονικού παραθύρου το οποίο αναπαρίσταται από την παράμετρο History Length (HL). Π.χ. αν ένας πράκτορας κρατά γεγονότα κατά τη διάρκεια ενός χρονικού παραθύρου HL = 20, τότε κρατάει τα γεγονότα που έχουν συμβεί κατά τα είκοσι τελευταία βήματα στη μνήμη του.

**Event Sequences ( ES )** : Είναι η καταγραφή των γεγονότων που ο πράκτορας παρατηρεί μέσα στο ιστορικό του. Για παράδειγμα, αν το HL = 5, τότε το ES ενός πράκτορα που παρατηρεί δύο άλλους πράκτορες A και B φαίνεται στην εικόνα 5.1.

**Special Events Set ( SES )** : Κάποια από τα γεγονότα που παρατηρεί ο πράκτορας τα ονοματίζει ως special events και είναι τα signaling events, και σε αυτά συμπεριλαμβάνονται όπως έχουμε ήδη πει γεγονότα που προκαλούν κάποια «συναισθηματική αντίδραση» στον πράκτορα, είτε θετική (rewards) είτε αρνητική (sanctions). Θεωρούμε ότι ο πράκτορας έχει τη δυνατότητα να διαχωρίσει σε rewards και sanctions τα γεγονότα. Έτσι σχηματίζεται το SES το οποίο θα μπορούσε π.χ. να είναι της μορφής  $SES = \{ yell, disapproval head-shake \}$ .

**Unique Events Set ( UES )** : Περιλαμβάνει τον αριθμό των διακριτών γεγονότων που συμβαίνουν κατά τη διάρκεια ενός συγκεκριμένου χρονικού διαστήματος. Π.χ. για το παράδειγμα με το πάρκο ένα UES μπορεί να είναι  $UES = \{ eat, litter, move, sanction \}$ .

**Event Pruning Threshold ( EPT )** : Στη σειρά των γεγονότων που ο πράκτορας παρατηρεί ώστε να καταλήξει σε μια νόρμα, υπάρχουν και γεγονότα που δεν σχετίζονται με αυτή. Για παράδειγμα στο σενάριο με τον πράκτορα στο πάρκο, η ενέργεια *move*, το γεγονός δηλαδή ότι ο πράκτορας κινείται, δεν αποτελεί σημαντικό γεγονός το οποίο καθορίζει μια νόρμα. Για να αποκλείσουμε αυτού του είδους τα γεγονότα έχει οριστεί το EPT. Αν αυτό, για παράδειγμα έχει καθοριστεί ίσο με 0.5, τα γεγονότα με Occurrence Probability μεγαλύτερο από 0.5 αποκλείονται από μια υποψήφια νόρμα.

**Occurrence Probability ( OP )** : Η πιθανότητα εμφάνισης ενός γεγονότος E δίνεται από τον παρακάτω τύπο :

$$OP(E) = \frac{\text{Number of occurrences of } E}{\text{Total number of events in ES}}$$

**Window size ( WS )** : Όταν ένας πράκτορας θέλει να συμπεράνει μία νόρμα κοιτάει στο ιστορικό του ένα συγκεκριμένο αριθμό γεγονότων που προηγούνται μιας κύρωσης. Π.χ. αν το WS = 3, τότε ο πράκτορας σχηματίζει το **Special Event Episode (SEE)** του με 3 γεγονότα να προηγούνται ενός special event. Το SEE είναι υποσύνολο του ES.

**Norm Identification Threshold ( NIT )** : Όταν ο πράκτορας είναι κοντά στο να καταλήξει σε υποψήφιας νόρμες, τότε μπορεί να μην ενδιαφέρεται για γεγονότα τα οποία έχουν μικρότερη πιθανότητα να αποτελέσουν νόρμα. Για παράδειγμα, αν το NIT ενός πράκτορα είναι 50, αυτό δηλώνει ότι ο πράκτορας αυτός ενδιαφέρεται για γεγονότα τα οποία έχουν 50% πιθανότητα να αποτελέσουν υποψήφια νόρμα.



**Norm Inference Frequency ( NIF )** : Είναι η συχνότητα κατά την οποία ο πράκτορας επιλέγει να επικαλείται το Norm Inference Component. Ανάλογα με τις επιλογές του πράκτορα αυτό μπορεί να γίνεται κάθε φορά που αυτός παρατηρεί ένα special event είτε περιοδικά. Η παράμετρος αυτή μπορεί να αλλάζει περιοδικά, ανάλογα αν μιλάμε για ένα περιβάλλον που οι νόρμες που το διέπουν δεν αλλάζουν συχνά, οπότε δεν είναι αναγκαίο να καλείται και συχνά το NIC, είτε όχι.

<i>Agent A</i>	<i>Agent B</i>
<i>do ( 1, eat( A ) )</i>	<i>do ( 1, eat( B ) )</i>
<i>do ( 2, litter( A ) )</i>	<i>do ( 2, move( B ) )</i>
<i>do ( 3, move( A ) )</i>	<i>do ( 3, move( B ) )</i>
<i>do ( 4, move( A ) )</i>	<i>do ( 4, sanction( B, A ) )</i>
<i>do ( 5, move( A ) )</i>	<i>do ( 5, move( B ) )</i>

Figure 4 Γεγονότα που παρατηρεί ένας πράκτορας

## 5.2 Candidate Norm Inference component

Μέχρι την αναγνώριση της νόρμας μεσολαβούν δύο στάδια. Αρχικά, σύμφωνα με το μηχανισμό για το διαχωρισμό των γεγονότων, τα Special Event Episodes (SEE) εξάγονται από τα Event Sequences (ES) που έχει ο πράκτορας καταγράψει. Σε δεύτερη φάση, και βασισμένος στα Special Event Episodes (SEE) που έχει εξάγει, παράγει την **Candidate Prohibition Norm Set (CPNS)** .

Ακολουθεί ο σχετικός αλγόριθμος :

```

1   foreach invocation of the norm inference component do
2       Create event episodes; /* see Algorithm */
3       Prune event episodes; /* see Algorithm */
4       Create Candidate Norm List ( CNL ); /* see Algorithm */
5   end

```

Αλγόριθμος 5.1 Norm Inference algorithm

### 5.2.1 Event Episodes (EE)

Ας υποθέσουμε ότι υπάρχουν τρεις πράκτορες, ο A, ο B και ο C. Ο πράκτορας C είναι αυτός που παρατηρεί τις αλληλεπιδράσεις μεταξύ των άλλων δύο πρακτόρων. Ένα παράδειγμα σειράς γεγονότων

που παρατηρεί ο πράκτορας C θα μπορούσε να είναι αυτό που παρουσιάζεται στον *Εικόνα 5.1*. Υποθέτουμε επίσης ότι οι πράκτορες μπορούν να κάνουν μόνο μία ενέργεια κάθε χρονική στιγμή. Στο προαναφερθέν παράδειγμα, ο πράκτορας A τη στιγμή 1 τρώει (eat), τη χρονική στιγμή 2 ρυπαίνει (litters) και τις επόμενες μετακινείται (moves). Ο πράκτορας B τη χρονική στιγμή 1 τρώει (eat), τις επόμενες δύο μετακινείται, τη χρονική στιγμή 4 επιπλήττει (sanction) τον πράκτορα A που ρύπανε το περιβάλλον και τέλος μετακινείται πάλι.

Όταν ο πράκτορας παρατηρεί ένα special event, εξάγει τη σειρά των γεγονότων που έχουν λάβει μέρος μεταξύ των πρακτόρων που αλληλεπιδρούν και που προηγούνται του special event από την καταγεγραμμένο του ιστορικό (Event Sequences). Στο δοθέν παράδειγμα ο πράκτορας – παρατηρητής διαπιστώνει ότι κάτι που έκανε ο πράκτορας A ή κάτι που απέτυχε να κάνει πρέπει να προκάλεσε την αντίδραση του B. Ο πράκτορας C εξαγάγει την παρακάτω σειρά γεγονότων που έλαβαν μέρος μεταξύ του A και του B.

$$\{ A, B \} \rightarrow happens(1, eat(A)) - happens(2, litter(A)) - happens(3, move(A)) - happens(4, sanction(B, A))$$

Όπως αναφέρθηκε και παραπάνω, ονομάζουμε μία σειρά γεγονότων που προηγείται ενός special event ως *event episode (EE)*. Για ευκολία συμβολισμού η σειρά γεγονότων θα αναπαρίσταται μόνο με το πρώτο γράμμα του γεγονότος. Δηλαδή το παραπάνω θα γίνει :

$$\{ A, B \} \rightarrow e-l-m-s$$

Ο πράκτορας μπορεί να παρατηρεί πολλές εναλλακτικές σειρές γεγονότων που προηγούνται ενός special event. Αν τώρα περιορίσουμε το WS σε 3 μπορούμε να κατασκευάσουμε μία λίστα με τέτοιες σειρές γεγονότων, την **Event Episode List (EEL)**. Ένα παράδειγμα μιας τέτοιας λίστας φαίνεται και παρακάτω.

$$(e-l-m-s, l-e-l-s, m-e-l-s, e-l-e-s, e-l-e-s, l-e-l-s, e-e-l-s, m-e-l-s, e-l-m-s, e-l-e-s)$$

Παρακάτω ακολουθεί ο αλγόριθμος βάσει του οποίου εξάγεται η Event Episode List από τη σειρά των γεγονότων που παρατηρεί ο πράκτορας.

```

Input: Event Sequence (ES), Window Size (WS)
Output: Event Episode List (EEL)
1  foreach invocation of the norm inference component do
2      foreach special event in ES do
3          Create an Event Episode (EE) with the last n events that precede the special event
           where n=WS;
4          Store EE in EEL;
5      end
6  end

```

#### Αλγόριθμος 5.2 Event Episode List algorithm

## 5.2.2 Pruning Event Episodes List (PEEL)

Για κάθε special event στο Event Sequence (ES), ο πράκτορας κατασκευάζει το Event Episode (EE) με  $n$  events να προηγούνται του special event, όπου  $n = WS$ . Το Event Episode (EE) προστίθεται στο Pruning Event Episode List (PEEL). Στόχος είναι να περικοπούν τα γεγονότα που μπορεί να μην είναι σχετικά με την νόρμα. Όπως αναφέρθηκε και παραπάνω, τέτοιου είδους γεγονότα αφαιρούνται από το EEL βάσει του γεγονότος ότι η πιθανότητα εμφάνισής τους (Occurrence probability) είναι πάνω από το Norm Pruning Threshold (NPT).

