



## Ευχαριστίες

*Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας θα ήθελα να ευχαριστήσω την κα. Δασκαλοπούλου Ασπασία αρχικά για την εμπιστοσύνη που μου έδειξε με την ανάθεση της εργασίας αυτής και στη συνέχεια για τη στήριξη και την καθοδήγηση της.*

*Επιπλέον θα ήθελα να ευχαριστήσω τον κ. Ακρίτα Αλκιβιάδη που δέχτηκε να αναλάβει την επίβλεψη της διπλωματικής μου εργασίας.*

*Τέλος οφείλω ένα μεγάλο ευχαριστώ στους γονείς μου και στην αδερφή μου που ήταν πάντα δίπλα μου όλα αυτά τα χρόνια καθώς επίσης και στους φίλους μου για την συμπαράσταση και την πίστη που μου έδειξαν σε όλη τη διάρκεια των σπουδών μου.*

*Ζησιού Ελένη*

*Βόλος, Σεπτέμβριος 2012*

## Περίληψη

Η Τροπική λογική είναι ένα είδος τυπικής λογικής η οποία επεκτείνει την κλασική προτασιακή και κατηγορηματική λογική, συμπεριλαμβάνοντας τελεστές εκφρασμένους ως 'modalities'.

Σκοπός της παρούσας διπλωματικής εργασίας είναι να μελετήσουμε τη λογική αυτή ως προς τη σύνταξη και τη σημασιολογία, να εκφράσουμε τα βασικά συστήματα αξιωμάτων και τέλος να παρατηρήσουμε τις διαφορετικές περιοχές αυτής της λογικής ανάλογα με τις ερμηνείες των 'modalities'.

Πιο συγκεκριμένα αρχικά αναφερθήκαμε σε μερικούς βασικούς ορισμούς που θα μας βοηθήσουν να αντιληφθούμε την Τροπική λογική και τους τελεστές της και στη συνέχεια αναπτύξαμε δύο εφαρμογές που αφορούν τον Κόσμο των Κύβων στο περιβάλλον SWI-Prolog, στις οποίες εφαρμόσαμε τα αξιώματα και τους τελεστές που ορίζονται από τα πλαίσια ορθολογικών πρακτόρων BDI και Cohen & Levesque.

## Abstract

Modal logic is a type of formal logic that extends propositional and predicate logic, including operators expressed as ‘modalities’.

The scope of this thesis is to study Modal logic and examine its syntax and semantics. Furthermore critical part of this thesis is the observation of the different subfields, according to the basic systems of axioms and ‘modalities’.

Specifically, we analyzed some basic definitions which will help us to understand the Modal logic and its operators. In addition we developed two applications, based on the ‘Blocks World’, using the SWI-Prolog platform as a tool to apply the axioms and the operators which are defined by the BDI and Cohen & Levesque frameworks.

*Αυτή η σελίδα είναι σκόπιμα κενή.*

## Πίνακας Περιεχομένων

<b>1 Εισαγωγή</b> .....	<b>7</b>
1.1 Τροπική Λογική και Νοήμονες Πράκτορες.....	7
1.2 Μερικές Ερμηνείες των Τροπικών Τελεστών.....	7
1.2.1 Epistemic Logic.....	7
1.2.2 Temporal Logic.....	8
1.2.3 Logic of Agency.....	8
1.3 Οργάνωση Κειμένου.....	8
<b>2 Θεωρητικό Υπόβαθρο</b> .....	<b>10</b>
2.1 Εισαγωγή στη Σύνταξη της Τροπικής Λογικής.....	10
2.2 Kripke Semantics.....	11
2.2.1 Ορισμός της Αλήθειας.....	12
2.2.2 Εγκυρότητα και Πλαίσια.....	14
2.3 Soundness and Completeness.....	16
2.3.1 The Canonical Model Method.....	17
<b>3 Συστήματα Τροπικής Λογικής</b> .....	<b>18</b>
3.1 Normal Systems.....	18
3.1.1 Κανόνας Αντικατάστασης και Δυϊσμός.....	19
3.2 The Schemas and their Combinations.....	19
<b>4 Υλοποίηση και Κριτική Ανάλυση</b> .....	<b>21</b>
4.1 BDI Logic.....	21
4.1.1 Πρώτη Εφαρμογή – Κόσμος των Κύβων.....	22
4.1.2 Εκτέλεση Πρώτης Εφαρμογής.....	24
4.2 Intention Logic of Cohen and Levesque.....	28
4.2.1 Δεύτερη Εφαρμογή – Κόσμος των Κύβων.....	28
4.2.2 Εκτέλεση Δεύτερης Εφαρμογής.....	30
4.3 Σχετικά Με Την Υλοποίηση.....	32
<b>5 Επίλογος</b> .....	<b>33</b>
5.1 Σύνοψη και Συμπεράσματα.....	33
5.2 Μελλοντικές Επεκτάσεις.....	34
<b>6 Βιβλιογραφία</b> .....	<b>36</b>

# 1 Εισαγωγή

---

## 1.1 Τροπική Λογική και Νοήμονες Πράκτορες

Η λογική έχει τις ρίζες της στη Φιλοσοφία και εκφράζει τις αρχές του έγκυρου συλλογισμού και συμπερασμού. Μέσω λογικών συστημάτων οι αρχές αυτές βρίσκουν εφαρμογή στην Πληροφορική. Πιο συγκεκριμένα το σύστημα Τροπικής λογικής αφορά τη μελέτη της διαδικασίας συλλογισμού των 'modalities' (που μπορεί να εκφράζουν το χρόνο, τη γνώση, την πεποίθηση, την επιθυμία ή και την πρόθεση ενός πράκτορα) η οποία καταλήγει, υπακούοντας σε τροπικές παραδοχές, στο συμπερασμό ότι τελικά το τροπικό αποτέλεσμα είναι έγκυρο. Οι 'modalities' αυτές αναπαριστώνται στον προγραμματισμό πρακτόρων ως διανοητικές καταστάσεις (mental state) που υποδεικνύουν την πορεία δράσης του πράκτορα. Υπάρχουν πολλές γλώσσες προγραμματισμού που υποστηρίζουν την ανάπτυξη ορθολογικών πρακτόρων και ιδιαίτερα την BDI (Belief - Desire - Intention) αρχιτεκτονική.

Στα πλαίσια αυτής της διπλωματικής εργασίας αρχικά μελετάμε το σύστημα της Τροπικής Λογικής (Modal Logic) όσον αφορά το συντακτικό και σημασιολογικό κομμάτι καταγράφοντας τα βασικά αξιώματα και τους κανόνες αυτής της λογικής. Ενώ στη συνέχεια, αναπτύξαμε κάποιες εφαρμογές που βασίζονται σε διάφορες ερμηνείες των τροπικών τελεστών και σε διαφορετικά αξιώματα της Επιστημικής λογικής, που επηρεάζουν ανάλογα και τον τρόπο που ο πράκτορας αντιλαμβάνεται το περιβάλλον του και δρα σε αυτό.

## 1.2 Μερικές Ερμηνείες των Τροπικών Τελεστών

### 1.2.1 Epistemic Logic

Η Επιστημική λογική είναι η πιο γνωστή περιοχή της Τροπικής λογικής και αφορά τη γνώση και τις πεποιθήσεις. Οι πιθανοί κόσμοι αντιμετωπίζονται ως καταστάσεις γνώσης ενώ οι βασικοί τροπικοί τελεστές της λογικής αυτής είναι οι «believe» (πιστεύω) και «know» (γνωρίζω).

Τα πιο γνωστά συστήματα αξιωμάτων που υπακούουν οι τελεστές της Επιστημικής λογικής είναι τα KD45 και K45. Ενώ η λογική αυτή βρίσκει εφαρμογή σε πολλούς τομείς όπως στη ρομποτική, στην ασφάλεια δικτύων και σε εφαρμογές κρυπτογράφησης.

## 1.2.2 Temporal Logic

Η temporal logic είναι η λογική του χρόνου. Όλοι οι πιθανοί κόσμοι  $w_1, w_2, \dots, w_n$  αποτελούν ουσιαστικά διαφορετικά στιγμιότυπα του τρέχοντα κόσμου σε διαφορετικές χρονικές στιγμές. Με άλλα λόγια σε μια temporal logic έχουμε ένα ιστορικό κόσμων στο οποίο κάθε κόσμος αντιπροσωπεύει μελλοντικές καταστάσεις του τωρινού.

$$w_1 \longrightarrow w_2 \longrightarrow \dots w_n$$

Οι τροπικοί τελεστές αυτής της λογικής είναι παρόμοιοι με αυτούς της Epistemic, μόνο που εδώ μεταφράζονται με άξονα τον χρόνο. Έτσι αντί για τους τροπικούς τελεστές «believe» και «know» χρησιμοποιούμε τον τελεστή «henceforth» που δηλώνει ότι μια πρόταση θα είναι αληθής από εδώ και στο μέλλον.

Δύο χαρακτηριστικά μοντέλα της Temporal logic είναι τα CTL (Computation Tree Logic) και LTL (Linear Time Logic). Το μοντέλο CTL εκφράζει μια λογική χρόνου διακλάδωσης στην οποία ο χρόνος αναπαριστάται με τη μορφή δομής δένδρου και το μέλλον δεν είναι προκαθορισμένο. Το μοντέλο LTL ονομάζεται και προτασιακή Temporal logic στην οποία ο χρόνος αναπαριστάται γραμμικά.

## 1.2.3 Logic for Agency

Η Τροπική λογική βρίσκει εφαρμογή στο χώρο του προγραμματισμού ορθολογικών πρακτόρων (Rational agent). Χρησιμοποιώντας τροπικούς τελεστές μπορούμε να αποδώσουμε στον πράκτορα ορισμένες συμπεριφορές και να προσδιορίσουμε τον τρόπο που αυτός δρα και λαμβάνει αποφάσεις. Οι συμπεριφορές αυτές γενικά μπορούν να χωριστούν σε τρεις κατηγορίες :

Informational attitudes: περιγράφουν το πώς ο πράκτορας αντιλαμβάνεται τον κόσμο του, για παράδειγμα τι γνωρίζει, τι πιστεύει και για τι είναι σίγουρος.

Motivational attitudes: περιγράφουν τις αλλαγές που ο πράκτορας θέλει να πραγματοποιήσει στον κόσμο του και εκφράζονται συνήθως με επιθυμίες, προτιμήσεις και στόχους.

Normative attitudes: περιγράφουν τι πρέπει να κάνει ο πράκτορας, δηλαδή τις υποχρεώσεις και τις απαγορεύσεις του.

## 1.3 Οργάνωση Κειμένου

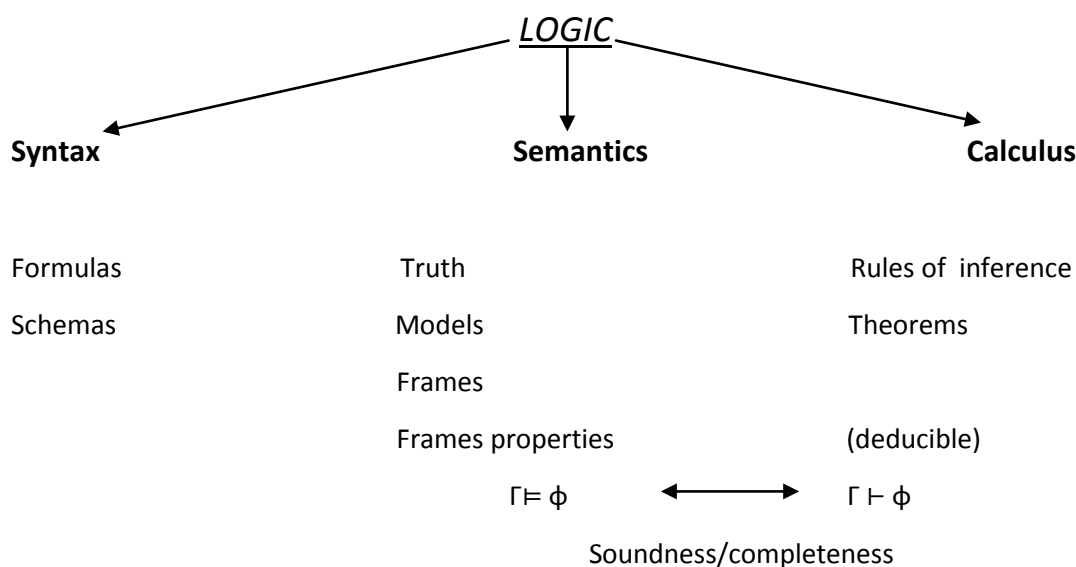
Στο 1<sup>ο</sup> κεφάλαιο έχουμε μια εισαγωγή στο θέμα της διπλωματικής και αναφέρουμε χαρακτηριστικά μερικές ερμηνείες τροπικών τελεστών που θα μας βοηθήσουν στη συνέχεια στην αποσαφήνιση ορισμένων εννοιών. Στο κεφάλαιο 2 αναλύουμε την Τροπική λογική ως προς την σύνταξη και τη σημασιολογία, δίνοντας και κάποια παραδείγματα στους αντίστοιχους ορισμούς. Συνεχίζουμε στο κεφάλαιο 3 με θέματα θεωρητικού υπόβαθρου και



πιο συγκεκριμένα με τα συστήματα Τροπικής λογικής με τα οποία θα ασχοληθούμε και στο κομμάτι των εφαρμογών. Στο κεφάλαιο 4 περιγράφουμε τα στάδια της υλοποίησης των εφαρμογών και των frameworks που μελετήθηκαν. Τα συμπεράσματα που προκύπτουν και οι μελλοντικές επεκτάσεις σχετικά με τις εφαρμογές διατυπώνονται στο κεφάλαιο 5. Τέλος στο κεφάλαιο 6 παρουσιάζεται η βιβλιογραφία που απαιτήθηκε.

