

Ατομική Διπλωματική Εργασία

**Ανάπτυξη λογισμικού για την προσομοίωση
βιολογικών συστημάτων μοντελοποιημένα
σε μια άλγεβρα διεργασιών.(8)**

Κυριακή Χατζηκώστα

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2010

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ανάπτυξη λογισμικού για την προσομοίωση
βιολογικών συστημάτων μοντελοποιημένα
σε μια άλγεβρα διεργασιών.**

Κυριακή Χατζηκώστα

Επιβλέπουσα Καθηγήτρια

κυρία Φιλίππου Άννα

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

Μάιος 2010

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά την κυρία Άννα Φιλίππου που μου έδωσε την δυνατότητα να εργαστώ πάνω στο αντικείμενο της παρούσας διπλωματικής. Με την διπλωματική αυτή είχα την ευκαιρία να μάθω περισσότερα για την χρήση και τις δυνατότητες των αλγεβρών διεργασιών σε θέματα που αφορούν την μοντελοποίηση βιολογικών συστημάτων. Παράλληλα είχα την ευκαιρία να κινηθώ στο χώρο της βιο-πληροφορικής, αναπτύσσοντας ένα λογισμικό με την χρήση του οποίου κάποιος μπορεί να προσομοιώσει βιολογικά συστήματα μοντελοποιημένα στην άλγεβρα διεργασιών BioAmbients. Ευχαριστώ λοιπόν την κυρία Άννα Φιλίππου τόσο για την ευκαιρία που μου έδωσε να κινηθώ στο χώρο της βιο-πληροφορικής όσο και για την βοήθεια και την καθοδήγηση της για την ολοκλήρωση της διπλωματικής αυτής εργασίας.

Ευχαριστώ,

Κυριακή Χατζηκώστα.

Περίληψη

Η επιστήμη της πληροφορικής θέτει τις υπηρεσίες της στον κόσμο των βιολόγων αναπτύσσοντας λογισμικά που απαλλάσσουν τους ερευνητές βιολόγους από την ανάγκη να οργανώσουν και χρησιμοποιήσουν ένα τεράστιο δεδομένων, τον οποίο αντλούν από τα βιολογικά συστήματα. Έτσι και το λογισμικό που έχει αναπτυχθεί στα πλαίσια αυτής της διπλωματικής εργασίας. Στο έγγραφο θα βρείτε τη μελέτη που προηγήθηκε της ανάπτυξης του λογισμικού καθώς και τα κύρια χαρακτηριστικά του. Συγκεκριμένα το πρώτο κεφάλαιο είναι μια εισαγωγή με το σκοπό και το κίνητρο αυτής της διπλωματικής εργασίας. Ακολουθούν οι άλγεβρες διεργασιών και οι τεχνικές αναπαράστασης βιολογικών συστημάτων που μελετήθηκαν στο δεύτερο κεφάλαιο. Το τρίτο κεφάλαιο είναι μια εφαρμογή ενός παραδείγματος στις άλγεβρες διεργασιών και τις τεχνικές που μελετήθηκαν και αναφέρονται στο κεφάλαιο δύο. Στο κεφάλαιο τέσσερα παρουσιάζονται τα κριτήρια επιλογής της άλγεβρας διεργασιών καθώς και μια επιπλέον ανάλυση της άλγεβρας που έχει επιλεγεί. Στο κεφάλαιο πέντε θα βρείτε τα χαρακτηριστικά του λογισμικού που έχει αναπτυχθεί ενώ στο κεφάλαιο έξι θα βρείτε ένα παράδειγμα βιολογικού συστήματος που συναντούμε στον πραγματικό κόσμο, μοντελοποιημένο στην άλγεβρα διεργασιών BioAmbients. Στο κεφάλαιο επτά θα βρείτε σε ποιούς τομείς μπορεί να χρησιμοποιηθεί το λογισμικό που αναπτύχθηκε καθώς και τις προοπτικές του στο μέλλον. Τέλος καταγράφονται κάποια γενικά συμπεράσματα τόσο για τη χρήση της άλγεβρας διεργασιών BioAmbients, όσο και για την χρήση του λογισμικού που έχει αναπτυχθεί.

Περιεχόμενα

Κεφάλαιο 1 Εισαγωγή

- 1.1 Σκοπός της διπλωματικής εργασίας. 1
- 1.2 Κίνητρο. 1

Κεφάλαιο 2 Τεχνικές και μέθοδοι μοντελοποίησης βιολογικών συστημάτων που μελετήθηκαν.

- 2.1 Τί είναι οι άλγεβρες διεργασιών. 3
- 2.2 Μοντελοποίηση βιολογικών συστημάτων με τη χρήση της άλγεβρας διεργασιών Beta-binders. 4
- 2.3 Μοντελοποίηση βιολογικών συστημάτων με τη χρήση Petri Nets. 7
- 2.4 Μοντελοποίηση βιολογικών συστημάτων με τη χρήση της άλγεβρας διεργασιών BioAmbients. 8

Κεφάλαιο 3 Μοντελοποίηση παραδείγματος στις τρεις μεθόδους μοντελοποίησης συστημάτων που μελετήθηκαν.

- 3.1 Μοντελοποίηση παραδείγματος με τη χρήση άλγεβρας διεργασιών σε Beta Binders. 12
- 3.2 Μοντελοποίηση παραδείγματος με τη χρήση Petri Nets. 16
- 3.3 Μοντελοποίηση παραδείγματος με τη χρήση της άλγεβρας διεργασιών BioAmbients. 18

Κεφάλαιο 4 Επιλογή της άλγεβρας διεργασιών που θα αναγνωρίζει το λογισμικό.

- 4.1 Κριτήρια επιλογής άλγεβρας διεργασιών. 26

4.2 Επιπλέον ανάλυση της επιλεγόμενης άλγεβρας διεργασιών. 27

Κεφάλαιο 5 Ανάπτυξη λογισμικού.

5.1 Ανάλυση απαιτήσεων και προδιαγραφών του λογισμικού. 33

5.2 Επιλογή γλώσσας προγραμματισμού. 34

5.3 Χαρακτηριστικά λογισμικού και δυνατότητες χρήσης. 35

Κεφάλαιο 6 Μοντελοποίηση δημιουργίας και ενεργοποίησης πρωτεϊνών σε BioAmbients.