Ο υπολογισμός του Pruning Event Episode List (PEEL) υπολογίζεται βάσει του παρακάτω ψευδοκώδικα

**Input:** Event Episode List (EEL), Unique Event Set (UES), Norm Pruning Threshold (NPT)

**Output:** Pruned Event Episode List (PEEL)

*/\* construct Unique Event Set (UES) \*/*

```
1   foreach event E in ES do
2       if E not_exists_in Unique Event Set (UES) and E not_exists_in Special Event Set (SES)
3           then
4               Add E to UES;
5           end
6       end
7   end
```

*/\* construct Removable Event Set (RES) \*/*

```
6   foreach event E in Unique Event Set (UES) do
7       Calculate OP(E);
8       if OP(E) >= NPT then
9           Add event to Removable Event Set (RES);
10      end
11  end
```

*/\* construct Pruned Event Episode List (PEEL) \*/*

```
12  foreach Event Episode (EE) in EEL do
13      foreach event E in an EE do
14          if event E in RES then
15              Remove event from EE;
16          end
17      end
18  end
```

*Αλγόριθμος 5.3 Pruning Event Episode List*



### 5.2.3 Υποψήφιες Νόρμες (Candidate Norms)

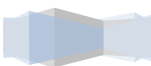
Ο αλγόριθμος για την παραγωγή των υποψήφιων νορμών παρουσιάζεται από τον αλγόριθμο 5.4 και αποτελεί μία τροποποίηση το αλγορίθμου WINEPI, ο οποίος ανακαλύπτει συχνά επεισόδια , σειρές γεγονότων δηλαδή, τα οποία εμπεριέχονται σε Event Sequences. Ο αλγόριθμος 5.4 λειτουργεί επαναληπτικά. Ο αριθμός των επαναλήψεων ισούται με το Window Size (WS) που έχει οριστεί. Τα επιμέρους επεισόδια της πρώτης επανάληψης είναι μήκους 1. Η λίστα με τα επιμέρους επεισόδια **Sub Episode List (SEL)** για την επανάληψη 1 περιέχει όλα τα γεγονότα που συμπεριλαμβάνονται στο UES. Παραδείγματος χάριν, για το παράδειγμα που δόθηκε στην *Εικόνα 5.1*, κατά την πρώτη επανάληψη του αλγορίθμου η λίστα SEL θα περιλαμβάνει τα γεγονότα e, l και m. Μετά από κάθε επανάληψη υπολογίζονται οι πιθανότητες εμφάνισης για κάθε επιμέρους επεισόδιο. Αν η πιθανότητα εμφάνισης ενός επιμέρους επεισοδίου στην PEEL είναι μεγαλύτερη ή ίση από το NIT τότε το επεισόδιο προστίθεται στο Candidate Norm List (CNL). Επίσης κάθε υποψήφια νόρμα προστίθεται σε μία προσωρινή λίστα η οποία χρησιμοποιείται για τη δημιουργία του SELγια την επόμενη επανάληψη. Για τον υπολογισμό του SEL<sub>next</sub> γίνεται χρήση άλλου αλγορίθμου και η τρέχουσα τιμή του εκχωρείται στη μεταβλητή SEL<sub>current</sub>.

**Input:** Special Event Episode List (SEEL), Unique Event Set (UES), Window Size(WS), Norm Inference Threshold(NIT)

**Output:** Candidate Prohibition Norms Set (CPNS)

```
1  begin
2      CNL ← 0;
3      iterNum ← 1;
4      Sub-Episode List (SELcurrent) ← UES;
5      while iterNum ≤ WS do
6          SELtemp ← 0;
7          foreach Sub-Episode(SE) in SEL that appears in PEEL do
8              if iterNum = 1 then
9                  if OP(SE) ≥ NIT then
10                     Add SE to CNL, Add SE to SELtemp;
11                 end
12             end
13             else
14                 if each event in SE exists in CNL and OP(SE) ≥ NIT then
15                     CNS ← SE, SELtemp ← SE;
16                 end
17             end
18         end
19         Construct SELnext using SELtemp ; /* Algorithm 5.5*/
20         if iterNum < WS then
21             iterNum ← iterNum + 1;
22             SELcurrent ← SELnext;
23         else
24             return CNL;
25         end
26     end
27 end
```

Αλγόριθμος 5.3 Candidate Norm List (CNL)



Κατά τη δεύτερη επανάληψη, κάθε επιμέρους επεισόδιο θα περιλαμβάνει δύο γεγονότα. Το επιμέρους επεισόδιο θα προστίθεται στο CNL αν :

- Κάθε γεγονός στο επιμέρους επεισόδιο υπάρχει ήδη στο CNL.
- Η πιθανότητα εμφάνισης του επιμέρους επεισοδίου σε ένα PEEL πρέπει να είναι μεγαλύτερη ή ίση του NIT.

Ο παρακάτω αλγόριθμος υπολογίζει τα SEL. Το μέγιστο μήκος μιας υποψήφιας νόρμας αλλά και ο αριθμός των επαναλήψεων του προηγούμενου αλγορίθμου είναι όσο και το WS , όπως προαναφέρθηκε.

```
Input: Candidate norms list (tempList)
Output: Sub Episode List (SEL)
1  foreach candidate norm in tempList do
2      Generate Sub-Episodes (SE) of length n using other candidate norms (allowing
      repetition of events);
3      Add each generated SE to SEL;
4  end
```

#### Αλγόριθμος 5.5 Creation of Sub – Episode List

#### Παράδειγμα

Αφού πλέον έχουν παρουσιαστεί οι παράμετροι και οι αλγόριθμοι βάσει των οποίων υπολογίζεται μία υποψήφια νόρμα απαγόρευσης, ακολουθεί ένα σχετικό παράδειγμα . Υποθέτουμε τη σειρά των γεγονότων ( PEEL ) και τους πράκτορες που λαμβάνουν μέρος σε αυτά όπως παρουσιάζονται στην εικόνα 5.1. Επιπλέον, υποθέτουμε ότι πριν από ένα special event προηγούνται τρία γεγονότα. Επίσης, το NIT θέτεται σε 50% και το unique event set (UEE) είναι το { e, l, m }.

Στην πρώτη επανάληψη, το SEL<sub>current</sub> περιλαμβάνει επιμέρους γεγονότα μήκους 1, τα οποία είναι e, l και m. Για κάθε ένα από αυτά υπολογίζεται η πιθανότητα εμφάνισης. Σύμφωνα με το παράδειγμά μας, η πιθανότητα εμφάνισης για τα e και l είναι 100 % ενώ για το m είναι 40 %. Από τη στιγμή που τα e και l έχουν πιθανότητα εμφάνισης μεγαλύτερη από το NIT, ο πράκτορας προσθέτει αυτά τα δύο γεγονότα στις υποψήφιας νόρμες. Για τη δεύτερη επανάληψη, ο πράκτορας υπολογίζει επιμέρους επεισόδια μήκους 2. Και αυτά, για να αποτελέσουν υποψήφιας νόρμες πρέπει η πιθανότητα εμφάνισής τους να είναι μεγαλύτερη του NIT. Έτσι, βάσει των υποψήφιας νορμών που έχουν υπολογιστεί κατά την τρέχουσα επανάληψη γίνεται ο υπολογισμός των επόμενων υποψήφιας νορμών. Στο παράδειγμά μας, μετά τη δεύτερη επανάληψη οι υποψήφιας νόρμες που προκύπτουν είναι οι { ee, el, le, ll}. Από αυτές μόνο οι el και le είναι υποψήφιας αφού ξεπερνούν το NIT. Να υπενθυμίσουμε ότι το WS = 3 οπότε ο αλγόριθμος καταλήγει στην τρίτη επανάληψη όπου οι υποψήφιας νόρμες που προκύπτουν είναι οι { lel,

ele}. Αυτές έχουν πιθανότητα εμφάνισης κάτω από το NIT. Τελικά, οι υποψήφιας νόρμες απαγόρευσης είναι οι { e, l, el, le } .

### 5.3 Επιβεβαίωση Νόρμας

Για να επιβεβαιώσει ο πράκτορας ότι η νόρμα που έχει συμπεράνει είναι ορθή, συμβουλευτείται περιοδικά έναν άλλο γειτονικό του πράκτορα. Η διαδικασία που ακολουθείται μπορεί να περιγραφεί με το ακόλουθο παράδειγμα:

Έστω δύο πράκτορες που αλληλεπιδρούν. Ο A έχει επιλέξει μία υποψήφια νόρμα π.χ. el. Ρωτάει τον B κατά πόσο η συγκεκριμένη νόρμα ισχύει στο περιβάλλον τους. Αν ο B απαντήσει ότι η νόρμα αυτή ισχύει, τότε ο A αποθηκεύει αυτή τη νόρμα. Διαφορετικά, αν δηλαδή η απάντηση είναι αρνητική, ο A ρωτάει τον B για την επόμενη υποψήφια νόρμα.

Συγκεκριμένα, στην περίπτωση που ο πράκτορας B απαντήσει θετικά, αυτό σημαίνει ότι για την περίπτωση της υποψήφιας νόρμας el, ότι είναι μία νόρμα απαγόρευσης και ότι η σειρά των γεγονότων αυτών οδηγεί στην κυρώσεις. Ο πράκτορας τότε ρωτάει αν το l (*littering*) είναι αυτό που ευθύνεται για την κύρωση. Αν η απάντηση είναι θετική, τότε το littering θεωρείται ότι απαγορεύεται. Διαφορετικά ο πράκτορας συνεχίζει την ίδια διαδικασία επιβεβαίωσης νόρμας με τον επόμενο πράκτορα. Η διαδικασία αυτή συνεχίζεται έως ότου είτε *να βρεθεί μία νόρμα* είτε *μέχρι να μη βρεθεί κάποια νόρμα* οπότε η διαδικασία επανεκινείται μόλις μία τιμωρία προς πράκτορα παρατηρηθεί. Ακόμα και αν επιβεβαιωθεί μία υποψήφια νόρμα ως νόρμα της κοινωνίας, ο πράκτορας συνεχίζει τη διαδικασία επιβεβαίωσης και με τις υπόλοιπες υποψήφιας νόρμες που έχει στη βάση του. Έχοντας τελειώσει με τις νόρμες κατά της ρύπανσης ( littering ) συνεχίζει με τις υποψήφιας νόρμες κατά του φαγητού ( eating ).