## 2 Θεωρητικό Υπόβαθρο

Η Τροπική λογική ορίζεται από τη σύνταξη, τη σημασιολογία και τις λογικές σχέσεις. Για να μπορέσουμε επομένως να κατανοήσουμε την λογική αυτή και τον τρόπο που βρίσκει εφαρμογή στον προγραμματισμό πρακτόρων θα ορίσουμε στα επόμενα κεφάλαια κάποιες βασικές έννοιες και θα διατυπώσουμε παραδείγματα για αυτές.



Σχ. 1: *Logic = Syntax + Semantics (or Calculus)*.

### 2.1 Εισαγωγή στη Σύνταξη της Τροπικής Λογικής

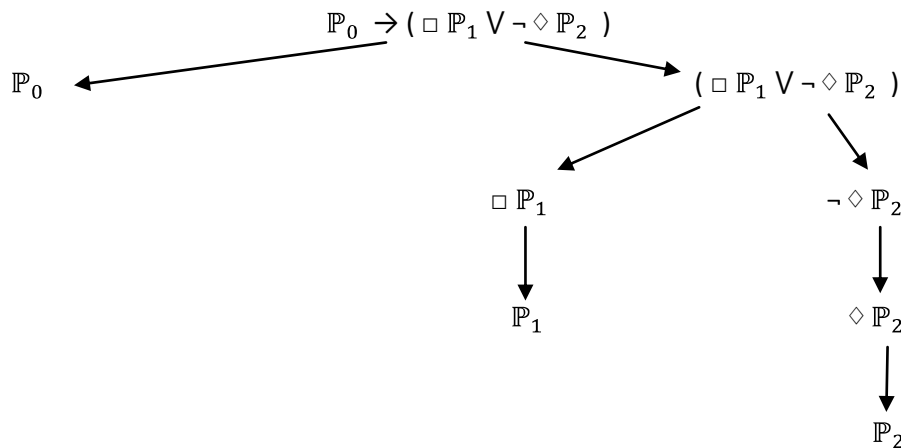
Η Τροπική λογική είναι ένας κλάδος της λογικής που ασχολείται με έννοιες του τύπου «είναι αναγκαίο» και «είναι πιθανό».

Η γλώσσα αποτελείται από:

1. Ατομικές προτάσεις της μορφής  $\mathbb{P}_n$  ( $n=0,1,\dots$ ).
2. Σταθερές «T» «F», οι οποίες αντιπροσωπεύουν το αληθές και το ψευδές αντίστοιχα.
3. Τα κλασσικά σύμβολα  $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$  που ισχύουν και στην προτασιακή λογική.
4. Το σύμβολο της αναγκαιότητας  $\square$  και το σύμβολο του πιθανού  $\diamond$ .

Ένας τρόπος να αναπαραστήσουμε τις ατομικές προτάσεις είναι με την μορφή δένδρου (construction tree). Κάθε πρόταση αποτελείται από υποπροτάσεις (subsenteses) που εμφανίζονται στους κόμβους του δένδρου ενώ κάθε κλάδος δείχνει την σειρά της εφαρμογής της συντακτικής λειτουργίας.

Έστω για παράδειγμα ότι θέλουμε να αναπαραστήσουμε με μορφή δένδρου την εξής πρόταση:  $\mathbb{P}_0 \rightarrow (\Box \mathbb{P}_1 \vee \neg \Diamond \mathbb{P}_2)$



Σχ. 2: Construction tree for  $\mathbb{P}_0 \rightarrow (\Box \mathbb{P}_1 \vee \neg \Diamond \mathbb{P}_2)$

Με τον όρο schema (σχήμα) εννοούμε ένα σύνολο προτάσεων που συνήθως εμφανίζεται με μία συγκεκριμένη μορφή. Για παράδειγμα το

$$5. \Diamond A \rightarrow \Box \Diamond A,$$

αποτελεί ένα σχήμα σύμφωνα με το οποίο ότι είναι πιθανό είναι και απαραίτητα πιθανό.

Μερικά από τα πιο βασικά σχήματα Τροπικής λογικής είναι τα ακόλουθα:

- T.  $\Box A \rightarrow A$
- K.  $\Box (A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$
- DF  $\Diamond$ .  $\Diamond A \leftrightarrow \neg \Box \neg A$
- 4.  $\Box A \rightarrow \Box \Box A$
- B.  $A \rightarrow \Box A$
- D.  $\Box A \rightarrow \Diamond A$

## 2.2 Kripke Semantics

Όπως γνωρίζουμε στην προτασιακή λογική υπάρχουν φόρμουλες που είναι πάντα αληθείς ανεξάρτητα από την τιμή αληθείας των προτάσεων που εμπλέκονται. Είναι οι γνωστές ταυτολογίες. Κάτι αντίστοιχο συμβαίνει και με την Τροπική λογική.

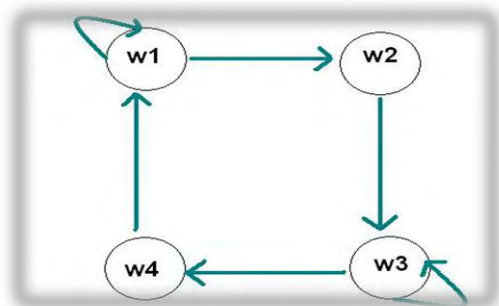
Οι προτάσεις εξετάζονται μέσα σε κόσμους. Πριν δώσουμε τον ορισμό της αλήθειας πρέπει να ορίσουμε την έννοια του μοντέλου στο οποίο θα εξετάσουμε την αλήθεια.

Πλαίσιο ονομάζεται ένα ζεύγος  $F = \langle W, R \rangle$  τέτοιο ώστε :

- Το  $W$  είναι μια συλλογή πιθανών κόσμων, και
- Το  $R$  είναι μια διμελής σχέση επί του  $W$  που ονομάζεται σχέση προσπελασιμότητας

(Alwen Tiu, 2009).

Το πλαίσιο αναπαριστάται και ως ένας κατευθυνόμενος γράφος, πιθανώς πεπερασμένος, όπου το  $W$  είναι το σύνολο των ακμών και  $R$  το σύνολο των κορυφών. Για παράδειγμα για ένα Πλαίσιο  $F = \langle W, R \rangle$  με πιθανούς κόσμους τους  $w_1, w_2, w_3, w_4$  και σχέσεις  $\langle w_1, w_1 \rangle, \langle w_1, w_2 \rangle, \langle w_2, w_3 \rangle, \langle w_3, w_3 \rangle, \langle w_3, w_4 \rangle, \langle w_4, w_1 \rangle$  σχηματίζεται ο εξής γράφος:



Σχ. 3: Γράφος πιθανών κόσμων.

Μοντέλο (model) Τροπικής λογικής ή μοντέλο Κρίκε ονομάζεται το ζεύγος  $\langle F, V \rangle$ , όπου  $F$  είναι ένα πλαίσιο, και το  $V$  είναι μια συνάρτηση η οποία αντιστοιχίζει κάθε προτασιακή μεταβλητή σε ένα υποσύνολο του  $W$  (Alwen Tiu, 2009).

Επομένως για ένα πλαίσιο  $F = \langle W, R \rangle$  θα έχουμε το μοντέλο  $M$  βασισμένο στο πλαίσιο  $F$  και θα γράφουμε  $M = \langle W, R, V \rangle$ . Στο γράφο του πλαισίου Κρίκε προστίθεται τώρα και η συνάρτηση αντιστοίχισης μεταξύ των κόσμων.

### 2.2.1 Ορισμός της Αλήθειας

Με την θεωρία των πιθανών κόσμων ο Κρίκε εισήγαγε την έννοια της προσβασιμότητας (accessibility). Οι σχέσεις προσβασιμότητας που έχουν οι κόσμοι καθορίζουν την τιμή αληθείας της κάθε πρότασης.

Σύμφωνα λοιπόν με τον ορισμό του Κρίκε:

«*Necessarily  $p$  is true at a world  $w$  if and only if  $p$  is true at every world  $w'$  accessible from  $w$* » (Edward N. Zalta, 1995).

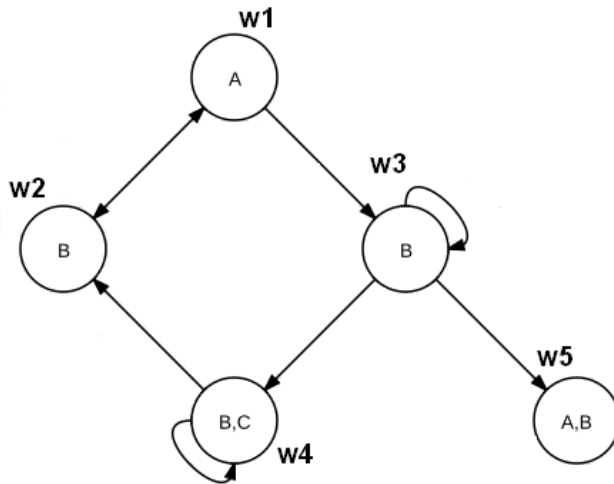
Έτσι λοιπόν για ένα κόσμο  $W$  αν μια πρόταση  $p$  είναι αληθής σε όλους τους προσβάσιμους από τον  $W$  κόσμους τότε και η  $\Box p$  είναι αληθής στο  $W$ . Ενώ αντίστοιχα αν μια πρόταση  $p$  είναι αληθής σε τουλάχιστον έναν προσβάσιμο κόσμο τότε και η  $\Diamond p$  είναι αληθής στον  $W$ .

Έστω  $\alpha$  είναι ένας πιθανός κόσμος στο μοντέλο  $M = \langle W, R, V \rangle$  και  $A$  μια πρόταση. Γράφουμε  $\models_{\alpha}^M A$  για να δείξουμε ότι η πρόταση  $A$  αληθεύει στο κόσμο  $\alpha$  στο μοντέλο  $M$ . Αναδρομικά ορίζουμε την αλήθεια με τις ακόλουθες συνθήκες:

1.  $\models_{\alpha}^M \mathbb{P}_i$  αν και μόνο αν  $\alpha \in \mathbb{P}_i$ .
2.  $\models_{\alpha}^M \top$ .
3. Δεν ισχύει ποτέ  $\models_{\alpha}^M \perp$ .

4.  $\models_{\alpha}^M \neg A$  αν και μόνο αν δεν ισχύει  $\models_{\alpha}^M A$ .
5.  $\models_{\alpha}^M A \wedge B$  αν και μόνο αν  $\models_{\alpha}^M A$  και  $\models_{\alpha}^M B$ .
6.  $\models_{\alpha}^M A \vee B$  αν και μόνο αν  $\models_{\alpha}^M A$  ή  $\models_{\alpha}^M B$ .
7.  $\models_{\alpha}^M A \rightarrow B$  αν και μόνο αν  $\text{An } \models_{\alpha}^M A$  τότε  $\models_{\alpha}^M B$ .
8.  $\models_{\alpha}^M A \leftrightarrow B$  αν και μόνο αν  $\models_{\alpha}^M A$  αν και μόνο αν  $\models_{\alpha}^M B$ .
9.  $\models_{\alpha}^M \Box A$  αν και μόνο αν για κάθε  $\beta$  στο μοντέλο  $M$ ,  $\models_{\beta}^M A$ .
10.  $\models_{\alpha}^M \Diamond A$  αν και μόνο αν για κάποιο  $\beta$  στο μοντέλο  $M$ ,  $\models_{\beta}^M A$ .

(Brian F. Chellas, 1980)



Σχ. 4: Μοντέλο – παράδειγμα αληθείας.

Για παράδειγμα στο σχήμα που παρουσιάζεται δίπλα, από τις εκφράσεις που διατυπώσαμε παραπάνω παρατηρούμε ότι αληθεύουν τα εξής:

- ✓  $w4 \models B \wedge C$
- ✓  $w3 \models \Box B$
- ✓  $w3 \models \Diamond C$
- ✓  $w1 \models \Diamond \Box B$

Μία πρόταση  $A$  που αληθεύει για κάθε μοντέλο  $M$  και για κάθε κόσμο  $\alpha \in W$  λέγεται έγκυρη και γράφουμε  $\models A$ .

Σε αυτό το σημείο πρέπει να αναφερθούμε στις ταυτολογίες, γνωστές από την προτασιακή λογική, και να δώσουμε τον ορισμό του όρου «ταυτολογικό επακόλουθο» (tautological consequence) .

Η Τροπική λογική είναι μια επέκταση της κλασσικής λογικής, όποτε κάθε πρόταση που αληθεύει στην κλασσική λογική αληθεύει και στην Τροπική. Έτσι αν για μία πρόταση  $A$  γνωρίζουμε ότι είναι μια ταυτολογία τότε η πρόταση αυτή θα είναι έγκυρη και στο μοντέλο της Τροπικής λογικής. Το σύνολο των ταυτολογιών που ισχύουν στην προτασιακή λογική συμβολίζεται με PL. «Επομένως για μια πρόταση  $A$  η οποία είναι ταυτολογικό επακόλουθο των προτάσεων  $A_1, A_2, \dots, A_n$  θα λέμε ότι είναι έγκυρη σε οποιαδήποτε κλάση μοντέλων εφόσον κάθε μία από τις προτάσεις  $A_1, A_2, \dots, A_n$  είναι έγκυρες σε αυτό» (Brian F. Chellas, 1980).

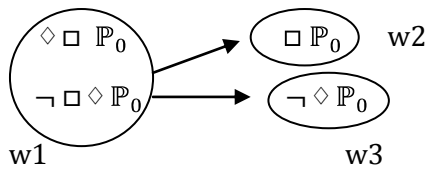
Για να μπορέσουμε όμως να ολοκληρώσουμε τον ορισμό της εγκυρότητας πρέπει να εξετάσουμε σε κάθε σύστημα Τροπικής λογικής το κάθε σχήμα ξεχωριστά. Για παράδειγμα, όπως θα δούμε στην παρακάτω ενότητα δεν είναι έγκυρα όλα τα σχήματα που αναφέραμε προηγουμένως σε όλα τα standard models.