6.1 Περιγραφή της διαδικασίας στον πραγματικό κόσμο. 41

6.2 Περιγραφή της διαδικασίας στον κόσμο των BioAmbients. 41

6.3 Μοντελοποίηση του βιολογικού συστήματος σε BioAmbients. 45

Κεφάλαιο 7 Χρήση και Προοπτικές του λογισμικού.

7.1 Χρήση του λογισμικού στο χώρο των βιολογικών επιστημών. 52

7.2 Μελλοντικές βελτιστοποιήσεις του λογισμικού. 53

Κεφάλαιο 8 Συμπεράσματα.

8.1 Συμπεράσματα για τη χρήση της άλγεβρας διεργασιών BioAmbients. 54

8.2 Συμπεράσματα για τη χρήση του λογισμικού για την προσομοίωση βιολογικών συστημάτων. 55

Βιβλιογραφία: 56

Κεφάλαιο 1

Εισαγωγή

1.1 Σκοπός της διπλωματικής εργασίας.

1.2 Κίνητρο.

1.1 Σκοπός της διπλωματικής εργασίας.

Η παρούσα διπλωματική εργασία έχει να κάνει με τη δημιουργία ενός λογισμικού που θα προσομοιώνει αντιδράσεις που συναντούμε σε βιολογικά συστήματα. Συγκεκριμένα τα βιολογικά αυτά συστήματα θα δίνονται σε μορφή κειμένου στο πρόγραμμα και μοντελοποιημένα σε μια άλγεβρα διεργασιών. Το πρόγραμμα πρέπει να αναγνωρίζει την σύνταξη της άλγεβρας διεργασιών BioAmbients και να προσομοιώνει συστήματα μοντελοποιημένα σε αυτή την άλγεβρα διεργασιών.

1.2 Κίνητρο.

Η μοντελοποίηση βιολογικών οργανισμών για την ανάλυση της συμπεριφοράς τους είναι ένα τεράστιο θέμα που ταλαιπωρεί τον κλάδο των βιολογικών επιστημών, καθώς οι βιολόγοι χρειάζεται να οργανώσουν ένα τεράστιο όγκο πληροφοριών, να τον κατανοήσουν και να τον χρησιμοποιήσουν κατά τη διάρκεια των ερευνών τους.

Η πολλά υποσχόμενη προσέγγιση των βιολογικών συστημάτων μέσω των αλγεβρών διεργασιών επιτρέπει την μοντελοποίηση και προσομοίωση της δυναμικής συμπεριφοράς τους. Ο συνδυασμός των δυνατοτήτων των αλγεβρών διεργασιών και της υπολογιστικής δύναμης των υπολογιστών στις μέρες μας, με παρακίνησε στη δημιουργία ενός λογισμικού, που θα παίρνει σαν είσοδο του ένα βιολογικό σύστημα μοντελοποιημένο σε μια άλγεβρα διεργασιών και θα παρουσιάζει στο χρήστη κάθε πιθανή αντίδραση και την δυσδιάστατη υπόσταση του βιολογικού συστήματος ανά πάσα στιγμή. Επίσης του παρέχεται η δυνατότητα να προσομοιώσει κάποια αντίδραση που επιλέγει ο ίδιος, ή μία τυχαία αντίδραση, και να παρακολουθήσει τις αλλαγές που επιφέρει αυτή η αντίδραση στο βιολογικό σύστημα που εξετάζει.

Κεφάλαιο 2

Τεχνικές και μέθοδοι μοντελοποίησης βιολογικών συστημάτων που μελετήθηκαν.

2.1 Τί είναι οι άλγεβρες διεργασιών.

2.2 Μοντελοποίηση βιολογικών συστημάτων με τη χρήση της άλγεβρας διεργασιών Beta-binders.

2.3 Μοντελοποίηση βιολογικών συστημάτων με τη χρήση Petri Nets.

2.4 Μοντελοποίηση βιολογικών συστημάτων με τη χρήση της άλγεβρας διεργασιών BioAmbients.

2.1 Τί είναι οι άλγεβρες διεργασιών.

Οι άλγεβρες διεργασιών είναι ένας μαθηματικοποιημένος τρόπος να εκφράζουμε την συμπεριφορά παράλληλων συστημάτων. Για να την ορίσουμε χρειαζόμαστε την σύνταξη με βάση την οποία θα κτίζουμε το σύστημα, του τελεστές που θα μας επιτρέπουν να ορίσουμε την επικοινωνία μεταξύ των διεργασιών, και τους κανόνες απλοποίησης που θα εφαρμόζονται μετά από κάποια επικοινωνία μεταξύ των διεργασιών. Η χρήση τους για την μοντελοποίηση βιολογικών συστημάτων ήταν αναμενόμενη καθώς θα βιολογικά συστήματα αποτελούν παράδειγμα παράλληλων συστημάτων και ξεκίνησε σχετικά πρόσφατα.

Για την επιλογή της άλγεβρας διεργασιών που θα χρησιμοποιούσα για την μοντελοποίηση των βιολογικών συστημάτων και κατ επέκταση αυτή που θα αναγνώριζε το λογισμικό σαν είσοδο χρειάστηκε να μελετήσω κάποιες άλγεβρες διεργασιών και τεχνικές μοντελοποίησης βιολογικών συστημάτων, να τις κατανοήσω, να τις συγκρίνω και να επιλέξω αυτή που φαίνεται να έχει καλές προοπτικές και είναι εύκολη στην κατανόηση και στη χρήση . Εύκολη στην κατανόηση χρειάζεται να είναι καθώς στόχος μας είναι να βοηθήσουμε τους βιολόγους να μοντελοποιούν πολύπλοκα βιολογικά συστήματα και να τα προσομοιώνουν. Έτσι έπρεπε να επιλέξω μια άλγεβρα διεργασιών που μοντελοποιεί βιολογικούς οργανισμούς με τέτοιο τρόπο ώστε το μοντέλο αυτό να μην διαφέρει κατά πολύ από τα πραγματικά δεδομένα και την δομή του βιολογικού συστήματος.

Συγκεκριμένα μελέτησα τα εξής:

- Μοντελοποίηση βιολογικών συστημάτων με τη χρήση της άλγεβρας διεργασιών Beta-binders
- Μοντελοποίηση βιολογικών συστημάτων με τη χρήση Petri Nets.
- Μοντελοποίηση βιολογικών συστημάτων με τη χρήση της άλγεβρας διεργασιών BioAmbients.

Ακολουθεί περεταίρω ανάλυση των τεχνικών αυτών.

2.2 Μοντελοποίηση βιολογικών συστημάτων με τη χρήση της άλγεβρας διεργασιών Beta-binders.