Από τη στιγμή που ο πράκτορας επιβεβαιώσει τις υποψήφιας νόρμες του, μπορεί να τις χρησιμοποιεί εφόσον αυτές είναι σταθερές για ένα χρονικό διάστημα



# 6

## Νόρμες Υποχρέωσης ( Obligation Norms )

Σε αυτό το κεφάλαιο παρουσιάζεται πως ένας αυτόνομος πράκτορας είναι σε θέση να αναγνωρίσει νόρμες υποχρέωσης χρησιμοποιώντας τον Obligation Norm Inference ( **ONI** ) αλγόριθμο. Το παράδειγμα που χρησιμοποιείται είναι αυτό του πράκτορα που βρίσκεται σε ένα εστιατόριο και πρέπει να συμπεράνει ότι πρέπει να δώσει ένα φιλοδώρημα στο σερβιτόρο που τον εξυπηρετεί. Όπως συμπεραίνεται και μέσω του παραδείγματος παρακάτω, ο πράκτορας είναι ικανός να προσθέτει, αφαιρεί και τροποποιεί δυναμικά τις νόρμες του. Επίσης, όπως και στην περίπτωση των μηχανισμών για την αναγνώριση των νορμών απαγόρευσης, ο πράκτορας έχει τη δυνατότητα να τροποποιεί και τις παραμέτρους των μηχανισμών αυτών ώστε να φτάσει στο επιθυμητό αποτέλεσμα της εύρεσης της νόρμας.

### 6.1 Το παράδειγμα του εστιατορίου

Στο παράδειγμα του εστιατορίου, ο πράκτορας γνωρίζει ότι οι ενέργειες που μπορεί να γίνουν από έναν πράκτορα είναι οι εξής : η άφιξη ενός πράκτορα ( arrive ), η παραγγελία ενός προϊόντος ( order ), το να φαγωθεί το προϊόν που είχε παραγγελθεί ( eat ), η πληρωμή του προϊόντος ( pay ), το να δοθεί φιλοδώρημα στον σερβιτόρο ( tip ), και αναχώρηση από το εστιατόριο ( depart ). Ένας πράκτορας παρατηρητής παρακολουθεί και πάλι τις ενέργειες των υπόλοιπων πρακτόρων που βρίσκονται στο περιβάλλον – εστιατόριο και προσπαθεί να συμπεράνει τις νόρμες που διέπουν το περιβάλλον αυτό. Έστω ότι σε αυτό το σενάριο υπάρχουν οι πράκτορες A, B,C και W. Οι πράκτορες A και B είναι οι πελάτες τους εστιατορίου, ο πράκτορας C είναι ο πράκτορας παρατηρητής και ο πράκτορας W είναι ο σερβιτόρος. Ο πράκτορας A εισέρχεται στο εστιατόριο, παραγγέλνει φαγητό, τρώει, πληρώνει το λογαριασμό και εξέρχεται του καταστήματος. Μετά από την αναχώρηση του ο πράκτορας A δέχεται κύρωση από τον πράκτορα σερβιτόρο W. Ο πράκτορας παρατηρητής βλέπει ότι ο πράκτορας A δέχεται κύρωση και καταγράφει τα γεγονότα στην βάση αποθήκευσης των πεποιθήσεών του ( **belief base** ). Το πιθανότερο είναι να μην γνωρίζει ότι η κύρωση προήλθε από το γεγονός ότι ο πελάτης δεν έδωσε φιλοδώρημα στον σερβιτόρο. Στο μεταξύ, ο πράκτορας B μπορεί να έχει εισέλθει του εστιατορίου και να έχει παραγγείλει χωρίς αυτό να φαίνεται να επηρεάζει το ότι έχει να κάνει με τον πράκτορα A.



Μια πιθανή σειρά γεγονότων για το παραπάνω σενάριο θα μπορούσε να είναι η εξής:

```
do(1, arrive(A))
do(2, arrive(B))
do(3, order(A,W))
do(4, eat(A))
do(5, pay(A,W))
do(6, depart, (A))
do(7, sanction(W,A))
do(8, order(B,W))
```

## 6.2 Αναγνώριση Νόρμας Υποχρέωσης ( Obligation Norm identification - ONI)

Ο στόχος του μηχανισμού εύρεσης νορμών υποχρέωσης είναι να διευκρινιστούν τα γεγονότα τα οποία ενώ αναμενόταν να γίνουν δεν έγιναν και αυτό είχε ως αποτέλεσμα ο πράκτορας να δεχθεί κάποια κύρωση/ τιμωρία, όπως στο προαναφερθέν παράδειγμα , όπου ο πράκτορας A δέχθηκε κύρωση (sanction) επειδή δεν έδωσε φιλοδώρημα στον πράκτορα W. Ο παρακάτω αλγόριθμος χρησιμοποιείται για να βρει ακριβώς το λόγο που ο πράκτορας δέχθηκε την κύρωση.

### 6.2.1 Παράμετροι αλγόριθμου

Οι παράμετροι που χρησιμοποιούνται στον αλγόριθμο είναι ακριβώς οι ίδιοι με αυτούς που χρησιμοποιούνται και στον αλγόριθμο Απαγόρευσης, εκτός από μία μικρή διάκριση που γίνεται στο threshold σε  $NIT_{\alpha}$  και  $NIT_{\beta}$ .

### 6.2.2 Ο αλγόριθμος

Ο αλγόριθμος μπορεί να διαχωριστεί σε τέσσερα βήματα:

1. Εξάγονται τα event episodes ( EE ) από τα event sequences που παρατηρεί ο πράκτορας. Αυτά τα event episodes αποθηκεύονται στην Event Episodes List ( EEL ).
2. Σε δεύτερη φάση, βασισμένα στα special events set, τα event episodes διαχωρίζονται σε δύο λίστες. Η πρώτη λίστα, περιέχει όλα τα event episodes τα οποία περιέχουν τουλάχιστον ένα sanctioning event και ονομάζεται **Special Event Episode List ( SEEL )**. Η δεύτερη λίστα περιέχει όλα τα event episodes τα οποία δεν περιέχουν sanctioning event και ονομάζεται **Normal Event Episode List**.



3. Κατά την τρίτη φάση, όλα τα επιμέρους επεισόδια που έχουν πιθανότητα εμφάνισης μεγαλύτερη ή ίση από το  $NIT_a$  αποθηκεύονται στην **Norm – Related Event Episode Set ( NREES)** χρησιμοποιώντας τον αλγόριθμο CNL που περιγράφηκε στην προηγούμενη ενότητα.
4. Στην τέταρτη φάση, για κάθε event episode στην NREES, εξάγονται όλες οι ακολουθίες από την NEEL και αποθηκεύονται σε μία προσωρινή λίστα που ονομάζεται **tempEEList**. Βασισμένος σε αυτές τις λίστες ο CNL αλγόριθμος μπορεί να συμπεράνει όλες τις μεταθέσεις των σειρών με πιθανότητα εμφάνισης μεγαλύτερη ή ίση από το  $NIT_b$ . Αυτές αποθηκεύονται στο **Candidate Obligation Norm Set ( CONS )**.

```

1  begin
2      Create event episodes list (EEL);
3      Create special event episodes list (SEEL) and normal event episodes list (NEEL);
4      Extract norm-related event episodes set (NREES) from SEEL;
5      Create Candidate Obligation Norm Set (CONS) from NEEL using NREES ;
6  end

```

#### Αλγόριθμος 6.1 ONI αλγόριθμος

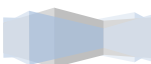
### 6.2.3 Event Episodes

Ο πράκτορας αποθηκεύει τις αλληλεπιδράσεις που παρατηρεί στο περιβάλλον του και που έχουν μήκος όσο η μεταβλητή History Length (HL). Από το καταγεγραμμένο ιστορικό των Event Sequences εξάγονται τα Event Episodes ( EE ). Ένα παράδειγμα ενός EE το οποίο είναι βασισμένο στην παραπάνω σειρά γεγονότων που παρουσιάστηκαν στο παράδειγμα είναι το εξής:

$$\{A, W\} : ( \text{do} ( 3, \text{order}, A, W ) - \text{do} ( 4, \text{eat}, A ) - \text{do} ( 5, \text{pay}, A, W ) - \text{do} ( 6, \text{depart}, A ) - \text{do} ( 7, \text{sanction}, W, A ) )$$

Βάσει αυτής της σειράς γεγονότων ο πράκτορας – παρατηρητής υποθέτει ότι κάτι που δεν έκανε ο πράκτορας A προκάλεσε την τιμωρία (sanction). Σε αντίθεση με την αναγνώριση νορμών απαγόρευσης, όπου η κύρωση προερχόταν από κάτι που είχε κάνει ο πράκτορας, στην αναγνώριση νορμών υποχρέωσης ο πράκτορας κυρώνεται για κάτι που δεν έκανε. Άρα για να αναγνωριστεί μια νόρμα υποχρέωσης πρέπει να εντοπιστεί τα γεγονότα που «λείπουν». Τότε θα είναι σε θέση και ο πράκτορας παρατηρητής να εξαγάγει τη σειρά των γεγονότων event episodes που έλαβαν μέρος μεταξύ των A και W.

Έστω λοιπόν ότι έχουμε το παρακάτω event episode list ( EEL ):

$$\begin{aligned} \{A,W\} &\rightarrow (e - p - d - s) \\ \{B,W\} &\rightarrow (e - p - t - d) \\ \{C,W\} &\rightarrow (a - o - e - p - t) \end{aligned}$$


$$\begin{aligned} \{D,W\} &\rightarrow (a - o - e - p - d - s) \\ \{E,W\} &\rightarrow (p - d - s) \\ \{F,W\} &\rightarrow (o - e - p - d - s) \\ \{G,W\} &\rightarrow (a - o - e) \\ \{H,W\} &\rightarrow (a - o - e - p) \\ \{I,W\} &\rightarrow (a - d) \\ \{J,W\} &\rightarrow (a - o - e - p - t - d) \end{aligned}$$

Για τον σχηματισμό των παραπάνω ο περιορισμός είναι το HL να είναι ίσο με 6. Τα γράμματα μέσα στα άγκιστρα αναπαριστούν τους πράκτορες που αλληλεπιδρούν και αυτά μέσα στις παρενθέσεις τις ενέργειες που λαμβάνουν μέρος στη μεταξύ τους αλληλεπίδραση.

### 6.2.4 Special Event Episode List ( SEEL ) και την Normal Event Episode List ( NEEL )

Όπως έχει πάλι αναφερθεί κάποιες EEL έχουν κυρώσεις σαν ένα από τα γεγονότα τους. Ο πράκτορας γνωρίζει ποια γεγονότα αποτελούν sanction γεγονότα αφού αυτά συμπεριλαμβάνονται στο **Special Events Set (SES)**. Με βάση την λίστα EEL, ο πράκτορας σχηματίζει δύο λίστες, όπως αναφέρεται και παραπάνω. Η μία λίστα είναι αυτή που περιέχει και sanctioning events ενώ η δεύτερη δεν περιέχει. Οι λίστες αυτές ονομάζονται, όπως έχουμε πει, Special Event Episode List ( SEEL) και Normal Event Episode List (NEEL).