## 2.2.2 Εγκυρότητα και Πλαίσια

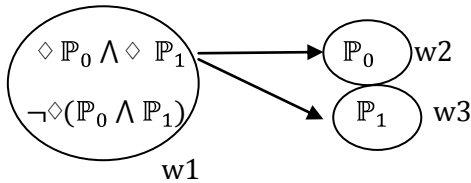
Για να αποδείξουμε ότι ένα σχήμα είναι έγκυρο πρέπει να επιχειρηματολογήσουμε ότι ένα στιγμιότυπο του είναι αληθές σε μία αυθαίρετη κατάσταση, σε μία αυθαίρετη σχεσιακή δομή ενώ για να αποδείξουμε ότι δεν είναι έγκυρο πρέπει να δηλώσουμε την σχεσιακή δομή και την αντίστοιχη κατάσταση.

Στη συνέχεια παρουσιάζονται μερικά σχήματα τα οποία δεν είναι έγκυρα και χρησιμοποιούνται τα λεγόμενα «*falsifying models*» όπως αναφέρονται από τον *Edward N. Zalta* (*Edward N. Zalta, 1995*).

- $G. \diamond \Box P_0 \rightarrow \Box \diamond P_0$

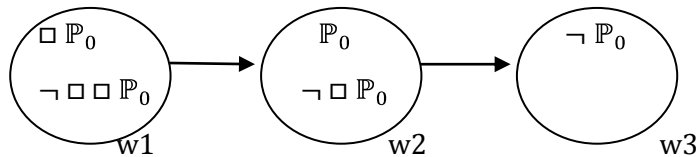


- $(\diamond P_0 \wedge \diamond P_1) \rightarrow \diamond(P_0 \wedge P_1)$



Όπως φαίνεται και στο μοντέλο από τον κόσμο w1 μπορούμε να πάμε σε κόσμο που μπορεί να αληθεύει το  $P_0$  ή το  $P_1$  όμως δεν ισχύει το  $\diamond(P_0 \wedge P_1)$ . Έτσι καταλήγουμε ότι το παραπάνω στιγμιότυπο σχήματος δεν είναι έγκυρο.

- 4.  $\Box P_0 \rightarrow \Box \Box P_0$



Στον κόσμο w3 αληθεύει το  $\neg P_0$  (not  $\models_{w3}^M P_0$ ) οπότε στον κόσμο w2 θα αληθεύει το  $\neg \Box P_0$  (not  $\models_{w3}^M \Box P_0$ ) και επειδή ο κόσμος w2 είναι ο μόνος προσβάσιμος στον w1 κόσμος θα αληθεύει και το  $\neg \Box \Box P_0$  (not  $\models_{w3}^M \Box \Box P_0$ ). Επομένως καταλήξαμε ότι το σχήμα δεν είναι έγκυρο.

Αντιθέτως για να αποδείξουμε ότι το σιγμύτυπο  $\Box \mathbb{P}_0 \wedge \Box \mathbb{P}_1 \rightarrow \Box (\mathbb{P}_0 \wedge \mathbb{P}_1)$  είναι έγκυρο επιχειρηματολογούμε ως εξής:

Επιλέγουμε αυθαίρετα ένα μοντέλο  $M = \langle W, R, V \rangle$  και έναν κόσμο  $w \in W$ . Υποθέτουμε ότι  $\models_w^M \Box \mathbb{P}_0 \wedge \Box \mathbb{P}_1$  τότε θα ισχύουν  $\models_w^M \Box \mathbb{P}_0$  και  $\models_w^M \Box \mathbb{P}_1$ . Επομένως για έναν κόσμο  $w' \in W$  θα ισχύει  $\models_{w'}^M \mathbb{P}_0$  και  $\models_{w'}^M \mathbb{P}_1$ . Έτσι έχουμε  $\models_w^M \Box \mathbb{P}_0 \wedge \Box \mathbb{P}_1$  που συνεπάγεται  $\models_w^M \Box (\mathbb{P}_0 \wedge \mathbb{P}_1)$ .

Στο Βασικό Τροπικό λογισμό ανάλογα με την ερμηνεία που δίνουμε στους βασικούς τροπικούς τελεστές και τις ιδιότητες των σχέσεων προσβασιμότητας τα σχήματα, που στη γενική τους περίπτωση δεν είναι έγκυρα, μπορούν να θεωρηθούν ως αληθή.

Σε ένα μοντέλο Kripke θεωρούμε τις εξής ιδιότητες για μια σχέση προσβασιμότητας:

- Σειριακή (serial): Για κάθε  $x \in W$  υπάρχει  $y \in W$  τέτοιο ώστε  $xRy$ .
- Αντανακλαστική (reflexive): Για κάθε  $x \in W$ ,  $xRx$ .
- Συμμετρική (symmetric): Για κάθε  $x \in W$  και  $y \in W$ , αν  $xRy$  τότε και  $yRx$ .
- Μεταβατική (transitive): Για κάθε  $x \in W$  και  $y \in W$  και  $z \in W$ , αν  $xRy$  και  $yRz$  τότε και  $xRz$ .
- Ευκλείδεια (Euclidean): Για κάθε  $x \in W$  και  $y \in W$  και  $z \in W$ , αν  $xRy$  και  $xRz$  τότε και  $yRz$ .

*(Brian F. Chellas, 1980)*

Συνδυάζοντας τώρα τις ιδιότητες με τα κατάλληλα σχήματα που είδαμε προηγουμένως μπορούμε να ορίσουμε μια θεωρία αντιστοιχίας μεταξύ της σχέσης προσβασιμότητας και της εγκυρότητας αυτών των σχημάτων. Έτσι για ένα μοντέλο  $M = \langle W, R, V \rangle$ :

- Η σχέση  $R$  είναι ανακλαστική αν και μόνο αν  $\models^M \Box A \rightarrow A$  (T.).
- Η σχέση  $R$  είναι σειριακή αν και μόνο αν  $\models^M \Box A \rightarrow \Diamond A$  (D.).
- Η σχέση  $R$  είναι συμμετρική αν και μόνο αν  $\models^M A \rightarrow \Box \Diamond A$  (B.).
- Η σχέση  $R$  είναι μεταβατική αν και μόνο αν  $\models^M \Box A \rightarrow \Box \Box A$  (4.).
- Η σχέση  $R$  είναι ευκλείδεια αν και μόνο αν  $\models^M \Diamond A \rightarrow \Box \Diamond A$  (5.).

Διαφορετικά μπορούμε να πούμε ότι το κάθε σχήμα είναι έγκυρο (valid with respect to the class of models) σε αντιστοιχία με την κλάση του μοντέλου που υποδεικνύεται στους παραπάνω κανόνες. Αντίστοιχα μπορούμε να διατυπώσουμε τους κανόνες για κάθε πλαίσιο  $F = \langle W, R \rangle$  (valid with respect to the class of frames).

Ολοκληρώνοντας ως ένα επίπεδο και με τον ορισμό της εγκυρότητας πρέπει να αναφέρουμε τη σχέση των κανόνων συμπερασμού (rules of inference) με την διατήρηση της αλήθειας και της εγκυρότητας σε ένα σύστημα Τροπικής λογικής. Οι πιο σημαντικοί κανόνες συμπερασμού, τους οποίους θα εξετάσουμε εκτενέστερα και στα επόμενα κεφάλαια, είναι οι εξής:

- MP. (Modus Ponens)  $\text{Av } A \rightarrow B \text{ και } A \text{ τότε } B$ .
- RPL. (Rule of Propositional Logic)  $\text{Av } B \text{ είναι ταυτολογικό επακόλουθο των } A_1, A_2, \dots, A_n \text{ τότε από } A_1, A_2, \dots, A_n \text{ συνεπάγεται } B$ .
- RN. (Rule of Necessitation)  $\text{Av } A \text{ τότε } \Box A$ .

- RK. (Rule K)  $\text{An } A_1 \rightarrow A_2 \rightarrow \dots \rightarrow A_n \rightarrow A$  τότε  $\Box A_1 \rightarrow \Box A_2 \rightarrow \dots \rightarrow \Box A_n \rightarrow \Box A$ .

Οι κανόνες συμπερασμού παράγουν από τα αξιώματα ενός συστήματος τα αντίστοιχα θεωρήματα έτσι συμπερασματικά εφόσον τα αξιώματα είναι έγκυρα οι κανόνες αυτοί διατηρούν την εγκυρότητα και την αλήθεια σε ένα σύστημα Τροπικής Λογικής.



Εικ1. Οι κανόνες συμπερασμού εφαρμόζονται σε αξιώματα και αληθείς προτάσεις και παράγουν νέες αληθείς προτάσεις διατηρώντας έτσι την αλήθεια σε ένα σύστημα.

## 2.3 Soundness and Completeness

Τα θεωρήματα της ορθότητας (soundness) και της πληρότητας (completeness) αποτελούν τη σύνδεση μεταξύ της σύνταξης και της σημασιολογίας της Τροπικής λογικής.

Ουσιαστικά το θεώρημα της ορθότητας (soundness) εκφράζει το γεγονός ότι δεν μπορείς να αποδείξεις κάτι το οποίο είναι ψευδές ενώ το θεώρημα της πληρότητας (completeness) σε αντιδιαστολή εκφράζει ότι μπορείς να αποδείξεις οτιδήποτε το οποίο είναι αληθές.

Πριν όμως δώσουμε το ακριβή ορισμό των παραπάνω θεωρημάτων υπάρχουν κάποιες έννοιες που πρέπει να διασαφηνιστούν.

Ο όρος «deducible» περιφραστικά σημαίνει το να καταλήγεις σε συμπέρασμα μέσα από μια σειρά επιχειρημάτων. «Μια πρόταση  $A$  είναι *deducible* από ένα σύνολο προτάσεων  $\Gamma$  ορισμένο σε ένα σύστημα  $\Sigma$  ( $\Gamma \vdash_{\Sigma} A$ ) αν και μόνο αν το  $\Sigma$  περιέχει ένα θεώρημα της μορφής  $(A_1 \wedge A_2 \wedge \dots \wedge A_n) \rightarrow A$  όπου οι παράγοντες  $A_i$  ( $i=1(1)n$ ) είναι προτάσεις στο  $\Gamma$ » (Edward N. Zalta, 1995).

Επιπλέον ένα σύνολο προτάσεων  $\Gamma$  ορισμένο σε ένα σύστημα  $\Sigma$  λέμε ότι είναι «consistent» (συνεπές) μόνο στην περίπτωση που το (falsum)  $\perp$  δεν παράγεται από το σύνολο  $\Gamma$ . Αλλιώς γράφουμε  $\text{Con}_{\Sigma}\Gamma$  (Brian F. Chellas, 1980).

Στα πλαίσια της συνέπειας χαρακτηρίζουμε ένα σύνολο προτάσεων ορισμένο σε ένα σύστημα  $\Sigma$  ως «maximal» μόνο στην περίπτωση που είναι  $\Sigma$ -consistent και έχει μόνο  $\Sigma$ -consistent καθαρές επεκτάσεις. Επομένως όπως αναφέρει και στο βιβλίο του, *Modal Logic: An introduction*, ο Brian F. Chellas: «Ένα σύνολο προτάσεων  $\Gamma$  είναι *maximal* ( $\text{Max}_{\Sigma}\Gamma$ ) αν και μόνο αν είναι  $\text{Con}_{\Sigma}\Gamma$  και για κάθε πρόταση  $A$  ισχύει  $\text{Con}_{\Sigma}\Gamma \cup \{A\}$  για  $A \in \Gamma$ » (Brian F. Chellas, 1980).

Έτσι λοιπόν η έννοια της ορθότητας και της πληρότητας αναπτύσσουν μια σχέση αντιστοίχισης μεταξύ των deducible και των έγκυρων συνόλων προτάσεων  $\Gamma$ . Πιο συγκεκριμένα ένα σύστημα Τροπικής Λογικής  $\Sigma$  καλείται ορθό (sound) σε αντιστοιχία με μια κλάση μοντέλων  $C$  (sound with respect to a class of models  $C$ ) μόνο στην περίπτωση που κάθε θεώρημα του συστήματος  $\Sigma$  ορισμένο στην κλάση  $C$  είναι έγκυρο. Δηλαδή «για κάθε πρόταση  $A$  αν  $\vdash_{\Sigma} A$  τότε  $\models_C A$ » (Brian F. Chellas, 1980). Αντίστοιχα ένα σύστημα Τροπικής



Λογικής  $\Sigma$  καλείται πλήρες (complete) σε αντιστοιχία με μια κλάση μοντέλων  $C$  (sound with respect to a class of models  $C$ ) αν και μόνο αν κάθε έγκυρη πρόταση ορισμένη στην κλάση  $C$  είναι θεώρημα του συστήματος  $\Sigma$ . Δηλαδή «για κάθε πρόταση  $A$  αν  $\models_C A$  τότε  $\vdash_\Sigma A$ » (Brian F. Chellas, 1980).

Σε συνδυασμό αυτών των δύο θεωρημάτων χαρακτηρίζουμε ένα σύστημα  $\Sigma$  «determined» αν είναι πλήρες (complete) και ορθό (sound) σε αντιστοιχία με μια κλάση μοντέλων  $C$ .

Για να χαρακτηρίσουμε ένα σύστημα ορθό ή/και πλήρες υπάρχουν πολλοί διαφορετικοί τρόποι προσέγγισης. Σε αυτό το σημείο θα αναφέρουμε την βασική ιδέα που χρησιμοποιείται για την απόδειξη της πληρότητας κρίνοντας την ως πιο δύσκολη διαδικασία.

### 2.3.1 The Canonical Model Method

Κάθε σύστημα Τροπικής λογικής έχει ένα μοντέλο, όσο πιο γενικό γίνεται, που βασίζεται στη βασική σύνταξη της λογικής. Αυτό το μοντέλο ονομάζεται canonical model. Η ύπαρξη αυτού του μοντέλου σε ένα σύστημα αποδεικνύει και την πληρότητα του συγκεκριμένου συστήματος.

Στη συνέχεια παραθέτουμε ένα παράδειγμα που αφορά το canonical standard model για ένα normal system. Τα normal systems είναι μια κατηγορία συστημάτων την οποία θα αναλύσουμε στο επόμενο κεφάλαιο.