Η άλγεβρα αυτή αντιμετωπίζει τα οργανικά μόρια ενός συστήματος ως τα βασικά δομικά συστατικά, που συνενώνονται και σχηματίζουν πιο πολύπλοκα μόρια, όπως για παράδειγμα οι πρωτεΐνες, που σχηματίζονται από αμινοξέα, που βρίσκονται τοποθετημένα με μια συγκεκριμένη σειρά και αναδιπλώνονται με τέτοιο τρόπο ώστε μας δίνουν μια συγκεκριμένη πρωτεΐνη.

Οι αντιδράσεις σε ένα τέτοιο σύστημα μοντελοποιούνται και προσομοιώνονται ως μια επικοινωνία πάνω σε ένα κανάλι μεταξύ των αντιδρώντων μορίων.

Υπάρχουν τρία είδη αντιδράσεων: bimolecular: που γίνεται μεταξύ δύο διαφορετικών μορίων M_1 και M_2 , homodimerization: όπου δύο πανομοιότυπα μόρια M συνδέονται, και monomolecular: όπου έχουμε αλλαγή στη δομή ενός μορίου M .

Για την αναπαράσταση ενός μορίου γράφουμε $\beta(x, \Gamma)$ [διεργασίες], αν το μόριο έχει ενεργό κανάλι x μέσω του οποίου μπορεί να επικοινωνήσει με άλλα μόρια, ή $\beta^h(x, \Gamma)$ [διεργασίες], αν το μόριο δεν μπορεί να επικοινωνήσει με κάποιο άλλο μόριο. Ως x ορίζουμε το κανάλι πάνω στο οποίο μπορεί να επικοινωνήσει το μόριο και ως Γ τους τύπους του καναλιού. Για τον ορισμό των διεργασιών χρησιμοποιείται η σύνταξη της π -calculus.

Ακολουθούν η σύνταξη, τα αξιώματα και οι κανόνες απλοποίησης της άλγεβρας διεργασιών Beta-binders.

$P ::=$	
nil	Inactive
$\pi.P$	Action π has to be fired before executing P
$(P Q)$	Two parallel sub-processes P and Q
$\nu y.P$	y is a static binder of ν in P
$! \pi.P$	Multiple instances of guarded process $\pi.P$
$\pi ::=$	
$x(y)$	Receives over x a name y , y is a binder for the prefix
$\bar{x}(z)$	Sends name z on link x
hide(x)	x is a not available elementary beta binder
unhide(x)	x is an available elementary beta binder
expose(x)	Adds to the box the elementary beta binder $\beta(x, \Gamma)$

Εικόνα 2.1 Σύνταξη της άλγεβρας διεργασιών Beta-Binders

$$\begin{array}{l}
\text{(intra)} \quad \frac{P \equiv \nu \tilde{u} ((x(w), r_x). P_1 \mid (\bar{x}(z), r_x). P_2 \mid P_3)}{B \equiv \mathbf{B} [P] \xrightarrow{i; B; r_x \times n_I \times n_O; (1, 1)} \mathbf{B} [P_1 \mid P_2 \mid P_3]} \begin{cases} n_I = 1 + \text{In}_x(P_3) \\ n_O = 1 + \text{Out}_x(P_3) \end{cases} \\
\text{(hide)} \quad \frac{P \equiv \nu \tilde{u} ((\text{hide}(x), r_x). P_1 \mid P_2)}{B \equiv \mathbf{B}^* \beta(x, \Gamma) [P] \xrightarrow{h; B; r_x \times n_H; (1, 1)} \mathbf{B}^* \beta^h(x, \Gamma) [\nu \tilde{u} (P_1 \mid P_2)]} \\
\text{provided } x \notin \tilde{u} \text{ and } n_H = 1 + \text{Hide}_x(P_2) \\
\text{(unhide)} \quad \frac{P \equiv \nu \tilde{u} ((\text{unhide}(x), r_x). P_1 \mid P_2)}{B \equiv \mathbf{B}^* \beta^h(x, \Gamma) [P] \xrightarrow{u; B; r_x \times n_U; (1, 1)} \mathbf{B}^* \beta(x, \Gamma) [\nu \tilde{u} (P_1 \mid P_2)]} \\
\text{provided } x \notin \tilde{u} \text{ and } n_U = 1 + \text{Unhide}_x(P_2) \\
\text{(expose)} \quad \frac{P \equiv \nu \tilde{u} ((\text{expose}(x, \Gamma), r_x). P_1 \mid P_2)}{B \equiv \mathbf{B} [P] \xrightarrow{e; B; r_x; (1, 1)} \mathbf{B} \beta(x, \Gamma) [\nu \tilde{u} (P_1 \mid P_2)]} \\
\text{provided } x \notin \tilde{u}, x \notin \text{sub}(\mathbf{B}), x \notin \Gamma \text{ and } x \notin \text{fn}(P_2) \\
\text{(inter)} \quad \frac{P \equiv \nu \tilde{u} ((x(w), r_x). P_1 \mid P_2) \quad Q \equiv \nu \tilde{v} ((\bar{y}(z), r_y). Q_1 \mid Q_2)}{B \equiv \mathbf{B}_1 [P] \parallel \mathbf{B}_2 [Q] \xrightarrow{l; B; \alpha(\Gamma, \Delta); (1, 1)} \mathbf{B}_1 [P'] \parallel \mathbf{B}_2 [Q']} \\
\text{provided that (C1) holds} \\
\text{(inter}_h\text{)} \quad \frac{P \equiv \nu \tilde{u} ((x(w), r_x). P_1 \mid (\bar{y}(z), r_y). P_2 \mid P_3)}{B \equiv \mathbf{B} [P] \parallel \mathbf{B} [P] \xrightarrow{h; B; \alpha(\Gamma, \Delta)/2; (2, 1)} \mathbf{B} [Q_1] \parallel \mathbf{B} [Q_2]} \\
\text{provided that (C2) holds} \\
\text{(join)} \quad B \equiv \mathbf{B}_1 [P_1] \parallel \mathbf{B}_2 [P_2] \xrightarrow{J; B; c_j; (1, 1)} \mathbf{B}' [P_1 \sigma_1 \mid P_2 \sigma_2] \\
\text{provided that } f_{\text{join}}(\mathbf{B}_1, \mathbf{B}_2, P_1, P_2) = (\mathbf{B}', \sigma_1, \sigma_2, c_j) \text{ and } \mathbf{B}_1 [P_1] \neq \mathbf{B}_2 [P_2] \\
\text{(join}_h\text{)} \quad B \equiv \mathbf{B}_1 [P_1] \parallel \mathbf{B}_1 [P_1] \xrightarrow{J_h; B; c_j/2; (2, 1)} \mathbf{B}' [P_1 \sigma_1 \mid P_1 \sigma_2] \\
\text{provided that } f_{\text{join}}(\mathbf{B}_1, \mathbf{B}_1, P_1, P_1) = (\mathbf{B}', \sigma_1, \sigma_2, c_j) \\
\text{(split)} \quad B \equiv \mathbf{B} [P_1 \mid P_2] \xrightarrow{S; B; c_s; (1, 1)} \mathbf{B}_1 [P_1 \sigma_1] \parallel \mathbf{B}_2 [P_2 \sigma_2] \\
\text{provided that } f_{\text{split}}(\mathbf{B}, P_1, P_2) = (\mathbf{B}_1, \mathbf{B}_2, \sigma_1, \sigma_2, c_s) \\
\text{(struct)} \quad \frac{B_1 \equiv B'_1 \quad B'_1 \xrightarrow{\theta} B_2}{B_1 \xrightarrow{\theta} B_2} \quad \text{(redex)} \quad \frac{B_1 \xrightarrow{t; B; c; (n_1, n_2)} B_2}{B_1 \parallel B' \xrightarrow{t; B; c; (n_1 + n'_1, n_2 + n'_2)} B_2 \parallel B'} \\
\text{where } (n'_1, n'_2) = \text{Count}_t(B, B')
\end{array}$$