Για το παραπάνω EEL οι δύο λίστες που προκύπτουν είναι :

**SEEL:** (e - p - d - s)  
(a - o - e - p - d - s)  
(o - e - p - d - s)  
(p - d - s)

**NEEL:** (e - p - t - d)  
(a - o - e - p - t)  
(a - o - e)  
(a - o - e - p)  
(a - d)  
(a - o - e - p - t - d)

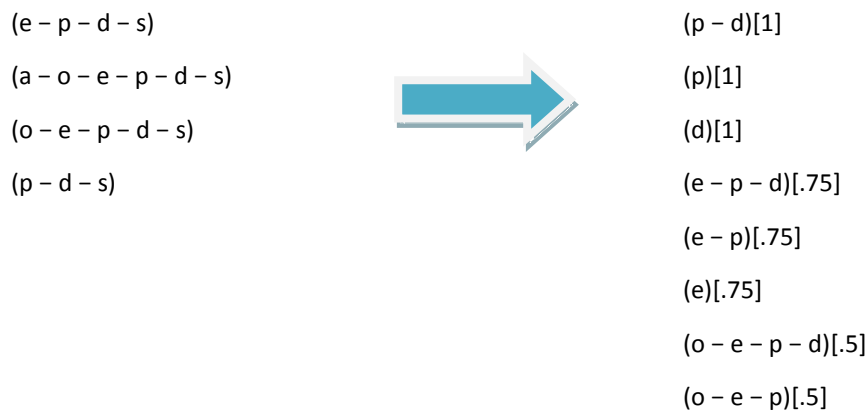
### 6.2.5 Norm - Related Event Set ( NREES )



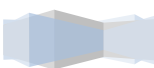
Από την SEEL ο πράκτορας, βλέποντας τη σειρά των γεγονότων μπορεί να συμπεράνει ποια είναι αυτά που κατά κάποιον τρόπο οδηγούν σε τιμωρία. Βλέποντας την παραπάνω SEEL λίστα, θα μπορούσαμε να συμπεράνουμε ότι η ακολουθία των γεγονότων  $p - d$  ή μεμονωμένα τα γεγονότα  $p$  ή  $d$  οδηγούν σε sanction (  $s$  ). Σε αντίθεση με το τι συμβαίνει με την αναγνώριση νορμών απαγόρευσης, κατά τη διαδικασία αναγνώρισης νορμών υποχρέωσης, αυτό που έχει σημασία να παρατηρήσει ο πράκτορας – παρατηρητής είναι οι σειρές των γεγονότων που συμβαίνουν με μεγαλύτερη συχνότητα πριν το γεγονός sanction. Ο πράκτορας αποθηκεύει τις ακολουθίες των γεγονότων με τη μεγαλύτερη συχνότητα στο **Norm – Related Event Set ( NREES )**.

Έστω ότι αναφερόμαστε σε ένα event episode του NREES σαν  $\alpha$ . Σε δεύτερη φάση ο πράκτορας πρέπει αν απομονώσει όλες τις επιμέρους σειρές του  $\alpha$  που βρίσκονται και στο NEEL με πιθανότητα εμφάνισης μεγαλύτερη ή ίση του  $NIT_{\alpha}$  τα οποία και θα συμπεριληφθούν στις υποψήφιες νόρμες υποχρέωσης ( **Candidate Obligation Norm Set – CONS** ) . Για να προσδιορίσει τα γεγονότα που σχετίζονται με τις νόρμες, ο πράκτορας, χρησιμοποιεί τον αλγόριθμο CNL. Βάσει αυτού, αναγνωρίζονται υποψήφιες νόρμες . Βασισμένοι στην SEEL, ο πράκτορας παράγει το NREES.

Παρακάτω παρουσιάζεται αυτή η διαδικασία με ένα παράδειγμα.



Στην αριστερή μεριά φαίνεται η SEEL και στα δεξιά του τόξου το παραγόμενο NREES όταν το  $NIT_{\alpha}$  είναι 0. Η πιθανότητα ενός EE στο NREES δίνεται μέσα στις τετραγωνισμένες αγκύλες. Όταν το  $NIT_{\alpha}$  είναι 0 παράγονται όλες οι πιθανές υπακολουθίες των EE στο SEEL. Όταν είναι ίσο με 100, ο αλγόριθμος καταλήγει στο NREES {  $p - d, p, d$  }.



## 6.2.6 Candidate Obligation Norm Set ( CONS )

Για να καταλήξει στη νόρμα υποχρέωσης, ο πράκτορας πρέπει πρώτα να προσδιορίσει αυτές τις «υπεραλληλουχίες» στη NEEL, που εμπεριέχουν τα EE στο NREES. Οι υπεραλληλουχίες αυτές αποθηκεύονται σε λίστα ( **tempEEList** ).

Βάσει των υπεραλληλουχιών αυτών οι οποίες είναι αποθηκευμένες στην tempEEList, ο αλγόριθμος PNI μπορεί να προσδιορίσει όλες τις μεταθέσεις των υπεραλληλουχιών των οποίων οι πιθανότητα εμφάνισης είναι μεγαλύτερη ή ίση του NITb . Αυτές οι υπεραλληλουχίες αποθηκεύονται στο CONS.

Ο αντίστοιχος αλγόριθμος παρουσιάζεται παρακάτω.

```
Input: Norm-Related Event Episode Set (NREES), Normal Event Episode List  
(NEEL), Norm Identification Threshold (NITb)  
Output: Candidate Obligation Norm Set (CONS)  
1  CONS ← 0;  
2  for each event episode NREE 2 NREES do  
3      tempEEList ← 0;  
4      foreach event episode EE ∈ NEEL do  
5          if EE is a supersequence of NREE then  
6              Add EE to tempEEList;  
7          end  
8      end  
9      Extract all candidate obligation norms (Input: tempEEList, Unique Event Set (UES), Window  
10     Size(WS), Norm Inference Threshold (NITb), Output: Candidate norms);  
11     Add candidate obligation norms to CONS;  
12 end  
13 return CONS;
```

Αλγόριθμος 6.2 Candidate Obligation Norm ( CONS) αλγόριθμος

Αν υποθέσουμε ότι το EE p – d είναι το μοναδικό EE στο NREES τότε στο παρακάτω παράδειγμα φαίνεται η NEEL στα αριστερά και η tempEEList που παράγεται από αυτή στα δεξιά.

(e – p – t – d)

(a – o – e – p – t)

(a – o – e)

(a – o – e – p)

(a – o – d)

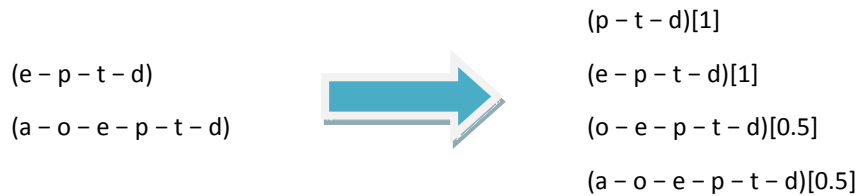
(a – o – e – p – t – d)



(e – p – t – d)

(a – o – e – p – t – d)

Από την tempEEList μπορούμε να παράγουμε το CONS. Στα παρακάτω αριστερά είναι η tempEEList και δεξιά βρίσκονται όλες οι παραλλαγές των «υπερσειρών» των  $p - d$  που μπορούν να ληφθούν από την tempEEList καθώς και τις πιθανότητες εμφάνισής τους.



### 6.2.6 Επιβεβαίωση Νόρμας (Norm Verification)

Ο μηχανισμός επιβεβαίωσης της Νόρμας Υποχρέωσης είναι ίδιος με αυτόν που αναφέρθηκε και για τη Νόρμα Απαγόρευσης. Ο πράκτορας περιοδικά ρωτάει στη γειτονιά του κάποιον πράκτορα ώστε εκείνος να του επιβεβαιώσει την ορθότητα μιας υποψήφιας νόρμας για το συγκεκριμένο περιβάλλον. Αν η απάντηση είναι θετική, ο πράκτορας αποθηκεύει τη νόρμα αυτή στις ισχύουσες νόρμες. Αν η απάντηση είναι αρνητική, ο πράκτορας προχωρά προς επιβεβαίωση της επόμενης υποψήφιας νόρμας του. Η διαδικασία αυτή συνεχίζεται έως ότου είτε ο πράκτορας βρει μία νόρμα είτε δε βρει καμία νόρμα οπότε και μπορεί να τροποποιήσει τις παραμέτρους των μηχανισμών του.



# 7

## Νόρμες Υπό προϋποθέσεις (Conditional Norms)

Στο κεφάλαιο αυτό εξετάζονται οι νόρμες που ισχύουν υπό κάποιες προϋποθέσεις. Μέχρι στιγμής οι νόρμες απαγόρευσης και υποχρέωσης εφαρμόζονταν σε όλους τους πράκτορες του περιβάλλοντος που εκείνες ίσχυαν, χωρίς να υπάρχει διαχωρισμός ή κάποια προϋπόθεση. Αυτό που ξεχωρίζει τις conditional norms από τις prohibition και τις obligation norms είναι το γεγονός ότι η απαγόρευση ή η υποχρέωση που επιβάλλουν είναι υπό κάποιες προϋποθέσεις.

Αυτό μπορεί να γίνει πιο κατανοητό μέσα από το παρακάτω παράδειγμα. Μια conditional norm θα μπορούσε να μην απαγορεύει τον πράκτορα να ρυπάνει το πάρκο αλλά αν εκείνος απείχε πάνω από 10 μέτρα από τον κάδο απορριμμάτων η απαγόρευση αυτή δε θα ίσχυε.