Έστω έχουμε το σύστημα  $K5$  το οποίο αποτελείται από το σχήμα  $K$  και το σχήμα 5.

$$\diamond \varphi \rightarrow \Box \diamond \varphi \quad 5.$$

Σύμφωνα με τη θεωρία αντιστοιχίας της εγκυρότητας των σχημάτων και της σχέσης προσβασιμότητας μπορούμε να πούμε ότι το σύστημα είναι sound with respect to the class of Euclidean models. Επομένως έχουμε  $xRy \ \& \ xRz \rightarrow yRz$  (Euclidean).

Από τον ορισμό των canonical standard models διατυπώνουμε τώρα τις εξής σχέσεις για να αποδείξουμε ότι το σύστημα είναι complete εάν από τις σχέσεις (1) και (2) μπορούμε να παράγουμε την (3).

- (1)  $\Box \varphi \in x \subseteq y$
- (2)  $\Box \varphi \in x \subseteq z$
- (3)  $\Box \varphi \in y \subseteq z$

Έστω  $\Box \varphi \in y$  τότε  $\diamond \Box \varphi \in x$ .

Από το σχήμα 5.  $\diamond \neg \varphi \rightarrow \Box \diamond \neg \varphi$

Βρίσκουμε το contrapositive  $\diamond \Box \neg \varphi \rightarrow \Box \neg \varphi$

Αντικαθιστούμε όπου  $\neg \varphi$  με  $\varphi$   $\diamond \Box \varphi \rightarrow \Box \varphi$

Οπότε το  $\Box \varphi \in x$  και συνεπάγεται το  $\varphi \in z$ . Από τις (1) και (2) παρήγαμε την (3) δηλαδή το σύστημα είναι complete.

# 3 Συστήματα Τροπικής Λογικής

## 3.1 Normal Systems

Έχουμε ήδη αναφερθεί στον όρο «σύστημα» Τροπικής λογικής και στο πως διαμορφώνεται ανάλογα με τα σχήματα και τους κανόνες συμπερασμού που ισχύουν σε αυτό. Γενικά ένα σύστημα Τροπικής λογικής είναι ένα σύνολο προτάσεων που περιλαμβάνει όλες τις ταυτολογίες και διέπεται από τον κανόνα συμπερασμού RPL (closed under the rule of inference RPL).

Την πιο βασική κατηγορία συστημάτων Τροπικής λογικής αποτελούν τα Κανονικά συστήματα (Normal systems). Η Κανονική λογική περιλαμβάνει τόσο τις ταυτολογίες και τα σχήματα που αποτελούν προτασιακό συμπερασμό, αλλά και όλους τους συνδυασμούς των σχημάτων Τροπικής λογικής. Συγκεκριμένα χαρακτηρίζουμε ένα σύστημα Τροπικής λογικής ως Κανονικό (Normal) «αν και μόνο αν,

- περιλαμβάνει το σχήμα  $DF\Diamond$ .  $\Diamond A \leftrightarrow \neg \Box \neg A$ ,
- διέπεται από τον κανόνα συμπερασμού  $RK$ .  $\frac{(A_1 \wedge A_2 \wedge \dots \wedge A_n) \rightarrow A}{(\Box A_1 \wedge \Box A_2 \wedge \dots \wedge \Box A_n) \rightarrow \Box A}$  ( $n \geq 0$ )»

(Brian F. Chellas, 1980).

Ο παραπάνω ορισμός βασίζεται σε δύο βασικούς κανόνες για να ορίσει μια Κανονική Τροπική λογική, συνδυάζοντας όμως τα διάφορα σχήματα και τους κανόνες συμπερασμού που ισχύουν σε κάθε Κανονικό τροπικό σύστημα μπορούν να προκύψουν διαφορετικοί συνδυασμοί που το υποδεικνύουν.

Γενικά όμως ένα σύστημα  $\Sigma$  είναι κανονικό εάν :

- Περιέχει όλες τις ταυτολογίες
- Περιέχει όλα τα στιγμιότυπα του σχήματος  $K$ .  $\Box (A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$
- Περιέχει όλα τα στιγμιότυπα του σχήματος  $DF\Diamond$ .  $\Diamond A \leftrightarrow \neg \Box \neg A$
- Παρουσιάζει κλειστότητα ως προς τον τελεστή  $\Box$  (closed under the Rule of necessitation)
- Παρουσιάζει κλειστότητα ως προς τον κανόνα συμπερασμού MP. (Modus Ponens)  $A \rightarrow B$  και  $A$  τότε  $B$ .

Έστω έχουμε ένα σύστημα  $\Sigma$  που περιέχει το σχήμα  $DF\Diamond$  ., τότε μπορούμε να πούμε ότι το σύστημα  $\Sigma$  είναι Κανονικό αν και μόνο αν περιέχει επίσης το σχήμα  $K$  και παρουσιάζει κλειστότητα ως προς τον κανόνα RN. Για να αποδείξουμε την πρόταση αυτή αρκεί να δείξουμε ότι το σύστημα  $\Sigma$  παρουσιάζει επίσης κλειστότητα ως προς τον κανόνα RK όπως υποδεικνύει και ο αρχικός ορισμός που ορίστηκε παραπάνω. Επομένως από RN.  $\frac{A}{\Box A}$  για μια ακολουθία της μορφής  $(A_1 \wedge A_2 \wedge \dots \wedge A_n) \rightarrow A$  που παράγεται από το σύστημα  $\Sigma$  προκύπτει  $(\Box A_1 \wedge \Box A_2 \wedge \dots \wedge \Box A_n) \rightarrow \Box A$  (κανόνας RK.).

Το μικρότερο Κανονικό σύστημα Τροπικής λογικής ονομάζεται Κ. Επομένως κάθε Κανονικό σύστημα είναι ένα Κ-σύστημα. Ανάλογα με τα σχήματα που περιλαμβάνει το κάθε Κανονικό σύστημα παίρνει και την ονομασία του, δηλαδή  $KS_1S_2 \dots S_n$  ονομάζεται ένα σύστημα εάν περιλαμβάνει κάθε στιγμιότυπο των σχημάτων  $S_1S_2 \dots S_n$ .

Δίνοντας την αντίστοιχη ερμηνεία σε κάθε σχήμα μπορούμε να διατυπώσουμε την «γνώση» του εκάστοτε συστήματος. Για παράδειγμα το σύστημα ΚΤ45 περιλαμβάνει κάθε στιγμιότυπο των σχημάτων Κ, Τ, 4 και 5 επομένως για αυτό το σύστημα μπορούμε να διατυπώσουμε τα εξής ως ανάλυση γνώσης:

- Ο πράκτορας γνωρίζει μόνο ότι είναι αληθές. (Τ.)
- Αν ο πράκτορας γνωρίζει κάποιο γεγονός τότε γνωρίζει ότι το γνωρίζει (4.)
- Αν ο πράκτορας δεν γνωρίζει κάποιο γεγονός τότε γνωρίζει ότι δεν το γνωρίζει. (5.)

Μερικά από τα πιο γνωστά Κανονικά συστήματα Τροπικής λογικής είναι τα εξής:

- ΚΤ
- ΚD: γνωστό και ως «Deontic Logic», λογική που αφορά τις υποχρεώσεις.
- ΚΤ4: γνωστό και ως S4.
- ΚΤ5: γνωστό και ως S5.

### 3.1.1 Κανόνες Αντικατάστασης και Διϊσμός

Κάθε Κανονικό σύστημα Τροπικής λογικής παρουσιάζει δύο βασικά θεωρήματα, το θεώρημα της αντικατάστασης και το θεώρημα του διϊσμού. Με τη χρήση αυτών των θεωρημάτων μπορούμε να αποδείξουμε αν ένα σχήμα Τροπικής λογικής αποτελεί θεώρημα ενός Κανονικού συστήματος.

Για παράδειγμα χρησιμοποιώντας τα παραπάνω θεωρήματα θέλουμε να αποδείξουμε ότι αν το σχήμα Κ. αποτελεί θεώρημα Κανονικού συστήματος τότε και το σχήμα  $K \diamond$ . θα αποτελεί επίσης θεώρημα του ίδιου συστήματος.

Κ.  $\Box (A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$

$K \diamond$ .  $(\neg \diamond A \wedge \diamond B) \rightarrow \diamond (\neg A \wedge B)$

Απόδειξη:

$\Box (\neg A \rightarrow \neg B) \rightarrow (\Box \neg A \rightarrow \Box \neg B)$

$((\Box \neg A \rightarrow \Box \neg B))^* \rightarrow (\Box (\neg A \rightarrow \neg B))^*$

$(\neg (\Box \neg A)^* \wedge (\Box \neg B)^*) \rightarrow \diamond (\neg (\neg A)^* \wedge (\neg B)^*)$

$(\neg \diamond A \wedge \diamond B) \rightarrow \diamond (\neg A \wedge B)$

Κ.

Dual

Dual's definition

Dual and Rep  $\rightarrow K \diamond$ .

## 3.2 The Schemas and their Combinations

Το μικρότερο Κανονικό σύστημα Τροπικής λογικής μπορεί να συνδυαστεί με οποιοδήποτε από τα βασικά σχήματα Τροπικής λογικής που έχουμε αναφέρει (D., T., B., 4., 5.) και να δημιουργήσει μέχρι 15 διαφορετικά συστήματα Κανονικής λογικής ( $2^5=32$  διαφορετικοί συνδυασμοί εκ των οποίων μόνο οι 15 είναι διακριτοί).

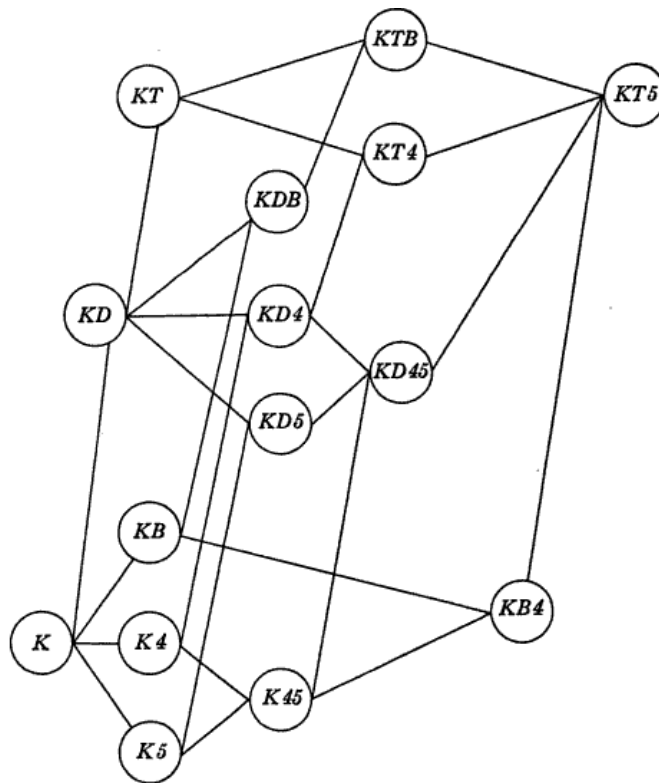
Για παράδειγμα το σύστημα  $KT_D = KT$ . Απόδειξη: Το πρώτο μέρος της απόδειξης είναι προφανές  $KT \subseteq KT_D$ , αρκεί επομένως να αποδείξουμε ότι  $KT_D \subseteq KT$ . Δηλαδή θέλουμε να αποδείξουμε ότι κάθε στιγμιότυπο του σχήματος  $D (\Box A \rightarrow \Diamond A)$  είναι θεώρημα του  $KT$

$\models_{KT} \Box A \rightarrow A$  T.

$\models_{KT} A \rightarrow \Diamond A$  duality

$\models_{KT} \Box A \rightarrow \Diamond A$  PL

Ομοίως μπορούμε να αποδείξουμε  $KT_5 = KT_{D5} = KT_{B5} = KT_{45} = KT_{DB5} = KT_{D45} = KT_{B45} = KT_{DB45}$ .



Εικ2. «Γενική εικόνα όλων των δυνατών διακριτών συνδυασμών» (Brian F. Chellas, 1980).

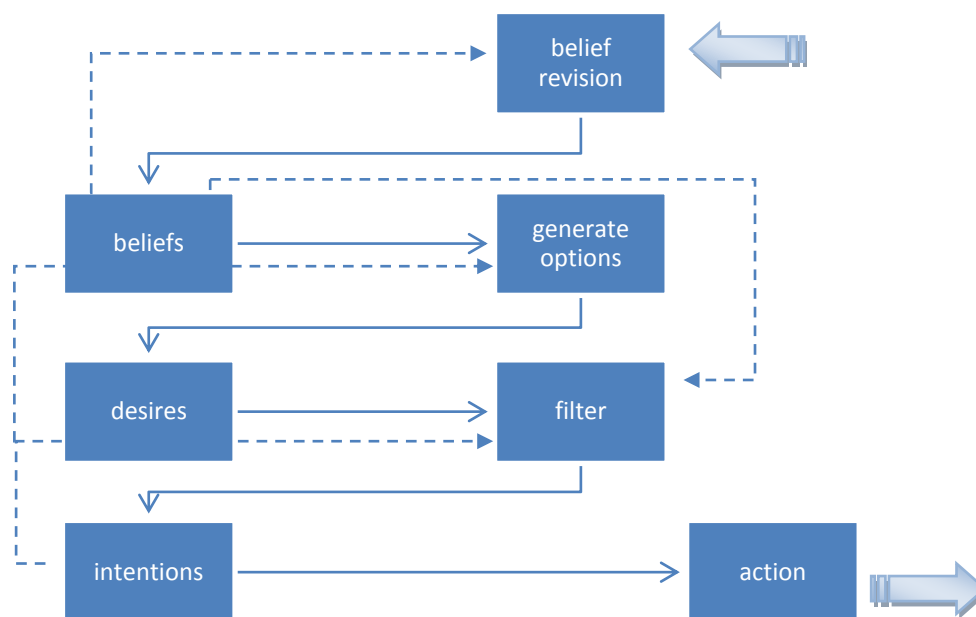
## 4 Υλοποίηση και Κριτική Ανάλυση

Στα πλαίσια της Epistemic Logic, της λογικής της γνώσης, θέλαμε να μελετήσουμε τις διάφορες αρχιτεκτονικές πρακτόρων που αφορούν την αναπαράσταση των πεποιθήσεων και της γνώσης του πράκτορα για τον κόσμο που δρα. Δημιουργήσαμε επομένως κάποιες εφαρμογές - παραδείγματα που βασίζονται στο περιβάλλον των Κύβων (Blocks World) και σε αυτές ο Πράκτορας διαμορφώνει τις πεποιθήσεις και επιλέγει τις ενέργειες που θα οδηγήσουν στην επίτευξη των στόχων του (Planning) ανάλογα με τα αξιώματα του μοντέλου και τα αξιώματα δέσμευσης στόχων του.