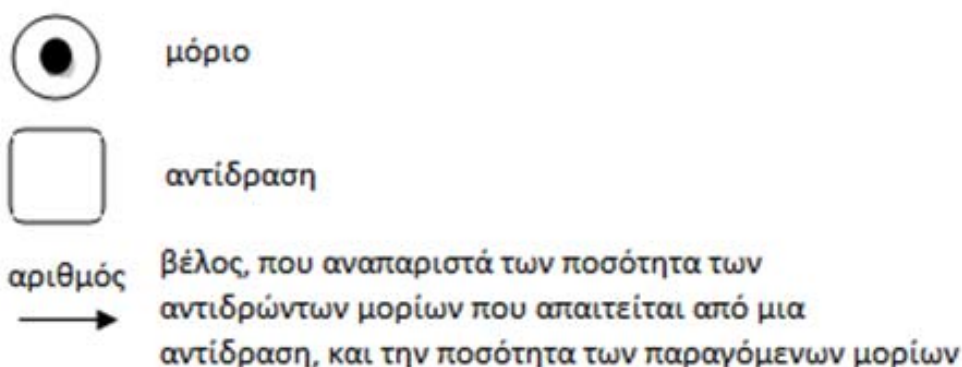
Εικόνα 2.2 Αξιώματα και κανόνες απλοποίησης της άλγεβρας διεργασιών Beta-Binders.

2.3 Μοντελοποίηση βιολογικών συστημάτων με τη χρήση Petri Nets.

Τα Petri Nets είναι μια τεχνική αναπαράστασης συστημάτων όπου διάφορες αντιδράσεις γίνονται ταυτόχρονα. Η τεχνική είναι τέτοια που τα συστήματα που κτίζονται είναι επεκτάσιμα, ενώ παρέχεται και μια γραφική αναπαράσταση των συστημάτων.

Η λογική πίσω από την προσομοίωση των αντιδράσεων είναι η εξής επαναλαμβανόμενη διαδικασία: επιλέγεται μια τυχαία αντίδραση που πληροί όλες τις προϋποθέσεις και εκτελείται. Αν δεν υπάρχουν τέτοιες αντιδράσεις τότε το σύστημα φτάνει σε αδιέξοδο.

Για την αναπαράσταση των βιολογικών συστημάτων χρησιμοποιούμε τα ακόλουθα σύμβολα:



Όπως φαίνεται και στο σχήμα ο άδειος κύκλος αναπαριστά ένα μόριο. Μόρια είναι τα οργανικά μόρια ενός βιολογικού συστήματος ή υποσυστήματα του(π.χ. πρωτεΐνες, ολόκληρο γονίδιο). Στη θέση του μαύρο κύκλου που φαίνεται μέσα στο μόριο, βάζουμε ένα αριθμό που αντιστοιχεί στην συσσώρευση του συγκεκριμένου μορίου. Η αντίδραση συμβολίζεται με ένα τετράγωνο στο ποίο καταλήγουν κάποια βέλη και από το οποίο εξέρχονται κάποια βέλη. Τα βέλη που ξεκινούν από κάποιο μόριο και καταλήγουν σε μια αντίδραση, δείχνουν ότι για να ενεργοποιηθεί η συγκεκριμένη αντίδραση χρειάζεται να υπάρχει το μόριο από το οποίο ξεκινά το βέλος και μάλιστα στην

ποσότητα την οποία βλέπουμε πάνω από το βέλος. Αντίστοιχα, τα βέλη που φεύγουν από τη αντίδραση και καταλήγουν σε κάποιο μόριο, δείχνουν την παραγωγή των συγκεκριμένων μορίων μετά από την συγκεκριμένη αντίδραση στην ποσότητα που βλέπουμε και πάλι πάνω από το βέλος. Μια αντίδραση καταναλώνει όλα τα μόρια που απαιτούνται και παράγει νέα.

2.4 Μοντελοποίηση βιολογικών συστημάτων με τη χρήση της άλγεβρας διεργασιών BioAmbients.

Για να κατανοήσουμε την άλγεβρα BioAmbients πρέπει να κατανοήσουμε την ιδέα πάνω στην οποία βασίζεται και τη μεθοδολογία που ακολουθείται για να μοντελοποιηθεί ένα βιολογικό σύστημα.

Η βασική της διαφορά με τις υπόλοιπες άλγεβρες διεργασιών είναι ότι εκμεταλλεύεται την ιεραρχία και την οργάνωση που έχει ένα βιολογικό σύστημα που είναι διαχωρισμένο σε διαμερίσματα. Οι περισσότερες τεχνικές για την μοντελοποίηση βιολογικών οργανισμών έχουν επικεντρωθεί στην μοντελοποίηση των διαφόρων χημικών αντιδράσεων που μπορούν να γίνουν σε ένα βιολογικό σύστημα, αγνοώντας την έννοια της οργάνωσης και του χώρου. Σε αυτή την προσέγγιση λαμβάνεται υπόψη τόσο η ιεραρχία κάθε μορίου όσο και η θέση του. Με τον όρο θέση δεν εννοούμε ακριβώς την τοποθεσία του μέσα στο βιολογικό σύστημα αλλά το διαμέρισμα στο οποίο ανήκει, καθώς αυτό περιορίζει και τις διάφορες αντιδράσεις στις οποίες μπορεί να λάβει μέρος.