Βασισμένοι σε αυτή τη λογική, οι πράκτορες που καλούνται να «υπακούσουν» σε conditional norms θα πρέπει όχι μόνο να λαμβάνουν υπ' όψιν τους τις νόρμες αυτές καθαυτές αλλά και τις προϋποθέσεις κάτω από τις οποίες αυτές ισχύουν. Αυτό αποτελεί μία αρκετά δύσκολη διαδικασία για να συμπεράνει ο πράκτορας, μιας που τα όρια μεταξύ του τι θεωρείται επιτρεπτό ή όχι είναι αρκετά δύσκολο να προσδιοριστούν από εκείνον. Π.χ. ο πράκτορας μπορεί να παρατηρεί ότι αν ένας άλλος πράκτορας ρυπάνει 25 μέτρα μακριά από τον κάδο απορριμμάτων τιμωρείται. Αν εκείνος ρυπάνει στα 30 μέτρα μακριά από τον κάδο γιατί θεωρεί ότι είναι επιτρεπτό και τιμωρηθεί (γιατί στην πραγματικότητα η νόρμα μπορεί να υπαγορεύει ότι ο πράκτορας δεν πρέπει να ρυπάνει αν βρίσκεται σε απόσταση μέχρι το πολύ 50 μέτρα από τον κάδο) θα επηρεαστεί αρνητικά.

### 7.1 Αναγνωρίζοντας νόρμες υπό προϋποθέσεις (Conditional Norms)

Κατά τη διάρκεια αναγνώρισης νορμών υπό προϋποθέσεις, η διαδικασία περιλαμβάνει δύο φάσεις. Η πρώτη είναι η αναγνώριση αυτής καθαυτής της νόρμας, είτε αυτή είναι νόρμα απαγόρευσης (prohibition norm) είτε αυτή είναι νόρμα υποχρέωσης (obligation norm). Κατά τη δεύτερη φάση διευκρινίζονται από τον πράκτορα οι προϋποθέσεις κάτω από τις οποίες ισχύει η νόρμα. Οι προϋποθέσεις διευκρινίζονται σύμφωνα με τα συμφραζόμενα, δηλαδή κάθε φορά π.χ. που ο πράκτορας παρατηρεί κάποιον άλλο πράκτορα να ρυπάνει το πάρκο συγκρατεί την απόσταση από τον κάδο απορριμμάτων ώστε όταν συγκεντρώσει αρκετές παρατηρήσεις να συμπεράνει από ποια απόσταση και μετά το να ρυπάνει το πάρκο δεν τιμωρείται. Οι επιπλέον πληροφορίες π.χ. για την απόσταση κ.α. μπορεί να είναι είτε μονοδιάστατες ποσότητες είτε πολυδιάστατες. Ο αλγόριθμος, για κάθε νόρμα στο σύνολο των νορμών, Norm Set ( NS ), υπολογίζει αυτές τις «πολυδιάστατες» πληροφορίες βασισμένος σε όσα βλέπει ο πράκτορας μέσα στο πεδίο παρατήρησής του. Ο πράκτορας

σύμφωνα με τον παρακάτω αλγόριθμο μπορεί να ρωτήσει τους πράκτορες της γειτονιάς του σχετικά με το κατά πόσο ισχύουν οι προϋποθέσεις αυτές ( γραμμή 9, 10).

Ακολουθεί ο αλγόριθμος που υπολογίζει τη νόρμα υπό προϋποθέσεις .

```
Input: Contextual Conditions
1 valueOfContextualCondition []  $\leftarrow$  0;
2 conditionalNormReferralConsidered  $\leftarrow$  true;
3 conditionalNormRecommenders  $\leftarrow$  0 ;
4 foreach norm inference cycle do
5     Obtain Norms Set (NS) ; /* By invoking Candidate Norm
6     Identification algorithm */
7     if NS  $\neq$  0 ; then
8         foreach norm n in NS do
9             valueOfContextualCondition [n]  $\leftarrow$ 
10            calculateContextualConditionalValue ; /* This is calculated based on the
11            available data on all punished agents within the visibility threshold
12            */
13            if conditionalNormReferralConsidered then
14                conditionalNormRecommenders  $\leftarrow$  getAgentsFromVicinity;
15                foreach agent 2 conditionalNormRecommenders do
16                    if agent.valueOfContextualCondition is better than
17                    valueOfContextualCondition then
18                        valueOfContextualCondition
19                        agent.valueOfContextualCondition;
20                end
21            end
22        end
23    end
24 end
```

#### Αλγόριθμος 7.1 Conditional Norm

Έτσι, μόλις ο πράκτορας συμπεράνει τη νόρμα συμπεραίνει και τις προϋποθέσεις κάτω από τις οποίες ισχύει η νόρμα.



# 8 Περιγραφή Υλοποίησης Αλγορίθμου

Στα πλαίσια της παρούσας μεταπτυχιακής εργασίας εξετάσθηκε η αρχιτεκτονική, βασισμένος στην οποία, ένας πράκτορας που δρα μέσα σε ένα πολυπρακτορικό περιβάλλον μπορεί να συμπεράνει νόρμες που διέπουν το περιβάλλον αυτό χωρίς να έχει πρότερη γνώση, αλλά απλά βασισμένος στα γεγονότα που παρακολουθεί να εξελίσσονται μπροστά του. Η αρχιτεκτονική αυτή προτάθηκε από τους Bastin Tony Roy Savarimuthu, Stephen Cranefield, Maryam A. Purvis και Martin K. Purvis στο επιστημονικό έγγραφο « Norm Identification in Multi – agent Societies» το 2011. Η αρχιτεκτονική αυτή δίνει απάντηση στο ερώτημα «πως ο πράκτορας μπορεί να συμπεράνει τις νόρμες που διέπουν ένα σύστημα », ένα πεδίο που μέχρι τότε δεν είχε τύχει ιδιαίτερης προσοχής στα πλαίσια των πολυπρακτορικών συστημάτων που διέπονται από νόρμες.

Η αρχιτεκτονική αναγνώρισης νορμών βασίζεται στον αλγόριθμο Candidate Norm Inference (CNI) ο οποίος και υλοποιήθηκε στα πλαίσια αυτής της εργασίας. Ο αλγόριθμος CNI αποτελεί μια τροποποιημένη έκδοση του αλγορίθμου εξόρυξης κανόνων WINEPI. Βάσει αυτού, ο πράκτορας είναι σε θέση εισερχόμενος σε ένα περιβάλλον, τις νόρμες του οποίου δε γνωρίζει *a priori*, να τις συμπεράνει. Επίσης μπορεί τροποποιώντας παραμέτρους του αλγορίθμου, και αυξάνοντας έτσι την προσαρμοστικότητά του σε ένα περιβάλλον το οποίο δεν είναι στατικό, να συμπεράνει εκ νέου νόρμες που το διέπουν ανά πάσα στιγμή.

Για την υλοποίηση χρησιμοποιήθηκε η Object και Component-oriented γλώσσα C Sharp ( C# ), η οποία είναι strong-typed και δηλωτική γλώσσα. Αναπτύχθηκε στην πλατφόρμα .NET του Microsoft Visual Studio 2012.

## 8.1 Υλοποίηση Αλγορίθμου

Όπως αναφέρθηκε και παραπάνω, κατά την περιγραφή της λογικής της αρχιτεκτονικής, ο πράκτορας, παρακολουθώντας το περιβάλλον του, αποθηκεύει τις ακολουθίες των γεγονότων που παρατηρεί να εξελίσσονται κατά τη διάρκεια της αλληλεπίδρασης μεταξύ των υπόλοιπων πρακτόρων μέσα στην belief base του. Από εκεί, αντλώντας την ακολουθία, πια, των γεγονότων (Event Sequences), κατασκευάζονται τα Event Episodes, οι μικρότερου μήκους ακολουθίες, δηλαδή ,των γεγονότων, μήκους τόσο όσο να εξασφαλιστεί η παραγωγή νόρμας. Παρακάτω παρουσιάζεται με μεγαλύτερη λεπτομέρεια η διαδικασία και η αντιστοιχία της με την υλοποίηση.



Για την προσομοίωση της διαδικασίας, αρχικά, υλοποιήθηκε μία κλάση στην οποία πραγματοποιείται ο έλεγχος για την ορθότητα των ακολουθιών των γεγονότων που λαμβάνει ο πράκτορας. Οι ακολουθίες των γεγονότων που παρατηρεί ο πράκτορας αποτελούνται από τα ίδια τα γεγονότα διαχωρισμένα με κάποιο delimiter χαρακτήρα.

Από τη στιγμή που ο πράκτορας προσλάβει την ακολουθία των γεγονότων, την αποθηκεύει στην belief base του. Στην υλοποίηση του αλγορίθμου θεωρήθηκε ότι οι βάσεις δεδομένων του πράκτορα μπορεί να υλοποιηθούν με την κλάση List<T> , η οποία είναι μία δυναμική δομή στην οποία μπορεί να προστίθενται τα γεγονότα με τη σειρά που τα «αντιλαμβάνεται» ο πράκτορας.

Όσον αφορά την Special Events Base του πράκτορα, έχει και αυτή υλοποιηθεί με δυναμική λίστα της κλάσης List<T>. Εδώ, έχει θεωρηθεί ότι ο πράκτορας γνωρίζει εκ των προτέρων ποια γεγονότα αποτελούν special events, δηλαδή γεγονότα που προκαλούν , στην προκειμένη περίπτωση, την τιμωρία του πράκτορα. Τα special events περιλαμβάνουν και γεγονότα που προκαλούν την επιβράβευση του πράκτορα, αλλά στην προκειμένη ασχολούμαστε μόνο με γεγονότα που οδηγούν στο συμπερασμό της νόρμας έπειτα από αποδοκιμασία του πράκτορα.

Το επόμενο στάδιο αποτελείται από τις διαδικασίες που συνθέτουν το πλαίσιο του αλγορίθμου για την αναγνώριση των νορμών, και η οποία περιγράφεται συνοπτικά στο διάγραμμα της εικόνας 8.1.

Σύμφωνα λοιπόν και με το παρακάτω διάγραμμα ροής, έχουν υλοποιηθεί οι παρακάτω επιμέρους βοηθητικές συναρτήσεις:

#### ***createUniqueEventSet***

Η συνάρτηση δέχεται σαν είσοδο από την belief base του πράκτορα το event sequence που έχει αποθηκευτεί και τη λίστα με τα special events και δίνει σαν έξοδο τα διακριτά γεγονότα που παρατηρούνται στη διάρκεια παρατήρησης του πράκτορα (History Length).

#### ***createRemovableEventSet***

Η συνάρτηση δέχεται ως είσοδο την σειρά γεγονότων (event sequence) που έχει παρατηρήσει ο πράκτορας, τα διακριτά γεγονότα που απαρτίζουν την ES, καθώς και το Norm Pruning Threshold (NPT). Για κάθε γεγονός υπολογίζεται η τιμή του Occurrence Probability του μέσα στη σειρά των γεγονότων που έχει παρατηρήσει ο πράκτορας. Αν είναι μεγαλύτερη της τιμής του NPT τότε το γεγονός προστίθεται στην Removable Event Set. Σε αυτό εισάγονται, γεγονότα τα οποία είναι σύνηθες να συμβούν και δεν οδηγούν σε sanction event, για παράδειγμα το γεγονός move του παραδείγματος του πάρκου.