### 4.1 BDI Logic

Οι αρχιτεκτονικές BDI (belief-desire-intention) προσπαθούν να προσδιορίσουν τις έννοιες της πεποίθησης, της επιθυμίας και της πρόθεσης. Σε ένα σύστημα πρακτόρων οι πεποιθήσεις αναπαριστούν τα δεδομένα που γνωρίζει ο πράκτορας για τον κόσμο του, οι επιθυμίες αναπαριστούν τις ενέργειες που επιθυμεί ο πράκτορας να εκτελέσει και σχετίζονται με τους στόχους του πράκτορα ενώ οι προθέσεις αφορούν τις επιθυμίες που πρέπει να επιτελέσει ο πράκτορας.

Η γενική διαδικασία που ακολουθείται σε ένα μοντέλο BDI φαίνεται στην Εικ3.



Εικ3. Διαδικασία μοντέλου BDI.

### 4.1.1 Πρώτη Εφαρμογή – Κόσμος των Κύβων

Ας ξεκινήσουμε να περιγράψουμε τον τρόπο που εφαρμόσαμε τις αρχές της BDI λογικής στον Κόσμο των Κύβων.

Στη συγκεκριμένη εφαρμογή ο κόσμος αποτελείται από έναν συγκεκριμένο αριθμό κύβων, ένα τραπέζι πάνω στο οποίο στοιβάζονται οι κύβοι και έναν πράκτορα που εκτελεί κάποιες ενέργειες. Η βάση γνώσης αναπαριστάται με τη μορφή  $\text{fact}(X)$  ενώ το κατηγορημα που ορίζει τους πύργους των κύβων είναι της μορφής  $\text{tower}([\text{list}])$ . Κάθε κύβος έχει ένα χρώμα και μπορεί να βρίσκεται είτε πάνω στο τραπέζι είτε πάνω σε κάποιον άλλον κύβο του ίδιου ή διαφορετικού χρώματος. Για την αναπαράσταση των πεποιθήσεων, των επιθυμιών και των προθέσεων χρησιμοποιούμε τους τελεστές  $\text{Bel}$ ,  $\text{Des}$  και  $\text{Int}$  αντίστοιχα.

Ο πράκτορας έχει κάποιες πεποιθήσεις, επιθυμίες και προθέσεις για τον κόσμο στον οποίο δρα. Θεωρήσαμε ότι ο πράκτορας έχει την πλήρη εικόνα του κόσμου του και ότι οι πεποιθήσεις του για τον κόσμο είναι αληθείς ( $\text{Bel}(X) \text{ :- fact}(X)$ ). Το σύστημα αξιωμάτων που υποστηρίζει το σύνολο των πεποιθήσεων στη BDI λογική είναι το KD45 (*Wiebe van der Hoek*<sup>1,2</sup>). Επομένως έχουμε τους εξής κανόνες όσον αφορά τις πεποιθήσεις :

$$\text{Bel}(p \rightarrow q) \rightarrow (\text{Bel } p \rightarrow \text{Bel } q) \quad \text{K.}$$

$$\text{Bel } p \rightarrow \neg \text{Bel } \neg p \quad \text{D.}$$

$$\text{Bel } p \rightarrow \text{Bel } \text{Bel } p \quad 4.$$

$$\neg \text{Bel } p \rightarrow \text{Bel } \neg \text{Bel } p \quad 5.$$

Επιπρόσθετα συνεπάγονται και δύο κανόνες συμπερασμού :

MP. (Modus Ponens) Αν  $p \rightarrow q$  και  $p$  τότε  $q$ .

RN. (Rule of Necessitation) Αν  $p$  είναι θεώρημα του συστήματος KD45 τότε το  $\text{Bel } p$  είναι επίσης θεώρημα του ίδιου συστήματος.

Για τις επιθυμίες και τις προθέσεις ορίζεται το σύστημα KD.

$$\text{Des}(p \rightarrow q) \rightarrow (\text{Des } p \rightarrow \text{Des } q) \quad \text{K.}$$

$$\text{Des } p \rightarrow \neg \text{Des } \neg p \quad \text{D.}$$

$$\text{Int}(p \rightarrow q) \rightarrow (\text{Int } p \rightarrow \text{Int } q) \quad \text{K.}$$

$$\text{Int } p \rightarrow \neg \text{Int } \neg p \quad \text{D.}$$

Ο πράκτορας έχοντας διαμορφώσει το σύνολο των πεποιθήσεων του για τον κόσμο, υιοθετεί κάποιες επιθυμίες και δεσμεύεται για κάποιες προθέσεις που τελικά θα αλλάξουν τον κόσμο στον οποίο δρα και επομένως και τις πεποιθήσεις του. Ο τρόπος με τον οποίο οι καταστάσεις αυτές του πράκτορα αλληλεπιδρούν ορίζονται με τα εξής αξιώματα συμβατότητας :

Επίγνωση των επιθυμιών και των προθέσεων (Awareness of desires and intentions) : Οι νέες επιθυμίες και προθέσεις που εισάγονται πρέπει να αναφέρονται ως γεγονότα στον πράκτορα.

$(Des \varphi) \rightarrow (Bel(Des \varphi))$

$(Int \varphi) \rightarrow (Bel(Int \varphi))$

Συνειδητή εκτέλεση ενεργειών (No unconscious actions) : Όταν ο πράκτορας εκτελεί μια πράξη τότε γνωρίζει ότι έχει γίνει.

$Done(\alpha) \rightarrow (Bel(Done \alpha))$

Σχέση επιθυμίας-πρόθεσης : Αν ο πράκτορας έχει πρόθεση να εκτελέσει μία ενέργεια τότε η ενέργεια αυτή ή το αποτέλεσμά της βρίσκεται στο σύνολο των επιθυμιών του. Επομένως από τις επιθυμίες του πράκτορα παράγονται και οι αντίστοιχες προθέσεις.

$(Int \alpha) \rightarrow (Des \alpha)$

Σχέση επιθυμίας-πεποίθησης : Αν ο πράκτορας επιθυμεί μία κατάσταση τότε πιστεύει ότι τελικά αυτή η επιθυμία του είναι εφικτή και κάποια στιγμή θα πραγματοποιηθεί. Η σχέση αυτή διαμορφώνεται έμμεσα στο πρόγραμμα καθώς ο πράκτορας αντιλαμβάνεται ως επιθυμίες συγκεκριμένες καταστάσεις που είναι συνεπείς με το σύνολο των πεποιθήσεων του.

$(Des \alpha) \rightarrow (Bel \alpha)$

Ενέργεια που εκτελείται : Αν ο πράκτορας έχει την πρόθεση να εκτελέσει μια ενέργεια τότε θα την εκτελέσει.

$(Int (Does \alpha)) \rightarrow (Does \alpha)$

Οι πεποιθήσεις του πράκτορα συνεχώς αλλάζουν καθώς αλλάζει και ο κόσμος στον όποιο δρα. Ξεκινώντας από μια αρχική κατάσταση του κόσμου ο πράκτορας εκτελεί μια σειρά ενεργειών με σκοπό να καταλήξει σε μια τελική κατάσταση και να αναθεωρήσει τις πεποιθήσεις του.

Ένας τρόπος για να αναπαραστήσουμε τη σχέση μεταξύ των πεποιθήσεων του παρόντος και του παρελθόντος είναι μια διαδικασία αναθεώρησης των πεποιθήσεων που είναι γνωστή ως στρατηγική δέσμευσης. Στον δικό μας Κόσμο των Κύβων έχουμε τρεις τύπους πρακτόρων με διαφορετικές στρατηγικές δέσμευσης (*Wiebe van der Hoek*<sup>1,2</sup>).

Αρχικά έχουμε τον *blindly committed* πράκτορα, ένας πράκτορα που παραμένει «πιστός» στις προθέσεις του μέχρις ότου αποκτήσει την πεποίθηση ότι τελικά η πρόθεση του έχει πραγματοποιηθεί. Στη BDI λογική το αξίωμα για την *blind* δέσμευση δηλώνει ότι :

$INTEND (inevitable \diamond \varphi) \rightarrow inevitable (INTEND (inevitable \diamond \varphi) \cup BEL \varphi)$ .

Η στρατηγική αυτή είναι προφανές ότι είναι πολύ δυνατή. Όπως φαίνεται και στον Κόσμο των Κύβων ο πράκτορας τελικά είτε θα αποκτήσει την πεποίθηση ότι η πρόθεση του έχει πραγματοποιηθεί είτε θα «κουβαλάει» τη συγκεκριμένη πρόθεση για πάντα.

Στη συνέχεια ορίζουμε τον πράκτορα *single-minded committed*. Ο πράκτορας αυτής της στρατηγικής δεν κρατάει τις πεποιθήσεις για πάντα ή μέχρι να πραγματοποιηθούν αλλά παραμένει «πιστός» σε αυτές όσο πιστεύει ότι υπάρχουν ακόμη τρόποι για να πραγματοποιηθούν. Το αξίωμα της *single-minded* δέσμευσης ορίζεται ως εξής :

$\text{INTEND}(\text{inevitable} \diamond \varphi) \rightarrow \text{inevitable}(\text{INTEND}(\text{inevitable} \diamond \varphi) \cup (\text{BEL} \varphi \vee \neg \text{BEL optional} \varphi))$ .

Στον Κόσμο των Κύβων ο αντίστοιχος *single-minded* πράκτορας θα αποδεσμευτεί από την πρόθεση του εφόσον έχουν εξεταστεί όλες οι υπόλοιπες επιθυμίες και έχουν μεταφραστεί στις αντίστοιχες προθέσεις. Εάν τελικά οι μετέπειτα προθέσεις του πράκτορα δεν αλλάζουν τις πεποιθήσεις του έτσι ώστε να δίνουν ένα διαφορετικό τρόπο για να επιτευχθεί η εν λόγω πρόθεση τότε τελικά ο πράκτορας σταματάει να έχει την πρόθεση αυτή.

Τέλος ο πράκτορας *open minded* αποτελεί την πιο χαλαρή σχέση δέσμευσης. Ο πράκτορας αυτού του είδους θα αποδεσμευτεί από τις προθέσεις του εφόσον τελικά δεν επιθυμεί να πραγματοποιήσει τη συγκεκριμένη πρόθεση χωρίς να τον ενδιαφέρει αν τελικά έχει πραγματοποιηθεί ή όχι. Το αξίωμα αυτού του είδους δέσμευσης έχει την εξής μορφή :

$\text{INTEND}(\text{inevitable} \diamond \varphi) \rightarrow \text{inevitable}(\text{INTEND}(\text{inevitable} \diamond \varphi) \cup (\text{BEL} \varphi \vee \neg \text{DES optional} \varphi))$ .

#### 4.1.2 Εκτέλεση Πρώτης Εφαρμογής

Στην εφαρμογή μας ο πράκτορας μπορεί να εκτελέσει κάποιες συγκεκριμένες ενέργειες επομένως μπορεί να διαμορφώσει αντίστοιχες επιθυμίες και να ικανοποιήσει τις προθέσεις που παράγονται από τις τελευταίες. Οι ενέργειες αυτές αφορούν την τοποθέτηση ενός κύβου πάνω σε έναν άλλον ή πάνω στο τραπέζι, τον «καθαρισμό» ενός κύβου έτσι ώστε να μην υπάρχει κανένας πάνω από αυτόν, το επαναχρωματισμό ενός κύβου, την δημιουργία ενός πύργου που αποτελείται από τρεις κύβους, την δημιουργία ενός πύργου που αποτελείται από δύο κύβους και την δημιουργία ενός πύργου που θα αποτελείται από κύβους μόνο του ίδιου χρώματος.