Στον πραγματικό κόσμο η ιεραρχία δίνεται με την ύπαρξη διαμερισμάτων τα οποία είναι κλειστά μέσα σε μια μεμβράνη, ή διαμερισμάτων που αποτελούνται από μια συλλογή από μόρια τα οποία σχηματίζουν ένα μικρό υποσύστημα.

Για να μοντελοποιήσουμε ένα σύστημα ακολουθούμε τρία βασικά βήματα. Αρχικά οργανώνουμε τις πληροφορίες και την γνώση που έχουμε για το βιολογικό σύστημα στον πραγματικό κόσμο. Συνεχίζουμε επιλέγοντας το κατάλληλο μαθηματικό πεδίο για να εκφράσουμε τις διάφορες λειτουργίες του συστήματος. Τέλος συνδυάζουμε τα δύο για μοντελοποιήσουμε το σύστημα.

Ο σεβασμός στην ιεραρχία και τη δομή των βιολογικών συστημάτων οδήγησε στη δημιουργία της άλγεβρας αυτής, που αναπαριστά τα διαμερίσματα των βιολογικών οργανισμών ως ambients. Ένα ambient είναι ένα οριοθετημένος χώρος μέσα στον οποίο γίνονται οι διάφορες αντιδράσεις. Η οριοθέτηση υπάρχει για να ξεχωρίζει τι είναι μέσα και τι έξω από το ambient. Μέσα στο κάθε ambient υπάρχουν διεργασίες που το ελέγχουν, και άλλα φωλιασμένα ambients.

Για να ορίσουμε ένα ambient γράφουμε:

$$n[P_1|\dots|P_p|m_1[\dots]|\dots|m_q[\dots]]$$

όπου n είναι το όνομα του ambient P_i είναι μια διεργασία του ambient και m_i είναι ένα φωλιασμένο ambient μέσα στο ambient n .

Οι αντιδράσεις προσομοιώνονται με επικοινωνία διεργασιών πάνω σε κάποια κανάλια, μεταξύ συγκεκριμένων ambients ή ακόμα και μέσα στο ίδιο το ambient.

Συγκεκριμένα οι διεργασίες ακολουθούν την ακόλουθη σύνταξη:

Σύνταξη			
m, n, p	Names	$P, Q \equiv$	
$\pi \equiv$	Actions	$(new\ n)\ P$	Restriction
$\$n!\{m\}$	Output action	$P Q$	Composition
$\$n?\{m\}$	Input action	$!P$	Replication
$\$ \equiv$	Directions	$[P]$	Ambient(membrane)
local	Intra-ambient	$\pi. P$	Communication prefix
s2s	Inter-siblings	$M.P$	Capability prefix
p2c	Parent to child	$\sum_{i \in I} \pi_i. P_i$	Communication choice
c2p	Child to parent	$\sum_{i \in I} \pi_i. P_i$	Capability choice
$M, N \equiv$			
enter n	Synch entry		
accept n	Synch accept		
exit n	Synch exit		

expel n	Synch expel	
merge + n	Synch merge with	
merge - n	Synch merge into	

Ενώ οι αντιδράσει-επικοινωνίες ακολουθούν τους πιο κάτω κανόνες απλοποίησης:

Κανόνες απλοποίησης		
$[(T + \text{enter } n.P) \mid Q] \parallel [(T' + \text{accept } n.R) \mid S] \rightarrow [[P \mid Q] \parallel R \mid S]$	Red in	
$[(T + \text{exit } n.P) \mid Q] \parallel [(T' + \text{expel } n.R) \mid S] \rightarrow [P \mid Q] \parallel [R \mid S]$	Red out	
$[(T + \text{merge } + n.P) \mid Q] \parallel [(T' + \text{merge- } n.R) \mid S] \rightarrow [P \mid Q \mid R \mid S]$	Red merge	
$(T + \text{local } n! \{m\}.P) \mid \text{local } n? \{p\}.Q \rightarrow T' \rightarrow P \mid Q \{p \leftarrow m\}$	Red local	
$(T + p2c n! \{m\}.P) \parallel [(c2p n? \{p\}.Q + T') \parallel R] \rightarrow P \mid Q \{p \leftarrow m\} \parallel R$	Red parent output	
$[R \mid (T + c2p n! \{m\}.P)] \parallel (c2p n? \{p\}.Q + T') \rightarrow [R \mid P] \parallel Q \{p \leftarrow m\}$	Red parent input	
$[R \mid (T + s2s n! \{m\}.P)] \parallel [(s2s n? \{p\}.Q + T') \mid S] \rightarrow [R \mid P] \parallel [Q \{p \leftarrow m\} \mid S]$	Red sibling	
$P \rightarrow Q \Rightarrow (\text{new } n)P \rightarrow (\text{new } n)Q$	Red res	
$P \rightarrow Q \Rightarrow [P] \rightarrow [Q]$	Red amb	

$P \rightarrow Q \Rightarrow P R \rightarrow Q R$	Red par
$P \equiv P', P \rightarrow Q, Q \equiv Q' \Rightarrow P' \rightarrow Q'$	Red \equiv (struct)

Κεφάλαιο 3

Μοντελοποίηση παραδείγματος στις τρεις μεθόδους μοντελοποίησης συστημάτων που μελετήθηκαν.

3.1 Μοντελοποίηση παραδείγματος με τη χρήση άλγεβρας διεργασιών σε Beta Binders.

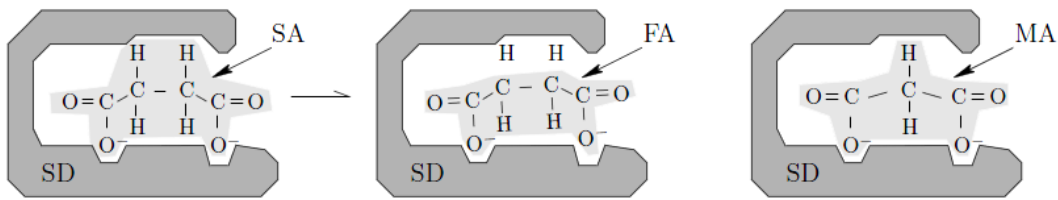
3.2 Μοντελοποίηση παραδείγματος με τη χρήση Petri Nets.