### **createPeel**

Η συνάρτηση δέχεται ως είσοδο την ακολουθία των γεγονότων (event Sequence), τα περιεχόμενα της Special Events Base, το Removable Event Set που περιγράψαμε προηγουμένως και το μέγεθος παραθύρου ( ws ) που καθορίζει το μήκος των Event Episodes. Το αποτέλεσμα της συνάρτησης είναι η λίστα Pruning Event Episode List (PEEL) η οποία και περιέχει τις υπακολουθίες των γεγονότων που καταλήγουν σε sanction χωρίς να περιέχονται τα συνήθη γεγονότα που δεν οδηγούν σε εμφάνιση νόρμας.

### **IsInPeel**

Δέχεται σαν είσοδο ένα sub-episode και την λίστα PEEL και ελέγχει αν το sub-episode εμπεριέχεται στην PEEL.

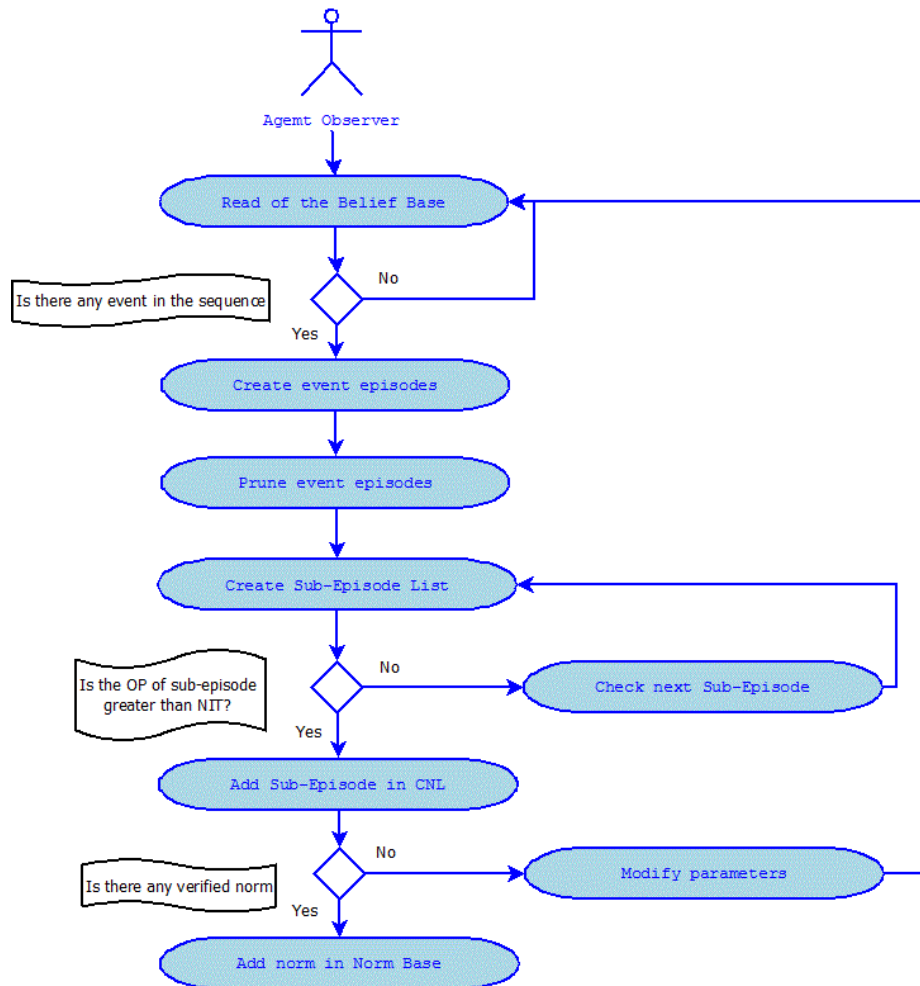


Figure 5 Γενική εικόνα μηχανισμού αναγνώρισης νορμών



### ***occurrenceProbability***

Δέχεται σαν είσοδο ένα sub-episode και τη λίστα PEEL. Ανακαλύπτει πόσες φορές έχει παρατηρηθεί το sub-episode μέσα στη λίστα και έπειτα υπολογίζει την Occurrence Probability του.

### ***isEachEventInCNL***

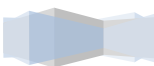
Δέχεται σαν είσοδο ένα sub-episode και τη λίστα Candidate Norm List η οποία περιέχει τις υποψήφιες νόρμες προς επιβεβαίωση και ελέγχει αν το sub-episode περιέχεται στις μέχρι εκείνη τη στιγμή υποψήφιες νόρμες.

### ***createSubEpisodeList***

Αποτελεί έναν από τους βασικότερους αλγορίθμους στη διαδικασία της αναγνώρισης νορμών αφού είναι αυτός που «ευθύνεται» για την κατασκευή της λίστας Sub-Episode List ( SEL ). Σαν είσοδο δέχεται μία λίστα με επεισόδια μήκους  $n$  η οποία αποτελεί το «σπόρο» για την κατασκευή λίστας με sub-episodes μήκους  $n+1$ . Ο αλγόριθμος αποτελεί μία τροποποίηση του πρώτου μέρους του αλγορίθμου WINEPI [18] ο οποίος χρησιμοποιείται για την κατασκευή και ανακάλυψη συχνών επεισοδίων από άλλα μικρότερου μήκους επεισόδια.

Ο αλγόριθμος υπολογίζει μία συλλογή επεισοδίων μήκους  $n$  βασισμένος σε μία συλλογή επεισοδίων μήκους  $n-1$ . Η διαφοροποίηση του αλγορίθμου WINEPI με την τροποποιημένη εκδοχή του που υλοποιείται εδώ είναι ότι μετά τον υπολογισμό της λίστας των νέων sub-episodes, δεν γίνεται «κλάδεμα» των λιγότερο συχνών. Αυτή είναι μία διαδικασία που υλοποιείται σε επόμενο αλγόριθμο. Αρχικά γίνεται μία κατά πλάτος έρευνα στα episodes για να βρεθεί η σχέση μεταξύ των sub-episodes τους, διαδικασία η οποία εξηγείται αναλυτικότερα παρακάτω. Το ψάξιμο ξεκινά από τα πιο γενικά επεισόδια, δηλαδή τα επεισόδια που περιλαμβάνουν μόνο ένα γεγονός (event). Σε κάθε διάτρεξη του συνόλου των επεισοδίων ο αλγόριθμος υπολογίζει ένα σύνολο νέων, υποψήφιων για νόρμες, επεισοδίων.

Πρέπει να σημειωθεί ότι σε κάθε επανάληψη του αλγορίθμου σαν συλλογή επεισοδίων που λειτουργούν ως «σπόροι» για τον αλγόριθμο, χρησιμοποιούνται τα επεισόδια τα οποία έχουν μεγαλύτερη συχνότητα.



## Επεξήγηση Αλγορίθμου

Στον αλγόριθμο, ένα επεισόδιο  $\alpha = (V, \leq, g)$  αναπαριστάται σαν ένας πίνακας που αποτελείται από γεγονότα (events). Ο πίνακας συμβολίζεται από το όνομα του επεισοδίου και τα περιεχόμενα του πίνακα δεικτοδοτούνται μέσα σε τετραγωνισμένες αγκύλες. Έτσι για παράδειγμα, ένα επεισόδιο  $\alpha$  που περιέχει τα γεγονότα A, C, C, και F αναπαριστάται σαν πίνακας ως  $\alpha[1] = A$ ,  $\alpha[2] = C$ ,  $\alpha[3] = C$  και  $\alpha[4] = F$ . Το σύνολο των επεισοδίων αναπαριστάται με το γράμμα  $F[i]$  με  $i$  το  $i$ -οστό επεισόδιο του συνόλου.

Τα επεισόδια ταξινομούνται μέσα στο σύνολο κατά τέτοια σειρά ώστε κάθε φορά αλλάζει μόνο ένα γεγονός στην σειρά των γεγονότων. Εξαιτίας αυτής της «ταξινόμησης» τα επεισόδια που έχουν κοινά τα πρώτα επεισόδια είναι διαδοχικά. Συγκεκριμένα, αν τα επεισόδια  $F_l[i]$  και  $F_l[j]$  μήκους  $l$  έχουν κοινά τα  $l-1$  πρώτα γεγονότα, τότε και όλα τα επεισόδια  $k$  τέτοια ώστε  $i \leq k \leq j$  θα μοιράζονται τα  $l-1$  πρώτα τους επεισόδια. Η μέγιστη ακολουθία από διαδοχικά επεισόδια που έχουν ίδια τα πρώτα  $l-1$  γεγονότα ονομάζεται *block*. Οι πιθανές υποψήφιες νόρμες σχηματίζονται από τους όλους τους συνδυασμούς δύο επεισοδίων στο ίδιο block. Στη μεταβλητή  $F_l.block\_start[j]$  αποθηκεύεται για κάθε επεισόδιο  $F_l[j]$ , με  $i$  τέτοιο ώστε  $F_l[i]$  να είναι το πρώτο επεισόδιο του block. Στο σύνολο  $C$  αποθηκεύονται τα σύνολα των νέων επεισοδίων που θα αποτελέσουν και υποψήφιες νόρμες.

Ο ψευδοκώδικας του αλγορίθμου που υλοποιήθηκε είναι ο παρακάτω:

```
1.   Ci+1 := 0;
2.   k := 0;
3.   if l = 1 then for h := 1 to |Fl| do Fl.block_start[h] := 1;
4.   for i := 1 to |Fl| do
5.       current_block_start := k + 1;
6.       for ( j := Fl.block_start[i]; Fl.block_start[j] = Fl.block_start[i]; j := j + 1 ) do
7.           /* Fl[i] and Fl[j] have l-1 first event types in common,
8.           build a potential candidate a as their combination */
9.           for x := 1 to l do a[x] := Fl[i][x];
10.          a[l+1] := Fl[j][l];
11.
12.          /* All sub-episodes are in Fl, store a as candidate: */
13.          k := k + 1;
14.          Ci+1[k] := a;
15.          Ci+1.block_start[k] := current_block_start;
16.   output Ci+1;
```

### Αλγόριθμος 8.1 Δημιουργία των υποψήφινων επεισοδίων

Η φάση της δημιουργίας των υποψήφινων επεισοδίων – νορμών διαχωρίστηκε από τη φάση όπου από το σύνολο των παραγόμενων επεισοδίων αφαιρούνται τα επεισόδια των οποίων το Occurrence Probability είναι κάτω του ορίου του Norm Identification Threshold (NIT) ώστε να μειωθεί η επεξεργασία που γίνεται σε κάθε πέρασμα της βάσης του πράκτορα.