Συγκεκριμένα οι ενέργειες αυτές υλοποιούνται με τα εξής κατηγορήματα :

- `put_on(BlockA,BlockB):-`
- `clear_off(BlockA):-`
- `change_color(BlockA,NewColor):-`
- `establish_three(BlockA, BlockB, BlockC):-`
- `create_tower(X,Y):-`
- `color(Tower_Color):-`

Ο χρήστης εισάγει στο σύστημα μια επιθυμητή τελική κατάσταση του κόσμου ορίζοντας ουσιαστικά τις αλλαγές που επιθυμεί να γίνουν. Για παράδειγμα αν ο χρήστης επιθυμεί ως τελική κατάσταση του κόσμου ο κύβος `BlockF` να βρίσκεται πάνω στον κύβο `BlockG`, να υπάρχει ένας πύργος με τους κύβους `BlockA`, `BlockB`, να υπάρχει ένας πύργος με κύβους μόνο κίτρινου χρώματος, το χρώμα του κύβου `BlockC` και του `BlockD` να είναι



κίτρινο και ο BlockE να μην έχει κανέναν άλλον κύβο στοιβαγμένο πάνω του τότε θα εισάγει στο σύστημα την εξής κατάσταση :

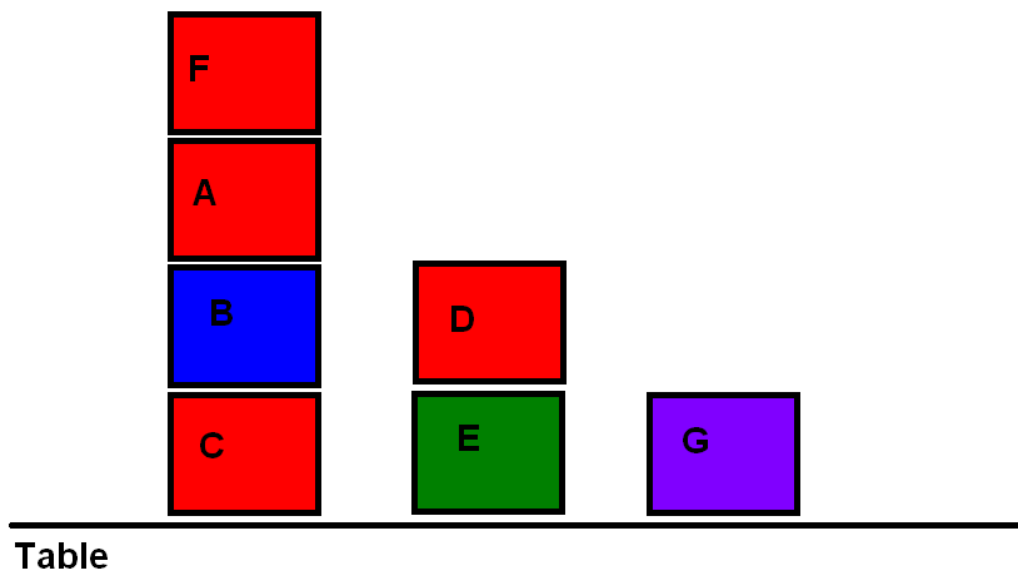
```
[on(BlockF, BlockG), tower ([BlockA, BlockB]), color (yellow), color (BlockC, yellow), color (BlockD, yellow), clear (BlockE)]
```

Στη συνέχεια ο χρήστης επιλέγει και τη στρατηγική δέσμευσης που θα ακολουθήσει ο πράκτορας προκειμένου να καταλήξει σε αυτή την τελική κατάσταση. Το πρόγραμμα μεταφράζει την δεδομένη κατάσταση σε αντίστοιχες επιθυμίες του πράκτορα από τις οποίες στη συνέχεια θα γίνει η επιλογή των προθέσεων.

Ο πράκτορας έχοντας τώρα διαμορφώσει το σύνολο των επιθυμιών του επιλέγει αυτές που θα ικανοποιήσει και εισάγει τις προθέσεις των ενεργειών που χρειάζονται να γίνουν για την ικανοποίηση αυτών των επιθυμιών. Η επιλογή των επιθυμιών γίνεται με τυχαίο τρόπο αφού ο πράκτορας κάθε φορά που επιλέγει μια επιθυμία διαγράφει και ένα καινούριο μονοπάτι προς το μέλλον που αρχικά δεν μπορεί να προσδιοριστεί. Ακριβώς για να δείξουμε αυτή τη λειτουργία δημιουργήσαμε μια σειρά από κατηγορήματα που αφορούν τη Random selection των επιθυμιών.

Ο πράκτορας σε αυτό το σημείο γνωρίζει τη στρατηγική δέσμευσης που θα ακολουθήσει καθώς και τις ενέργειες που έχει πρόθεση να εκτελέσει.

Έστω ο αρχικός κόσμος αποτυπώνεται ως εξής :

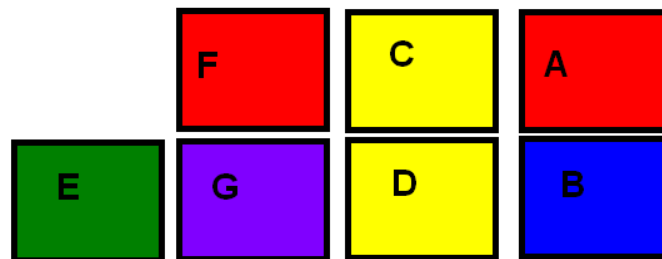


*Εικ4.Initial state.*

Σε περίπτωση που ο πράκτορας έχει οριστεί ως *blindly committed* (Type |: 1.), για κάθε πρόθεση εκτελεί την αντίστοιχη ενέργεια και αν η εκτέλεση είναι επιτυχής τότε αποδεσμεύεται από την πρόθεση αυτή ενώ αν η εκτέλεση αποτύχει ο πράκτορας συνεχίζει να είναι πιστός στην πρόθεση του μέχρι να ικανοποιηθεί. Δηλαδή οι προθέσεις που δεν ικανοποιούνται θα απασχολούν τον πράκτορα για πάντα.

`Blindly_committed (int (X)) :- does(X) AND retract (int (X)).`

Έτσι στο παράδειγμα που αναφέραμε τελικά θα εξεταστούν όλες οι επιθυμίες και θα ικανοποιηθούν όλες οι προθέσεις. Σε αυτή την περίπτωση πρέπει να σημειώσουμε ότι με όποια σειρά και αν επιλέξει ο πράκτορας να ικανοποιήσει τις προθέσεις του θα καταλήξει στην ίδια τελική κατάσταση εφόσον δεν υπάρχουν προθέσεις που συγκρούονται μεταξύ τους και επιπλέον ο χρωματισμός των δύο κύβων γίνεται την ίδια στιγμή.



**Table**

*Εικ5. Blindly committed/single minded agent.*

Αν ο πράκτορας οριστεί ως *single minded* (Type |: 2.) καλείται το αντίστοιχο κατηγορήμα που στην ουσία ακολουθεί την ίδια διαδικασία με τον *blindly committed* πράκτορα με τη μόνη διαφορά ότι αν τελικά έχουν επιλεγεί και εξεταστεί όλες οι επιθυμίες και δεν έχει αλλάξει το σύνολο των πεποιθήσεων έτσι ώστε τελικά η εκτέλεση να είναι επιτυχής, τότε ο πράκτορας αποδεσμεύεται.

`Single_minded (int (X)) :- (does (X) AND retract(X)) OR ((int (Y) AND X/= Y AND cut) OR (retract (X))).`

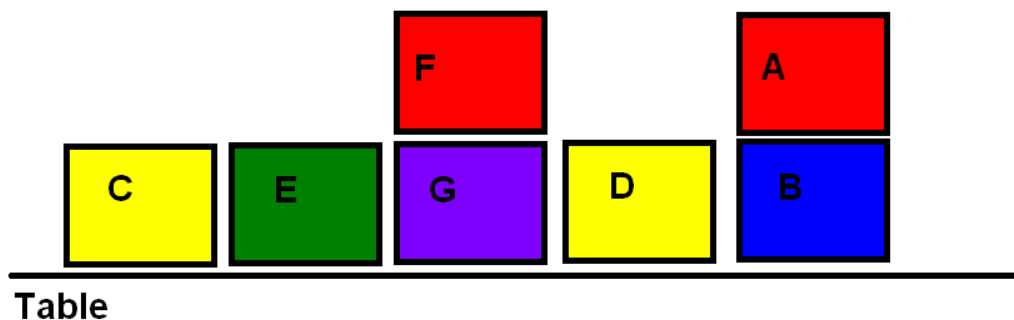
Ο πράκτορας στο συγκεκριμένο παράδειγμα θα καταλήξει στην ίδια κατάσταση με τον *Blindly committed* πράκτορα (Εικ5.). Σε αντίθεση όμως με τον *blindly committed* πράκτορα εάν δεν είχε εισαχθεί από την αρχή η επιθυμία του να χρωματίσει κίτρινους τους δύο κύβους αλλά ο πράκτορας αποκτούσε στην πορεία αυτή την πρόθεση τότε ο πράκτορας έχοντας αποδεσμευτεί ήδη από την πρόθεση του να δημιουργήσει ένα πύργο με κίτρινους κύβους δεν θα προβεί σε αυτή την ενέργεια. Επιπλέον αν ο *blindly committed* πράκτορας είχε αποκτήσει μια πρόθεση για την οποία δεν είχε την απαραίτητη πληροφορία για να την εκπληρώσει αλλά βασίζονταν μόνο σε πεποιθήσεις που συνεπάγονται της γνώσης (όπως

για παράδειγμα να αλλάξει το χρώμα του κύβου BlockE για τον οποίο συμπεραίνουμε ότι το χρώμα του είναι πράσινο, IF block(e) → color(e, green) AND block(e) THEN color(e, green)) θα έμενε πιστός σε αυτή του την πρόθεση για πάντα ενώ ο single minded πράκτορας έχοντας εξαντλήσει όλες τις εναλλακτικές διαδρομές τελικά θα αποδεσμεύονταν από αυτή.

Τέλος αν επιλεγεί open minded πράκτορας (Type |: 3.), οι προθέσεις παύουν να απασχολούν τον πράκτορα από την πρώτη κιόλας προσπάθεια εκτέλεσης είτε αυτή καταστεί επιτυχής είτε όχι.

Open\_minded (int (X)) :- (does(X) AND retract (int (X))) OR (retract (int (X))).

Έχοντας ο πράκτορας επιλέξει να ακολουθήσει τη σειρά [clear (BlockE) → color (yellow) → color (BlockC, yellow) → color (BlockD, yellow) → tower ([BlockA, BlockB]) → on(BlockF, BlockG)] παρατηρούμε πως δεν μπορεί να ικανοποιήσει την επιθυμία του για τη δημιουργία ενός πύργου με κύβους χρώματος κίτρινου εφόσον δεν υπάρχει κανένας κύβος αυτού του χρώματος. Αν η επιθυμία αυτή εξετάζονταν μετά την αλλαγή του χρώματος των κύβων BlockC και BlockD τότε θα είχε ικανοποιήσει όλες τις προθέσεις του. Σε αυτή όμως την περίπτωση ο πράκτορας αποδεσμεύεται από την πρόθεση αυτή χωρίς να εξετάσει αν τελικά μπορεί να καταστεί αληθής και ο τελικός κόσμος αποκτά την παρακάτω δομή :



Table

Εικ6. Open minded agent.

Οι διαφορές που παρουσιάζουν αυτές οι τρεις στρατηγικές δέσμευσης είναι πολύ μικρές και κυρίως προκύπτουν από τον τρόπο που ο πράκτορας επιλέγει να φτάσει στην τελική κατάσταση, δηλαδή την πορεία που θα ακολουθήσει στο χρόνο. Εάν για παράδειγμα όπως αναφέραμε και προηγουμένως ο open-minded πράκτορας εξέταζε με τέτοιο τρόπο τις προθέσεις του έτσι ώστε όλες να είναι εφικτές και τελικά να καταστούν αληθείς, τότε παρατηρούμε ότι και με τις τρεις στρατηγικές καταλήγουμε στο ίδιο επιθυμητό αποτέλεσμα.

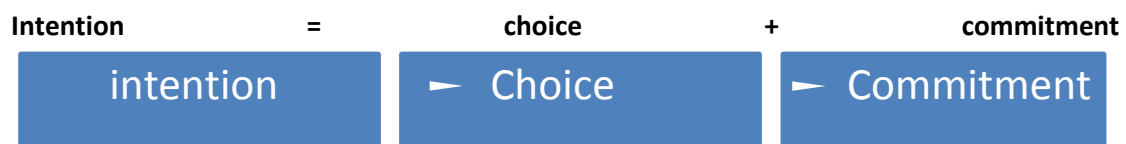
Με την ολοκλήρωση της κάθε ενέργειας που τελικά αλλάζει τον κόσμο ο πράκτορας αναθεωρεί τις πεποιθήσεις του και έχει επίγνωση όλων των ενεργειών που συνετέλεσαν σε αυτό.

Τέλος με τα κατηγορήματα `succeeds(event)` και `fails(event)` μπορούμε να παρατηρήσουμε πόσες φορές μια ενέργεια έχει αποτύχει να εκτελεστεί ή εκτελέστηκε με επιτυχία, ενώ με το κατηγορήμα `done(event)` κάθε ενέργεια που ικανοποίησε την αντίστοιχη πρόθεση προστίθεται στη βάση γνώσης του πράκτορα.

## 4.2 Intention Logic of Cohen and Levesque

Η λογική των Cohen & Levesque είναι επίσης μια Τροπική λογική με τελεστές που ορίζουν τις πεποιθήσεις και τους στόχους παράλληλα με την δυνατότητα έκφρασης των δράσεων δίνοντας έτσι την αίσθηση μιας linear time temporal λογικής που χρησιμοποιεί κάποιους επιπλέον τελεστές τροπικότητας. Σε αυτή τη λογική οι προθέσεις ορίζονται μέσω των «επίμονων» στόχων(persistent-goal) και αποδίδουν σε αυτές τους εξής λειτουργικούς κανόνες (*Wiebe van der Hoek*<sup>1,2</sup>):

- Οι προθέσεις θέτουν προβλήματα στον πράκτορα και με τη σειρά του ο πράκτορας αποφασίζει τον τρόπο επίλυσης.
- Οι προθέσεις θέτουν τα όρια και τους κανόνες για την αποδοχή νέων προθέσεων. Ένας πράκτορας δεν μπορεί να δεχτεί προθέσεις που έρχονται σε σύγκρουση με τις ήδη υπάρχουσες.
- Ο πράκτορας παρακολουθεί τις προσπάθειες για την επίτευξη των προθέσεων προκαλώντας ενδεχομένως ανασχεδιασμό.



Εικ7.Cohen and Levesque's definition of intentions.

### 4.2.1 Δεύτερη Εφαρμογή – Κόσμος των Κύβων

Στα πλαίσια της δεύτερης εφαρμογής τροπικών τελεστών στο περιβάλλον των Κύβων κάναμε την παραδοχή πως ο πράκτορας έχει την πλήρη γνώση της κατάστασης του κόσμου, δηλαδή οι πεποιθήσεις του αποτυπώνουν τον πραγματικό κόσμο, αλλά αντί για επιθυμίες ο πράκτορας υιοθετεί μια άλλη κατάσταση(mental state), τους στόχους. Ο πράκτορας πρέπει να πιστεύει ότι οι στόχοι του είναι εφικτοί και να επιτρέπει την εισαγωγή νέων στόχων μόνο εάν αυτοί είναι εφικτοί. Αυτή είναι και μια διακριτή διαφορά μεταξύ των

στόχων και των επιθυμιών, επομένως και των δύο αυτών αρχιτεκτονικών (BDI – Cohen & Levesque).