3.3 Μοντελοποίηση παραδείγματος με τη χρήση της άλγεβρας διεργασιών BioAmbients.

Για να καταλάβουμε καλύτερα τις άλγεβρες διεργασιών και τις τεχνικές αναπαράστασης βιολογικών οργανισμών που αναφέρονται πιο πάνω καλό είναι να δούμε την εφαρμογή τους σε ένα ρεαλιστικό σενάριο.

Θα μοντελοποιήσουμε λοιπόν την αναστολή του ενζύμου succinic hydrogenase, ενός ενζύμου που ρυθμίζει την οξειδωση του succinic acid. Η κανονική ρουτίνα ενός succinic dehydrogenase είναι να δεσμεύει ένα succinic acid, να αποδεσμεύει από αυτό δύο άτομα υδρογόνου σχηματίζοντας έτσι ένα fumaric acid, το οποίο και αποδεσμεύει. Το ένζυμο όμως αυτό μπορεί να δεσμεύσει και malonic acid, μιας και βιοχημική του δομή είναι η ίδια με αυτή του succinic acid. Μια τέτοια δέσμευση όμως αναστέλλει τη οξειδωση του succinic acid και κατ'επέκταση την παραγωγή fumaric acid, καθώς το succinic dehydrogenase έχει δεσμεύσει ένα malonic acid, και δεν μπορεί να δεσμεύσει

κάποιο succinic acid. Με αυτό τον τρόπο αναστέλλεται η λειτουργία του ενζύμου succinic dehydrogenase.



Εικόνα 3.1 Στο πρώτο και δεύτερο σχήμα βλέπουμε την γραφική αναπαράσταση της ομαλής λειτουργίας του ενζύμου succinic dehydrogenase, ενώ στο τρίτο σχήμα βλέπουμε την γραφική αναπαράσταση της αναστολής του ενζύμου succinic dehydrogenase.

Υπάρχει ανταγωνισμός μεταξύ του ποιός θα δεσμευτεί από το succinic dehydrogenase, οπότε ας δούμε τα δύο πιθανά σενάρια ένα προς ένα εφαρμοσμένα σε κάθε μια από τις τεχνικές. Πρώτο σενάριο θεωρούμε την οξείδωση του succinic acid, ενώ δεύτερο σενάριο θεωρούμε την δέσμευση malonic acid από το succinic dehydrogenase.

3.1 Μοντελοποίηση παραδείγματος με τη χρήση άλγεβρας διεργασιών σε Beta Binders.

Θεωρούμε τα succinic dehydrogenase(SD), succinic acid(SA) και malonic acid(MA) ως binders και γράφουμε:

$$\beta(x, D_{SD})[SD] \quad \beta(y, D_{SA})[SA] \quad \beta(y, D_{MA}) [MA], \text{ όπου}$$

$$D_{SD}=\{ s, f \} \quad D_{SA}=\{ s \} \quad D_{MA}=\{ f \}$$

Πρώτο σενάριο: Δέσμευση succinic acid από το ένζυμο succinic dehydrogenase.

Για την μοντελοποίηση αυτή χρειάζεται να ορίσουμε τις εξής διεργασίες:

$$SA = \overline{ox}(x).FA$$

$$SD = !ox(y) = ox(y).(\text{nil} \mid .!ox(y)) = ox(y).!ox(y)$$

οι οποίες χρειάζονται για την προσομοίωση της οξειδωσης και την παραγωγή του fumaric acid.

Η διαδικασία ολοκληρώνεται σε τρία βήματα:

α) δέσμευση του succinic acid από το succinic dehydrogenase.

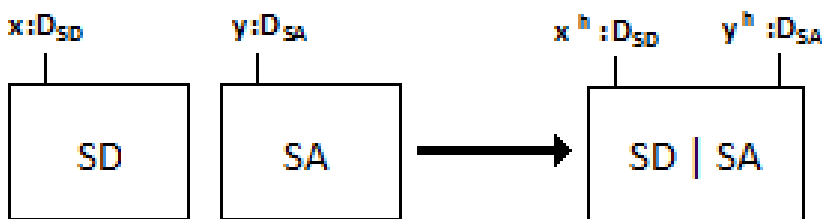
β) οξείδωση του succinic acid και παραγωγή του fumaric acid.

γ) διαχωρισμός του succinic dehydrogenase και του fumaric acid.

α) Δέσμευση του succinic acid από το succinic dehydrogenase

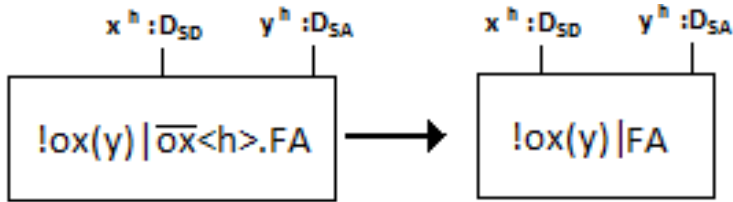
Η αντίδραση αυτή προσομοιώνεται με μια συνένωση των succinic acid και succinic dehydrogenase.

Για την συνένωση των δύο ορίζουμε μια συνάρτηση f_{join} , που δουλεύει όπως ορίζεται η join (βλέπε εικόνα), αλλά αναλαμβάνει να κρύψει τα κανάλια x και y καθιστώντας έτσι τα binders μη ενεργά για οποιαδήποτε άλλη αντίδραση.



$$\beta(x, D_{SD}) [SD] \parallel \beta(y, D_{SA}) [SA] \quad \rightarrow \quad \beta^h(x, D_{SD}) \beta^h(y, D_{SA}) [SD|SA]$$

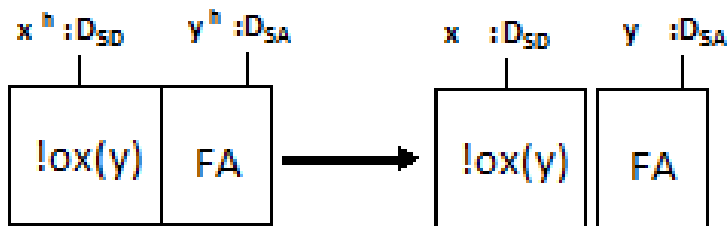
β) Οξείδωση του succinic acid και παραγωγή του fumaric acid.



$$\beta^h(x, D_{SD}) \beta^h(y, D_{SA}) [SD|SA] \rightarrow \beta^h(x, D_{SD}) \beta^h(y, D_{SA}) [SD|SA]$$

γ) Διαχωρισμός του succinic dehydrogenase και του fumaric acid.