Η χρονική πολυπλοκότητα του παραπάνω αλγορίθμου είναι πολυωνυμική στο μέγεθος των επεισοδίων και είναι ανεξάρτητη του μήκους των ακολουθιών των γεγονότων. Συγκεκριμένα, η αρχικοποίηση (γραμμή 3) διαρκεί  $O(|F_i|)$ . Το εξωτερικό for loop διατρέχεται  $O(|F_i|)$  φορές και το εσωτερικό  $O(|F_i|)$ . Μέσα στα δύο for loops μία πιθανή υποψήφια νόρμα σχηματίζεται σε χρόνο  $O(l+1)$ . Οπότε η συνολική χρονική πολυπλοκότητα είναι  $O(|F_i| + |F_i| |F_i| l) = O(l |F_i|^2)$

Στον παρακάτω παρουσιάζεται ένα μικρό δείγμα τρεξίματος του αλγορίθμου με ακολουθίες αριθμών ώστε να γίνει ο αλγόριθμος πιο κατανοητός.

Ακολουθίες γεγονότων μήκους 2 ( $F_2$ )	$F_2$ .block_start	Παραγόμενες ακολουθίες γεγονότων μήκους 3 ( $F_3$ )
<(1, 6)>	1	<(1, 6, 6)> <(1, 6, 7)> <(1, 6, 8)>
<(1, 6)>	1	<(1, 7, 7)> <(1, 7, 8)>
<(1, 6)>	1	<(1, 8, 8)>
<(1, 6)>	4	<(7, 8, 8)>

### **createCNL**

Δέχεται ως ορίσματα τη λίστα Pruning Event Episode List (PEEL), το UES, το μήκος του παραθύρου παρατήρησης WS, και το Norm Identification Threshold. Ο αλγόριθμος αυτός υλοποιεί ουσιαστικά τη διαδικασία της περικοπής των υποψήφιας νορμών οι οποίες έχουν πιθανότητα εμφάνισης μικρότερη από την τιμή του Norm Identification Threshold (NIT). Ο ψευδοκώδικας του αλγορίθμου είναι αυτός που παρουσιάζεται στον Αλγόριθμος 5.3. Όπως αναφέρθηκε αναλυτικότερα και παραπάνω ο αλγόριθμος λειτουργεί επαναληπτικά. Ο αριθμός των επαναλήψεων είναι ίσος με το WS.



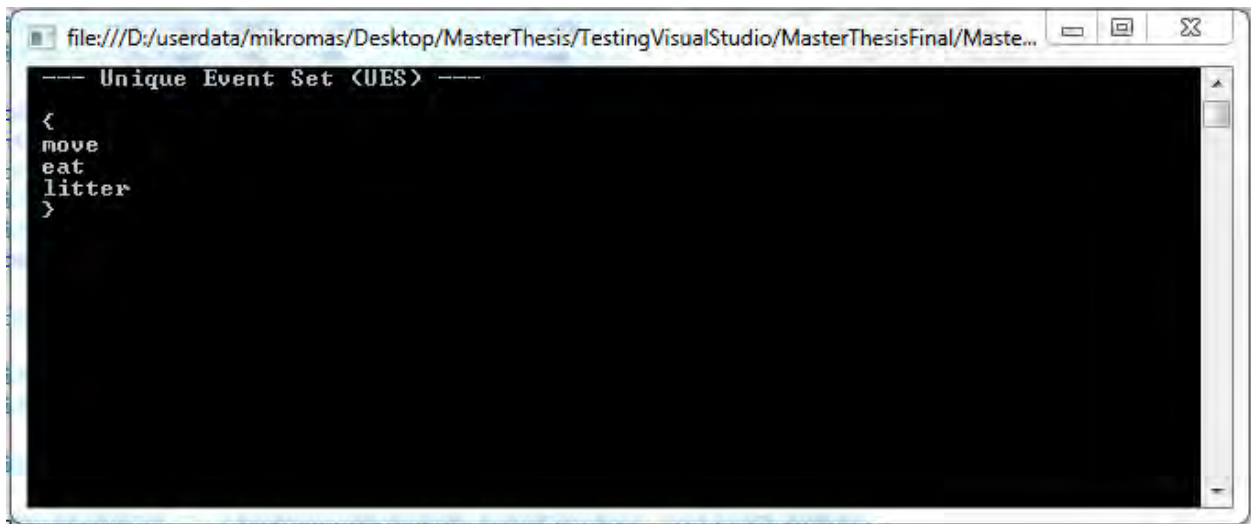
## Παράδειγμα Τρεξίματος του Αλγορίθμου

Στο παράδειγμα που ακολουθεί ο πράκτορας αντιλαμβάνεται την παρακάτω σειρά γεγονότων από το περιβάλλον του :

```
ES = {"move", "move", "move", "move", "eat", "litter", "move", "sanction", "eat", "move",  
"litter", "eat", "litter", "move", "sanction", "move", "move", "eat", "move", "litter",  
"sanction", "eat", "eat", "litter", "eat", "sanction", "move", "eat", "litter", "eat",  
"sanction", "move", "litter", "eat", "litter", "sanction", "eat", "eat", "litter",  
"sanction", "move", "move", "eat", "litter", "sanction", "eat", "eat", "litter", "move",  
"sanction", "move", "eat", "litter", "eat", "sanction", "move", "move", "move", "move",  
"move", "move", "move", "move", "move", "move", "move", "move", "move", "move", "move",  
"move", "move", "move", "move", "move", "move", "move", "move", "move", "move", "move", }
```

Όπως έχει περιγραφεί και παραπάνω, τόσο στη θεωρητική βάση της αρχιτεκτονικής αναγνώρισης νορμών με την οποία ασχολείται αυτή η εργασία, όσο και στην υλοποίησή της προγραμματιστικά θεωρείται ότι ο πράκτορας γνωρίζει εκ των προτέρων το σύνολο των special events , δηλαδή του Special Event Set (SES). Στο προκείμενο παράδειγμα σαν SES έχει θεωρηθεί το γεγονός “sanction”, δηλαδή  $SES = \{ sanction \}$ .

Ο αλγόριθμος υπολογίζει αρχικά από την παραπάνω ακολουθία το Unique Event Set ( UES ).



```
file:///D:/userdata/mikromas/Desktop/MasterThesis/TestingVisualStudio/MasterThesisFinal/Maste...  
---- Unique Event Set (UES) ----  
{  
  move  
  eat  
  litter  
}
```

Figure 6 Το Unique Event Set ( UES ) του παραδείγματος

Κατά την επόμενη φάση του αλγορίθμου, υπολογίζεται το Removable Event Set (RES). Ο υπολογισμός αυτός στηρίζεται στην τιμή OP για το κάθε γεγονός μέσα στο EE.

```
file:///D:/userdata/mikromas/Desktop/MasterThesis/TestingVisualStudio/MasterThesisFinal/Maste...
--- Unique Event Set <UES> ---
{
move
eat
litter
}

OP< move > = 0,53125 < occurrences_in_ES / number_of_events_in_ES > : <51 / 96 >
OP< eat > = 0,197916666666667 < occurrences_in_ES / number_of_events_in_ES > : <
19 / 96 >
OP< litter > = 0,145833333333333 < occurrences_in_ES / number_of_events_in_ES >
: <14 / 96 >

Norm Pruning Threshold = 0,5 =>

--- Removable Event Set <RES> ---
{
move
}
```

Figure 7 OP των γεγονότων του EE και το τελικό RES

Όπως παρατηρούμε και στην εικόνα, το  $OP(eat)$  ισούται με 0.53125, τιμή μεγαλύτερη του 0.5 που έχει οριστεί αρχικά για το NIT. Συνεπώς το γεγονός “move” θα είναι αυτό που θα περικοπεί.

Ύστερα υπολογίζεται η λίστα PEEL. Στο παράδειγμα έχει η τιμή για το WS είναι 3.



```
file:///D:/userdata/mikromas/Desktop/MasterThesis/TestingVisualStudio/MasterThesisFinal/Maste...
----- Pruning Event Episode List <PEEL> -----
{
eat
litter
sanction
}

{
eat
litter
sanction
}

{
eat
litter
sanction
}

{
litter
eat
sanction
}

{
litter
eat
sanction
}

{
eat
litter
sanction
}

{
eat
litter
sanction
}

{
eat
litter
sanction
}

{
eat
litter
sanction
}

{
litter
eat
sanction
}

{
litter
eat
sanction
}

{
litter
eat
sanction
}
```

Figure 8 PEEL

Τέλος υπολογίζεται το σύνολο των υποψήφιων προς επιβεβαίωση νορμών όπως φαίνεται και στην παρακάτω εικόνα.





```

file:///D:/userdata/mikromas/Desktop/MasterThesis/TestingVisualStudio/MasterThesisFinal/Maste...
Candidate Norms List <CNL> ---
{
  eat
}
{
  litter
}
{
  eat
  litter
}

```

Figure 9 Candidate Norm List

Στο παραπάνω αποτέλεσμα καταλήξαμε μετά από 3 επαναλήψεις του αλγορίθμου createCNL. Συγκεκριμένα, οι επαναλήψεις που έλαβαν χώρα έχουν ως εξής :

Sub-Episode που εξετάζεται	OP (SE)	SELtemp	CNL	SELnext
<i>iterNum = 1</i>				
{ eat }	1.0	{ eat }	[[ eat ]]	SELnext = [{eat, eat}, {eat, litter}, {litter, eat}, {litter, litter}]
{ litter }	1.0	{ eat, litter }	[[ eat ], { litter }]	
<i>iterNum = 2</i>				
{eat, eat}	0.0			
{eat, litter}	0.7	{eat, litter}	[[eat], {litter}, {eat, litter}]	SELnext = [[eat, eat, litter]]
{litter, eat}	0.3			
{litter, litter}	0.0			
<i>iterNum = 3</i>				
{eat, eat, litter}	0.0			

Συνεπώς το τελικό σύνολο των νορμών είναι το [{eat}, {litter}, {eat, litter}] όπως φαίνεται και στην εικόνα 8.5 κατά το τρέξιμο του αλγορίθμου.