Η λογική αυτή περιγράφεται αρχικά με τέσσερις βασικούς τελεστές (*Philip R. Cohen & Hector J. Levesque, 1990*):

1. Happens a: Η ενέργεια a θα είναι η επόμενη που θα εκτελεστεί.
2. Done a (time): Η ενέργεια a μόλις έχει εκτελεστεί (και με ποια σειρά σε σχέση με τις υπόλοιπες).
3. Bel φ: Ο πράκτορας έχει την πεποίθηση ότι η πρόταση φ είναι αληθής.
4. Goal φ: Ο πράκτορας έχει ως στόχο να κάνει την πρόταση φ αληθή.

Οι πεποιθήσεις και οι στόχοι του πράκτορα υπακούουν σε όλους τους κανόνες της Τροπικής λογικής, τους σχετικούς με τις συμπεριφορές του πράκτορα (Logical Omniscience).

Επιπλέον οι πεποιθήσεις ικανοποιούν (*John-Jules Meyer, 2010*):

Την ιδιότητα των συνεπών πεποιθήσεων (D.)  $Bel\ p \rightarrow \neg Bel\ \neg\ p$

Την ιδιότητα negative introspection (5.)  $\neg Bel\ p \rightarrow Bel\ \neg\ Bel\ p$

Την ιδιότητα positive introspection (4.)  $Bel\ p \rightarrow Bel\ Bel\ p$

Την ιδιότητα realism  $Bel\ p \rightarrow Goal\ p$  (or  $Bel\ p \rightarrow \neg Goal\ \neg\ p$  )

Την ιδιότητα expected consequence  $(Goal\ p \wedge Bel\ (p \rightarrow q)) \rightarrow Goal\ q$

Η λογική αυτή αφήνει επίσης περιθώρια για τον καθορισμό χρονικών λεπτομερειών. Μερικοί κλασσικοί χρονικοί τελεστές που χρησιμοποιήσαμε και στον Κόσμο των Κύβων είναι ο τελεστής BEFORE p q, ο οποίος δηλώνει πως πριν καταστεί η πρόταση p αληθής θα γίνει αληθής η πρόταση q, ο τελεστής LATER p, ο οποίος εκφράζει το γεγονός ότι μια πρόταση p δεν ανήκει στις πεποιθήσεις του πράκτορα όμως στο μέλλον θα καταστεί αληθής και ο τελεστής Always φ. Ο τελεστής Always φ ουσιαστικά εκφράζει τον τροπικό τελεστή  $\square$  ενώ χαρακτηρίζουμε μια πρόταση με το τελεστή αυτό εάν δεν υπάρχει κάποια ακολουθία γεγονότων που θα οδηγήσει στο  $\neg\ \varphi$ . Στον Κόσμο των Κύβων χρησιμοποιούμε την έκφραση Always\_not(φ) ακριβώς για να εκφράσουμε το αντίθετο, δηλαδή ότι δεν υπάρχει κάποια ακολουθία γεγονότων που θα μπορέσει να καταστήσει την πρόταση φ αληθή τη δεδομένη στιγμή.

Όσον αφορά τους στόχους του πράκτορα υπάρχουν δύο κατηγορίες, οι στόχοι που κάποια στιγμή θα επιτευχθούν (achievement goals) και οι επίμονοι στόχοι (persistent goals). Στην εφαρμογή της λογικής αυτής οι στόχοι απλά δηλώνουν μια κατάσταση και όχι την εκτέλεση κάποιας ενέργειας. Αυστηρός κανόνας της εφαρμογής είναι ότι τελικά ο πράκτορας αποδεσμεύεται από όλους τους achievement goals δηλαδή δεν θεωρεί για πάντα πως οι στόχοι του είναι εφικτοί.

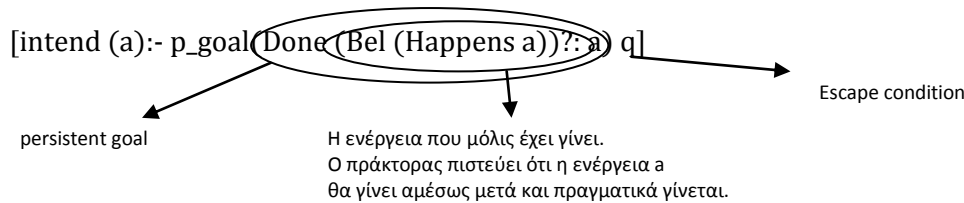
Στον Κόσμο μας ο πράκτορας δέχεται μια ακολουθία στόχων. Από αυτούς βρίσκει τους στόχους που μπορούν κάποια στιγμή να επιτευχθούν,

[ a\_goal (X):- LATER (X), not (BEL (X) ) ]

και στη συνέχεια ορίζει τους επίμονους στόχους σύμφωνα με τους οποίους θα αναθεωρηθούν οι πεποιθήσεις του και θα οριστούν οι τελικές προθέσεις του πράκτορα,

[p\_goal (X):- a\_goal (X), BEFORE (..belief revision..) (drop\_a\_goal)].

Όπως αναφέραμε και στην αρχή οι προθέσεις ορίζονται μέσω των persistent goals. Επομένως στην εφαρμογή μας ο πράκτορας έχει πρόθεση να κάνει μια ενέργεια a εάν το γεγονός ότι ο πράκτορας έχει την πεποίθηση ότι θα κάνει την πράξη αυτή πριν γίνει, αποτελεί έναν persistent goal.



Έστω για παράδειγμα ότι ο πράκτορας έχει πρόθεση να «καθαρίσει» το block A (clear\_off(BlockA)). Ο πράκτορας έχει έναν persistent goal να μεταβεί σε μία κατάσταση στην οποία ο πράκτορας έχει την πεποίθηση ότι έχει καθαρίσει το block A και πρόκειται να καθαρίσει το block A στη συνέχεια.

Χαρακτηριστικό λοιπόν της όλης διαδικασίας ενός πράκτορα της intention λογικής είναι ότι τα γεγονότα ακολουθούν μια γραμμική σειρά όσον αφορά το χρόνο και δημιουργούν ακολουθίες ανάλογα με τους στόχους και τις πεποιθήσεις του πράκτορα.

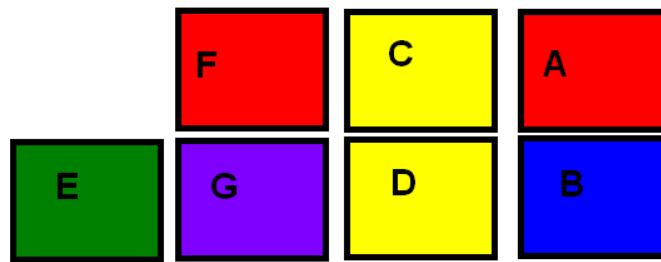
## 4.2.2 Εκτέλεση Δεύτερης Εφαρμογής

Στη δεύτερη εφαρμογή ο πράκτορας μπορεί να εκτελέσει τις ίδιες ενέργειες με την πρώτη εφαρμογή. Ωστόσο σε αυτή την περίπτωση έχουμε δύο mental states ενώ οι προθέσεις ορίζονται από τους στόχους.

Ο χρήστης δίνει μια τελική κατάσταση του κόσμου ομοίως με την πρώτη εφαρμογή. Το πρόγραμμα μεταφράζει τα δεδομένα που διαβάζει από τον χρήστη σε πιθανούς στόχους. Στη συνέχεια ο πράκτορας εξετάζει ποιοι από τους στόχους αυτούς μπορούν να πραγματοποιηθούν διαμορφώνοντας έτσι τους achievement goals όπως τους έχουμε ορίσει και παραπάνω. Για παράδειγμα εάν ο χρήστης έχει δηλώσει ότι στην τελική κατάσταση του κόσμου ο κύβος BlockA θα είναι κόκκινος όμως ο πράκτορας πιστεύει ήδη ότι ο κύβος BlockA είναι κόκκινος τότε αυτός ο πιθανός στόχος απορρίπτεται από τον πράκτορα.

Ο πράκτορας γνωρίζει πλέον ποιοι στόχοι του θα πραγματοποιηθούν κάποια στιγμή στο μέλλον και συνεχίζει διαμορφώνοντας τους persistent goals.

Στο παράδειγμα που μελετήσαμε και στην πρώτη εφαρμογή ο πράκτορας αναγνωρίζει όλους τους στόχους ως εφικτούς και ενεργεί αντίστοιχα. Παρατηρούμε πως και εδώ ο πράκτορας ουσιαστικά χρειάζεται «δύο κύκλους» εκτέλεσης, δηλαδή ενώ έχει κρίνει ότι μπορεί στο μέλλον να δημιουργήσει έναν πύργο με κίτρινους κύβους, αρχικά δεν μπορεί να πραγματοποιήσει τον στόχο αυτό και τον κατατάσσει στους επίμονους στόχους που θα εκτελεστεί μετά την εκτέλεση των επόμενων στόχων με τη σειρά που αυτοί υιοθετήθηκαν.



**Table**

*Εικ8.Cohen and Levesque's Final state (μετά την εκτέλεση όλων των στόχων και την επανεξέταση των 'επίμονων' στόχων).*

Έτσι ο πράκτορας έχει έναν persistent goal  $p$  εάν ο στόχος  $p$  είναι εφικτός, δηλαδή ανήκει στο σύνολο των achievement goal, αλλά επίσης και εάν ο πράκτορας πριν αποδεσμευτεί από τον στόχο αυτό είτε έχει την πεποίθηση ότι ο στόχος έχει πραγματοποιηθεί είτε έχει την πεποίθηση ότι ο στόχος δεν θα πραγματοποιηθεί ποτέ. Στην πρώτη περίπτωση προστίθεται στη βάση γνώσης του πράκτορα η πεποίθηση ότι ο στόχος  $p$  πραγματοποιήθηκε ( $Bel(p)$ ), ενώ στη δεύτερη περίπτωση προστίθεται ο στόχος  $p$  με τον χαρακτηρισμό του μη πραγματοποιήσιμου στόχου ( $Bel(Always\_not(p))$ ).

Εάν δηλαδή στο παράδειγμα μας μετά την εξέταση όλων των εφικτών στόχων από τον πράκτορα δεν μπορέσει να εκτελεστεί ο στόχος color (yellow) τότε ο πράκτορας θα αποκτήσει την πεποίθηση ότι δεν μπορεί να δημιουργηθεί ποτέ στο μέλλον ένας πύργος με κίτρινους κύβους και θα εισάγει στις πεποιθήσεις του το γεγονός :  $Bel(Always\_not(color(yellow)))$ .

Τέλος ο πράκτορας έχει αποκτήσει την πρόθεση για τον στόχο  $p$ . Ο πράκτορας ήδη πιστεύει ότι η αντίστοιχη ενέργεια που ικανοποιεί τον στόχο αυτό έχει γίνει και στη συνέχεια εκτελεί την ενέργεια αυτή εισάγοντας στη βάση γνώσης του πράκτορα το γεγονός αυτό μαζί με την χρονική σειρά με την οποία έγινε ( $Bel(done(Action,timestamp))$ ).

$Done(Action, Time) \vdash$

**Action** = on(BlockF, BlockG), **Time** = 1

**Action** = tower ([BlockA, BlockB]), **Time** = 2

**Action** = color (yellow) ...

**Action** = color (BlockC, yellow), **Time** = 3

**Action** = color (BlockD, yellow), **Time** = 4

**Action** = clear (BlockE), **Time** = 5

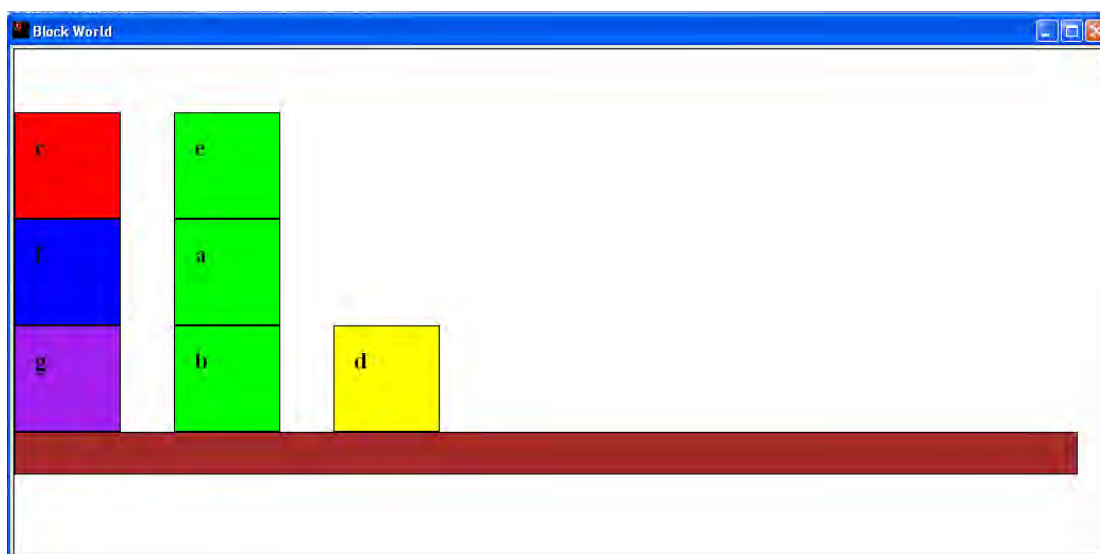
Δεύτερος κύκλος

**Action** = color (yellow), **Time** = 6

### 4.3 Σχετικά Με Την Υλοποίηση

Για την ανάπτυξη των παραπάνω εφαρμογών χρησιμοποιήσαμε την πλατφόρμα SWI-Prolog-Editor. Τα αντίστοιχα αξιώματα της κάθε λογικής αναπτύχθηκαν με τη μορφή κατηγορημάτων όπως αυτά ορίζονται στη γλώσσα προγραμματισμού Prolog ενώ χρειάστηκε να ορίσουμε και ορισμένους τελεστές που θα αναγνωρίζονται από το πρόγραμμα. Επιπλέον ως περιβάλλον του πράκτορα ορίσαμε τον Κόσμο των Κύβων, έναν από τους πιο διάσημους τομείς της τεχνητής νοημοσύνης, και συμπεριλάβαμε κάποια από τα βασικά χαρακτηριστικά του όπως η ύπαρξη όμοιων κύβων που μπορούν να στοιβαχτούν ο ένας πάνω στον άλλον ή πάνω σε ένα τραπέζι και την ύπαρξη ενός ρομπότ-πράκτορα που αναλαμβάνει να διαμορφώσει τον κόσμο αυτόν εκτελώντας μία ενέργεια σε κάθε χρονική στιγμή.