Για το διαχωρισμό των δύο ορίζουμε μια συνάρτηση f_{split} , που δουλεύει όπως ορίζεται η split (βλέπε εικόνα), αλλά αναλαμβάνει να κάνει και πάλι τα binders ενεργά, ελευθερώνοντας τα κρυμμένα κανάλια.



$$\beta^h(x, D_{SD}) \beta^h(y, D_{SA}) [SD|SA] \rightarrow \beta(x, D_{SD})[SD] || \beta(y, D_{SA})[SA]$$

Δεύτερο σενάριο: Δέσμευση malonic acid από το ένζυμο succinic dehydrogenase

Η διαδικασία σε αυτό το σενάριο ολοκληρώνεται σε δύο βήμα:

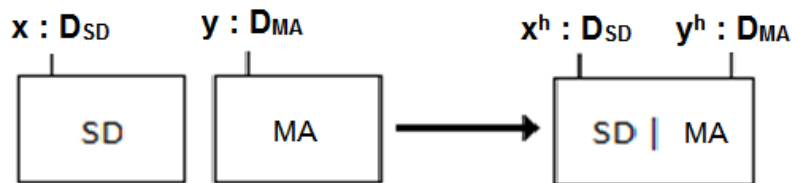
α) δέσμευση του malonic acid από το succinic dehydrogenase.

β) διαχωρισμός του succinic dehydrogenase και του malonic acid.

α) Δέσμευση του malonic acid από το succinic dehydrogenase.

Και εδώ όπως και πριν η δέσμευση προσομοιώνεται με μια συνένωση των malonic acid και succinic dehydrogenase.

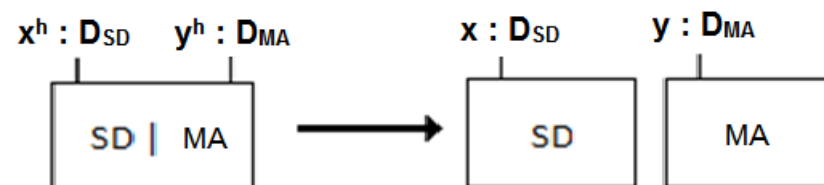
Για την συνένωση των δύο ορίζουμε μια συνάρτηση f_{join} , που δουλεύει όπως ορίζεται η join (βλέπε εικόνα), αλλά αναλαμβάνει να κρύψει τα κανάλια x και y καθιστώντας έτσι τα binders μη ενεργά για οποιαδήποτε άλλη αντίδραση.



$$\beta(x, D_{SD})[SD] \parallel \beta(y, D_{MA})[MA] \quad \rightarrow \quad \beta^h(x, D_{SD}) \beta^h(y, D_{MA}) [SD|MA]$$

β) Διαχωρισμός του succinic dehydrogenase και του malonic acid.

Για το διαχωρισμό των δύο ορίζουμε μια συνάρτηση f_{split} , που δουλεύει όπως ορίζεται η split (βλέπε εικόνα), αλλά αναλαμβάνει να κάνει και πάλι τα binders ενεργά, ελευθερώνοντας τα κρυμμένα κανάλια.



$$\beta^h(x, D_{SD}) \beta^h(y, D_{MA}) [SD|MA] \quad \rightarrow \quad \beta(x, D_{SD})[SD] \parallel \beta(y, D_{MA})[MA]$$

3.2 Μοντελοποίηση παραδείγματος με τη χρήση Petri Nets.

Πρώτο σενάριο: Δέσμευση succinic acid από το ένζυμο succinic dehydrogenase

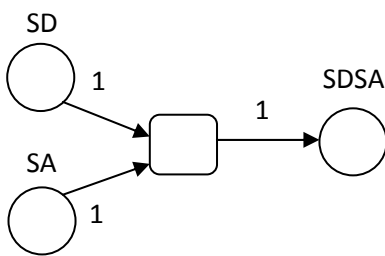
Και εδώ όπως και πριν διαδικασία ολοκληρώνεται σε τρία βήματα:

α) δέσμευση του succinic acid από το succinic dehydrogenase.

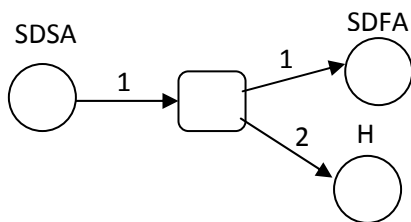
β) οξείδωση του succinic acid και παραγωγή του fumaric acid.

γ) διαχωρισμός του succinic dehydrogenase και του fumaric acid.

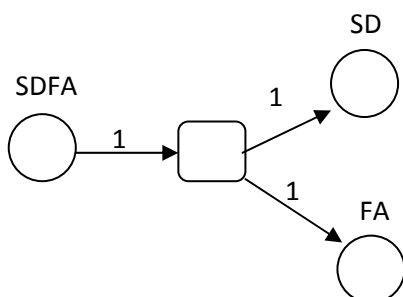
α) Δέσμευση του succinic acid από το succinic dehydrogenase



β) Οξείδωση του succinic acid και παραγωγή του fumaric acid.



γ) Διαχωρισμός του succinic dehydrogenase και του fumaric acid.

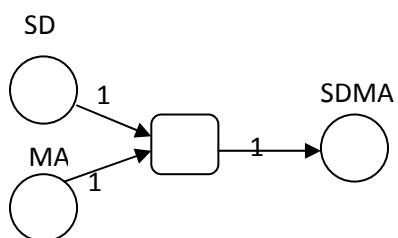


Δεύτερο σενάριο: Δέσμευση malonic acid από το ένζυμο succinic dehydrogenase

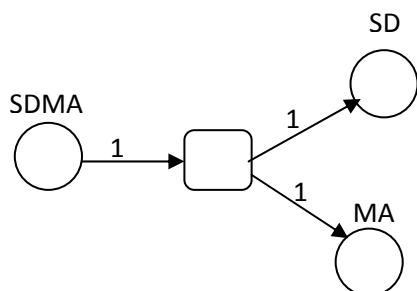
Και εδώ όπως και πριν η διαδικασία σε αυτό το σενάριο ολοκληρώνεται σε δύο βήμα:

- α) δέσμευση του malonic acid από το succinic dehydrogenase.
- β) διαχωρισμός του succinic dehydrogenase και του malonic acid.