# 9 Συμπεράσματα & Επεκτάσεις

## 9.1 Συμπεράσματα & Επεκτάσεις της αρχιτεκτονικής για την αναγνώριση νορμών

Σε αυτή τη μεταπτυχιακή εργασία εξετάστηκε η δυνατότητα του πράκτορα παρατηρώντας τους πράκτορες μιας κοινωνίας και μη γνωρίζοντας εξαρχής τις νόρμες που διέπουν αυτό το σύστημα να μπορεί να τις συμπεραίνει. Η συγκεκριμένη αρχιτεκτονική είχε προταθεί το 2011. Μέχρι τότε θεωρείτο κατά βάση ότι ο πράκτορας που δρούσε μέσα σε ένα πολυπρακτορικό περιβάλλον γνώριζε τις νόρμες που διέπουν το σύστημα a priori. Συνήθως, βάσει αυτής της λογικής, ο πράκτορας που δεν ήταν γνώστης των νορμών ενός περιβάλλοντος τις μάθαινε π.χ. από τον πράκτορα που είχε αναλάβει χρέη αρχηγού μέσα στην κοινωνία. Αυτή η «τακτική» όμως υποβόσκει την εξαπάτηση του πράκτορα σε περίπτωση που ο αρχηγός είναι κακόβουλος πράκτορας. Με το μηχανισμό που προτείνεται ο πράκτορας δρα κατανεμημένα για να καταφέρει να επιβεβαιώσει μία νόρμα. Επιπλέον, ο πράκτορας πλέον έχει τη δυνατότητα να προσαρμόζεται στις εκάστοτε συνθήκες που ισχύουν, να «προσαρμόζεται» σε δυναμικά μεταβαλλόμενα περιβάλλοντα. Τροποποιώντας τις παραμέτρους των αλγορίθμων του όποτε εκείνος θεωρεί ότι πρέπει να γίνει, είτε γιατί δεν κατάφερε να συμπεράνει μια νόρμα είτε γιατί διαπίστωσε ότι οι συνθήκες του περιβάλλοντος έχουν αλλάξει, καταλήγει στο να συμπεράνει τη νόρμα που ισχύει το συγκεκριμένο χρονικό διάστημα. Επιπρόσθετα, μπορεί να προσθέτει στη βάση δεδομένων του νόρμες, να τις αφαιρεί ή και να τις τροποποιεί ώστε οι νόρμες που διαθέτει να ισχύουν στο περιβάλλον του. Η προσαρμοστικότητα αυτή μπορεί να βρει εφαρμογή όχι μόνο σε στατικά περιβάλλοντα, όπως ίσχυε μέχρι τώρα, αλλά και σε δυναμικά, συνεχώς μεταβαλλόμενα.

Όσον αφορά την υλοποίηση του της αρχιτεκτονικής, όπως ειπώθηκε και παραπάνω, έγινε με την αντικειμενοστραφή γλώσσα C#. Αυτό που προστέθηκε επιπλέον μέσα σε αυτή τη διατριβή όσον αφορά το θεωρητικό επίπεδο της αρχιτεκτονικής, ήταν η κατασκευή και χρήση του αλγορίθμου που οδηγεί στη δημιουργία της λίστας των Sub-Episodes, διαδικασία η οποία δεν είχε διευκρινιστεί αρχικά. Ο αλγόριθμος περιγράφηκε αναλυτικά στο προηγούμενο κεφάλαιο.

Όσον αφορά τη μελλοντική έρευνα και υλοποίηση πάνω στην αναγνώριση νορμών, και πιο συγκεκριμένα, οι επεκτάσεις που θα μπορούσαν να γίνουν πάνω σε αυτή την αρχιτεκτονική, αφορούν στην αναγνώριση νορμών οι οποίες σχεδιάζονται έτσι ώστε να αποφευχθούν δυσάρεστες καταστάσεις στην κοινωνία, δηλαδή όταν η κοινωνία φτάνει σε αδιέξοδο ή όταν δεν μπορεί να επιτευχθεί αποσυμφόρηση της κατάστασης μεταξύ πράκτορα που τιμωρεί και εκείνου που τιμωρείται. Επιπλέον μπορεί να μελετηθεί περεταίρω η επιβεβαίωση της νόρμας όσον αφορά την επέκταση της γειτονιάς

στην οποία απευθύνεται ο πράκτορας. Άλλος ένας τομέας είναι αυτός του πεδίου παρατήρησης του πράκτορα. Στην παρούσα διπλωματική υποθέτουμε ότι ο πράκτορας παρατηρεί τις αλληλεπιδράσεις των υπόλοιπων πρακτόρων μέσα σε ένα περιορισμένο πεδίο γύρω από τον πράκτορα παρατηρητή. Επιπρόσθετα, στην παρούσα διπλωματική υποθέτεται ότι ο πράκτορας – παρατηρητής έχει τη δυνατότητα να διαχωρίζει τα special events από τα regular αφού τα γνωρίζει εκ των προτέρων. Συνεπώς άλλη μία επέκταση θα μπορούσε να είναι ο πράκτορας να έχει τη δυνατότητα να κρίνει και να διαχωρίζει τα γεγονότα τη στιγμή που αυτά συμβαίνουν, χωρίς να έχει a priori γνώση.

## 9.2 Αρχιτεκτονικές Αναγνώρισης Νορμών

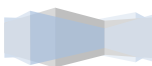
Από τη στιγμή που παρουσιάστηκε η συγκεκριμένη αρχιτεκτονική για την αναγνώριση νορμών σε πολυπρακτορικά περιβάλλοντα έχουν προταθεί αρχιτεκτονικές για το ίδιο πεδίο που ακολουθούν διαφορετικές προσεγγίσεις.

- **Αλγόριθμος LION** : Είναι ένας on-line αλγόριθμος και αποσκοπεί στη σύνθεση liberal normative systems. Ο αλγόριθμος σέβεται την αυτονομία των πρακτόρων προς το μεγαλύτερο δυνατό βαθμό και περιορίζει τη συμπεριφορά τους όταν μόνο είναι απαραίτητο για την αποφυγή ανεπιθύμητων καταστάσεων του συστήματος.
- **Norm Identification through Plan Recognition** : Ο αλγόριθμος αυτός στηρίζεται όπως και ο αλγόριθμος που παρουσιάζεται σε αυτή την εργασία στην παρατήρηση. Η βασική διαφορά και το πλεονέκτημά του είναι ότι, σε αντίθεση με τον αλγόριθμο που παρουσιάστηκε, η παρατήρηση στηρίζεται όχι μόνο σε γεγονότα που παραβιάζουν τις υπάρχουσες νόρμες αλλά και σε συμβατικά γεγονότα. Έτσι ο αλγόριθμος μπορεί να συμπεράνει νόρμες ακόμα και όταν δεν παρατηρούνται sanction events μέσα στο περιβάλλον.



## Βιβλιογραφία – Πηγές

- 1 Michael Wooldridge, **Εισαγωγή στα πολυπρακτορικά συστήματα** (μετάφραση – επιστημονική επιμέλεια ελληνικής έκδοσης : Ασπασία Δασκαλοπούλου, Επίκουρος Καθηγήτρια Πανεπιστημίου Θεσσαλίας )
- 2 Marina De Vos, Nicoletta Fornara, Jeremy V. Pitt, George Vouros, **Coordination, Organizations, Institutions, and Norms in Agent Systems VI**, COIN 2010 International Workshops, AAMAS 2010
- 3 Giulia Andrighetto, Daniel Villatoro, Rosaria Conte, **Norm internalization in artificial societies**, LABSS, Institute of Cognitive Science and Technologies, CNR, Rome, Italy & Artificial Intelligence Research Institute (IIIA), Spanish National Research Council, Spain
- 4 Mark d’Inverno , Michael Luck , Pablo Noriega , Juan A. Rodriguez-Aguilar , Carles Sierra, **Artificial Intelligence, Communicating open systems**, Department of Computing, Goldsmiths, University of London, Department of Informatics, King’s College London, Artificial Intelligence Research Institute, Spanish National Research Council, Campus de la Universitat Autònoma de Barcelona, 2012
- 5 Bastin Tony Roy Savarimuthu, Stephen Cranefield, Maryam A. Purvis, Martin K. Purvis, **Norm Identification in Multi – agent Societies**, Department of Information Science, University of Otago, The Information Science Discussion Paper Series, February 2010
- 6 Stuart Russell, Peter Norvig, **Artificial Intelligence: A Modern Approach**, Third Edition
- 7 Martha Finnemore, Kathryn Sikkink, **International Norm Dynamics and Political Change**, IO Foundation and the Massachusetts Institute of Technology, 1998
- 8 Stephen Cranefield, Tony Savarimuthu, Felipe Meneguzzi, Nir Oren, **A Bayesian Approach to Norm Identification** , Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015
- 9 Javier Morales, Maite Lopez-Sanchez, Juan A. Rodriguez-Aguilar, Michael Wooldridge, Wamberto Vasconcelos, **Synthesizing Liberal Normative Systems**, Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015
- 10 Natasha Alechina, Nils Bulling, Brian Logan, Mehdi Dastani, **Practical Run-Time Norm Enforcement with Bounded Lookahead**, Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015
- 11 Samhar Mahmoud, Simon Miles, Adel Taweel, Brendan Delaney, Michael Luck, **Norm Establishment Constrained by Limited Resources**, Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015
- 12 Timothy J. Norman, Daniela V. Carbogim, Erik C. W. Krabbe, Douglas N. Walton, **Argument and Multi-Agent Systems**, University of Aberdeen
- 13 Carl Hewitt, **Offices are open systems**, ACM Transactions on Information Systems, 1986
- 14 Michael Luck, Onn Shehory, Steve Willmott, Peter McBurney, **Agent technology: computing as interaction ( a roadmap for agent based computing )**, University of Southampton, 2005
- 15 Thomas R. Gruber, **A Translation Approach to Portable Ontology Specifications**, Computer Science Department Stanford University Stanford, California, 1993
- 16 Jürgen Habermas, **Between Facts and Norms**, 1992
- 17 Karl Tuyls, Gerhard Weiss, **Multiagent Learning: Basics, Challenges, and Prospects**, AI Magazine, 2012
- 18 Jussi Ahola, **Mining Sequential Patterns**, VTT Information Technology, 2001



## Παράρτημα Α

Ακρωνύμια	
CONS	Candidate Obligation Norm Set
CPNS	Candidate Prohibition Norm Set
EEL	Event Episode List
ES	Event Sequences
HL	History Length
NEEL	Normal Event Episode List
NIT	Norm Identification Threshold
NIF	Norm Inference Frequency
OP	Occurrence Probability
SEE	Special Event Episode
SEEL	Special Event Episode List
SEL	Sub Episode List
SES	Special Event Set
UES	Unique Event Set
WS	Window Size

Πίνακας Α.1 : Ακρωνύμια