Τα αποτελέσματα εμφανίζονται στο χρήστη μέσω ενός παραθύρου, κάθε φορά που καλείται το κατηγορημα `create_world`. Για το σχεδιασμό αυτού του παραθύρου χρησιμοποιήσαμε το εργαλείο XPCE της Prolog. Πρόκειται για ένα εργαλείο που αναφέρεται στο σχεδιασμό γραφικού περιβάλλοντος για Prolog-εφαρμογές. Ένα στιγμιότυπο του κόσμου όπως αυτό αναπαριστάται με XPCE φαίνεται στην *εικ.9*.



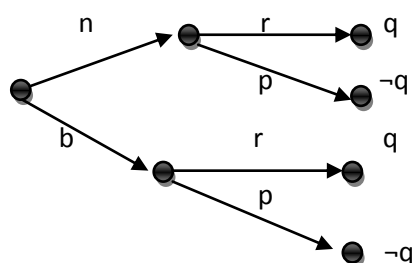
Εικ9.Εμφάνιση Κόσμου με XPCE.



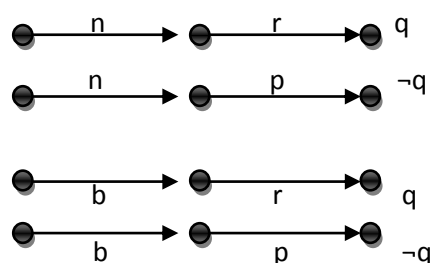
# 5 Επίλογος

## 5.1 Σύνοψη και Συμπεράσματα

Αρχικά αν μελετήσουμε τις δύο προσεγγίσεις ως προς τον τρόπο που εκτυλίσσονται τα γεγονότα όσον αφορά το χρόνο, έχουμε να τονίσουμε μια βασική διαφορά. Η Cohen & Levesque προσέγγιση αντιλαμβάνεται τα γεγονότα ως γραμμικές ακολουθίες και βασίζεται περισσότερο σε ένα χρονικό πλαίσιο και όχι τόσο σε ένα πλαίσιο δράσεων έτσι ώστε να καταλήξει στην τελική επιθυμητή κατάσταση. Αντίθετα στην προσέγγιση BDI η συμπεριφορά του πράκτορα βασίζεται σε μια δομή δένδρου όπου κάθε μονοπάτι του αντιπροσωπεύει μια πιθανή πορεία του πράκτορα. Στο μοντέλο αυτό δηλαδή ο πράκτορας μπορεί να εκφράσει τις πεποιθήσεις, τις επιθυμίες και τις προθέσεις του στην πάροδο του χρόνου και όχι πάνω σε προδιαγεγραμμένες γραμμές χρόνου.



Εικ10.1. Time tree.



Εικ10.2. Time line.

Στην προσέγγιση BDI οι προθέσεις παίζουν το ρόλο των μελλοντικών μονοπατιών που ο πράκτορας επιλέγει να ακολουθήσει. Η βασική λοιπόν διαφορά της με την προσέγγιση του Cohen & Levesque αφορά το γεγονός ότι οι προθέσεις αποτελούν μια βασική συμπεριφορά του πράκτορα και ότι η διαδικασία αναθεώρησης τους αφορά ουσιαστικά τη μελλοντική δέσμευση που θα ακολουθήσει ο πράκτορας.

Οι στόχοι στη λογική των Cohen & Levesque εκφράζονται με έναν πιο πρωτόγονο τρόπο και χωρίζονται σε δύο κατηγορίες ακριβώς γιατί διαφορετικά δεν θα μπορούσε να εκφραστεί το κατά ποσό ένας στόχος είναι εφικτός ή μόνιμος (achievement or persistent) για τον πράκτορα. Επιπλέον οι προθέσεις ορίζονται έτσι ώστε να υποδεικνύουν τις ενέργειες που θα γίνουν και θα οδηγήσουν στην κατάσταση στόχος. Αντίθετα στη BDI λογική η διαχώριση μεταξύ των επιθυμιών και των προθέσεων του πράκτορα επιτρέπει μια πιο ευέλικτη λειτουργία.

Όσον αφορά τις στρατηγικές δέσμευσης που συναντάμε στις δύο αυτές προσεγγίσεις πρέπει αρχικά να τονίσουμε ότι οι στρατηγικές δέσμευσης και η διαδικασία αναθεώρησης των πεποιθήσεων, των προθέσεων και των στόχων είναι βασικά χαρακτηριστικά των λογικών πρακτόρων. Θα μπορούσαμε να πούμε πως οι επίμονοι στόχοι έτσι όπως εκφράζονται στην προσέγγιση Cohen & Levesque αποτελούν την αντίστοιχη στρατηγική δέσμευσης του blindly-committed πράκτορα. Ωστόσο στην προσέγγιση του

Cohen & Levesque ο πράκτορας μπορεί να ορίζει έναν στόχο ως μη εφικτό ή το αντίθετο και να δεσμεύεται για αυτόν σε όλη τη διάρκεια της λειτουργίας του ενώ αντίθετα η BDI προσέγγιση προσπαθεί να εστιάσει περισσότερο στην αναθεώρηση των προθέσεων του πράκτορα και όχι τόσο στην απόλυτη δέσμευση. Συγκεκριμένα και στις τρεις περιπτώσεις στρατηγικών ο πράκτορας ελέγχει ξανά τις προθέσεις του και το αν αυτές τελικά μπορούν να εκπληρωθούν.

Επιπλέον στην δεύτερη υλοποίηση παρατηρούμε πως ο πράκτορας θεωρεί ότι η εκτέλεση της ενέργειας που θα ικανοποιήσει την αντίστοιχη πρόθεση του πράκτορα έχει γίνει πριν ουσιαστικά εκτελεστεί και συμπεριλαμβάνει το γεγονός στις πεποιθήσεις του. Αν τελικά ο πράκτορας έχει θεωρήσει σωστά ότι η συγκεκριμένη πρόθεση θα υλοποιηθεί έχει να κάνει με το κατά πόσο σωστά έχει εκτιμήσει τον αντίστοιχο στόχο ως εφικτό. Αντίθετα στην πρώτη υλοποίηση ο πράκτορας θεωρεί ότι η πρόθεση του έχει ικανοποιηθεί παρά μόνο όταν ολοκληρωθούν όλες οι απαραίτητες ενέργειες.

Ωστόσο και στις δύο εφαρμογές υποθέτουμε ότι ο πράκτορας είναι παντογνώστης, δηλαδή οι πεποιθήσεις που έχει αναπτύξει για τον κόσμο του είναι αληθείς. Στην αντίθετη περίπτωση ο πράκτορας λαμβάνει λάθος δεδομένα και ενεργεί λανθασμένα. Ενώ επιπλέον και στις δύο εφαρμογές ο πράκτορας παρουσιάζει *motivational attitudes* και πεποιθήσεις όπως αυτές ορίζονται στην Επιστημική λογική.

## 5.2 Μελλοντικές Επεκτάσεις

Σε αυτό το κεφάλαιο θα αναφερθούμε κυρίως σε επεκτάσεις και μελλοντικά πεδία έρευνας που αφορούν τα frameworks τα οποία αναφέραμε και εισάγαμε στις παραπάνω εφαρμογές.

Αρχικά μία σημαντική επέκταση που θα δημιουργήσει μια πιο ολοκληρωμένη εικόνα όλων των χαρακτηριστικών των εν λόγω προσεγγίσεων είναι η αντικατάσταση του ενός πράκτορα από ένα πολυπρακτορικό σύστημα στο οποίο οι πράκτορες θα πρέπει να επικοινωνούν μεταξύ τους και ανάλογα με τις προθέσεις που υιοθετούν να ενεργούν και να ενημερώνουν όλους τους υπόλοιπους πράκτορες έτσι ώστε να αναθεωρούν και αυτοί με τη σειρά τους τις πεποιθήσεις τους. Σε μία τέτοια πολυπρακτορική υλοποίηση πρέπει να ληφθούν και τα κατάλληλα μέτρα έτσι ώστε η ενέργεια του κάθε πράκτορα να γίνεται με συνέπεια και να μην αναιρεί την δράση των υπολοίπων.

Στις εφαρμογές μας θεωρήσαμε ότι ο πράκτορας έχει διαμορφώσει πεποιθήσεις που απεικονίζουν πιστά τον κόσμο του. Ως μελλοντική επέκταση θα μπορούσαμε να προγραμματίσουμε τον πράκτορα έτσι ώστε να μπορεί να διαμορφώνει μόνος του τις αρχικές πεποιθήσεις που αφορούν την αναπαράσταση του κόσμου με τους κατάλληλους αισθητήρες.

Ένα ακόμη χαρακτηριστικό θέμα με το οποίο θα ήταν ιδιαίτερα ενδιαφέρον να ασχοληθούμε είναι το ενδεχόμενο ο χρήστης να εισάγει, σε οποιαδήποτε από τις παραπάνω προσεγγίσεις, καταστάσεις του κόσμου που στην ουσία δεν μπορούν να ισχύουν διότι συγκρούονται. Για παράδειγμα εάν ο χρήστης εισάγει ως τελική κατάσταση ο κύβος

BlockA να βρίσκεται πάνω στον κύβο BlockB αλλά ταυτόχρονα και πάνω στο τραπέζι, το σύστημα θα πρέπει να μπορεί να αναγνωρίζει το λάθος και να δρα αντίστοιχα.

Επιπλέον η εισαγωγή κάποιων βελτιώσεων σχετικές με τον τρόπο που ο χρήστης εισάγει την επιθυμητή τελική κατάσταση και αφορούν κυρίως το γραφικό κομμάτι των εφαρμογών θα αποτελούσε ένα ακόμη πεδίο ανάπτυξης των εφαρμογών μας. Με το εργαλείο XPCF της Prolog μπορούμε να δημιουργήσουμε μια διεπαφή για τον χρήστη και ένα γραφικό περιβάλλον πιο προσιτό από αυτό της κονσόλας, στο οποίο θα εισάγονται τα δεδομένα. Ωστόσο κάτι τέτοιο θα ήταν αρκετά πολύπλοκο και ενδεχομένως θα μας αποπροσανατόλιζε από τον αρχικό στόχο που θέσαμε για τις εφαρμογές μας.

Τέλος θα πρέπει να αναφέρουμε ένα επιπλέον βασικό framework που αφορά των προγραμματισμό ορθολογικών πρακτόρων, το 'KARO framework' (*B. van Linder & J.-J. Ch. Meyer & W. van der Hoek, 1997*). Το σύστημα αυτό συνδυάζει τις έννοιες της γνώσης και της πεποίθησης με τις ικανότητες, τις ευκαιρίες και τα αποτελέσματα που παρουσιάζει ένας πράκτορας. Ωστόσο σε αυτό το framework οι διανοητικές αυτές συμπεριφορές διαμορφώνονται δυναμικά κάτι που απαιτεί ίσως μια εντελώς διαφορετική αντιμετώπιση όσον αφορά τον προγραμματισμό ενός πράκτορα.

## 6 Βιβλιογραφία

---

1. Brian F. Chellas (1980). *Modal Logic: An introduction*. Cambridge University Press.
2. Wiebe van der Hoek <sup>1,2</sup>. Logical Foundations of Agent-Based Computing. <sup>1</sup>Institute of Information and Computing Sciences, Department of Philosophy & <sup>2</sup>Department of Computer Science, The University of Liverpool United Kingdom.
3. Edward N. Zalta (1995). *Basic Concepts in Modal Logic*. Center for the Study of Language and Information, Stanford University.
4. Alwen Tiu (2009). *Introduction to Logic* [Πανεπιστημιακές σημειώσεις]. The Australian National University, Summer School in Logic and Learning: 26 January – 6 February 2009, Canberra.
5. Philip R. Cohen & Hector J. Levesque (1990). Intention is Choice with Commitment. *Artificial Intelligence*, 42, 213-261.
6. John-Jules Meyer (2002). *Intelligent Agents: Issues and Logics*. Utrecht University, Information and Computing Sciences.
7. John-Jules Meyer (2010). *Rao&Georgeff's BDI* [Πανεπιστημιακές σημειώσεις]. Utrecht University: Intelligent Agents.
8. John-Jules Meyer (2010). *Cohen&Levesque* [Πανεπιστημιακές σημειώσεις]. Utrecht University: Intelligent Agents.
9. Camilo Thorne (2005). *The BDI Model of Agency and BDI Logics* [Πανεπιστημιακές σημειώσεις]. Free University of Bolzano.
10. B. van Linder & J.-J. Ch. Meyer & W. van der Hoek (1997). *Formalizing Motivational Attitudes of Agents Using the KARO Framework*. Utrecht University: Information and Computing Sciences.
11. Άννα Φιλίππου (2011). *Τροπικές Λογικές*. [Πανεπιστημιακές σημειώσεις]. Πανεπιστήμιο Κύπρου : 'Λογική στην Πληροφορική', Χειμερινό εξάμηνο 2011.