α) Δέσμευση του malonic acid από το succinic dehydrogenase



β) Διαχωρισμός του succinic dehydrogenase και του malonic acid.



3.3 Μοντελοποίηση παραδείγματος με τη χρήση της άλγεβρας διεργασιών BioAmbients.

Για την μοντελοποίηση του βιολογικού αυτού συστήματος χρειάζεται να ορίσουμε τις εξής διεργασίες:

$$P_{SD1} = \text{accept sa.p2c sa !\{x1\}.expel fa. expel sa.P}_{SD}$$

$$P_{SD2} = \text{accept ma. expel ma. P}_{SD}$$

$$P_{SD} = (P_{SD1} + P_{SD2})$$

$$P_{MA} = \text{enter ma. exit ma.P}_{MA}$$

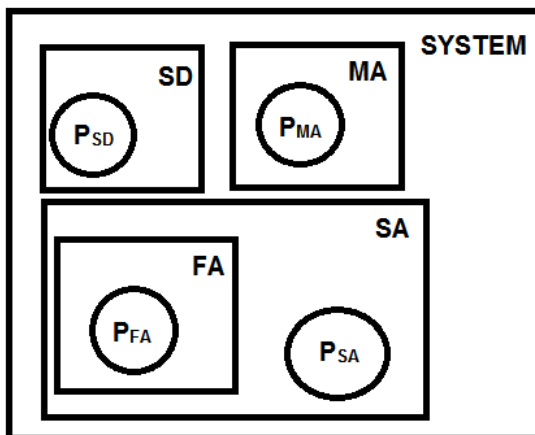
$$P_{SA} = \text{enter sa.c2p sa?\{x2\}. expel fa. exit sa.0}$$

$$P_{FA} = \text{exit fa. exit fa. P}_{ActiveFA}$$

$$P_{ActiveFA} = \text{η λειτουργία του fumalic acid.}$$

Αναπαριστούμε τα succinic dehydrogenase, succinic acid, malonic acid και fumalic acid ως ambients με ονόματα SD,SA,MA,FA αντίστοιχα. Το βιολογικό μας σύστημα ορίζεται ως εξής:

$$\text{SYSTEM}[\text{SD}[P_{SD}] \mid \text{MA}[P_{MA}] \mid \text{SA}[P_{SA} \mid \text{FA}[P_{FA}]]]$$



Πρώτο σενάριο: Δέσμευση succinic acid από το ένζυμο succinic dehydrogenase

Και εδώ όπως και πριν η διαδικασία της οξειδωσης ολοκληρώνεται σε τρία βήματα:

α) δέσμευση του succinic acid από το succinic dehydrogenase.

β) οξείδωση του succinic acid και παραγωγή του fumaric acid.

γ) διαχωρισμός του succinic dehydrogenase και του fumaric acid.

α) Δέσμευση του succinic acid από το succinic dehydrogenase.

Στο βήμα αυτό λαμβάνουν μέρος οι εξής διεργασίες:

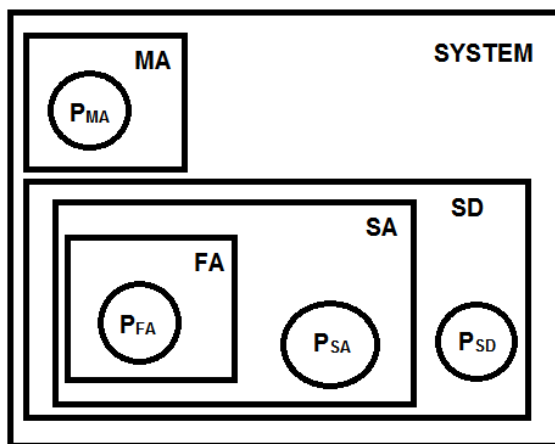
$$P_{SD1} = \text{accept sa.p2c sa !\{x1\}.expel fa. expel sa.P}_{SD}$$
$$P_{SD2} = \text{accept ma. expel ma. P}_{SD}$$
$$P_{SD} = (P_{SD1} + P_{SD2})$$
$$P_{SA} = \text{enter sa.c2p sa?\{x2\}. expel fa. exit sa.0}$$

και η διαδικασία προσομοιώνεται με μια (enter/accept) επικοινωνία των ambients SA και SD αντίστοιχα πάνω στο κανάλι sa. Συγκεκριμένα το ambient SA ζητά το SD, να του επιστρέψει την είσοδο, και το SD αποδέχεται την αίτηση. Αποδεχόμενος την αίτηση δεν μπορεί να γίνει ξανά δέσμευση άλλου SA ή MA μέχρι να μεσολαβήσουν τα βήματα της οξειδωσης και του διαχωρισμού.

Μετά την ολοκλήρωση αυτής της επικοινωνίας οι διεργασίες έχουν τη εξής λειτουργία:

$$P_{SD} = \text{p2c sa !\{x1\}.expel fa. expel sa.P}_{SD}$$
$$P_{SA} = \text{c2p sa?\{x2\}. expel fa. exit sa.0}$$

Και η δομή του συστήματος είναι η εξής:



β)Οξείδωση του succinic acid και παραγωγή του fumaric acid.

Η οξείδωση αυτή προσομοιώνεται με μια (p2s/c2p) επικοινωνία μεταξύ διεργασιών P_{SD} και P_{SA} των ambients SD και SA αντίστοιχα πάνω στο κανάλι sa. Το ambient FA μπορεί να κινηθεί έξω από το SA μόνο μετά από αυτή την επικοινωνία.

Στο βήμα αυτό λαμβάνουν μέρος οι εξής διεργασίες:

$$P_{SD} = p2c\ sa\ !\{x1\}.\expel\ fa.\ expel\ sa.P_{SD}$$

$$P_{SA} = c2p\ sa?\{x2\}.\expel\ fa.\ exit\ sa.0$$

Μετά την ολοκλήρωση αυτής της επικοινωνίας οι διεργασίες έχουν τη εξής λειτουργία:

$$P_{SD} = \expel\ fa.\ expel\ sa.P_{SD}$$

$$P_{SA} = \expel\ fa.\ exit\ sa.0$$

Η δομή του συστήματος δεν αλλάζει.

γ) Διαχωρισμός του succinic dehydrogenase και του fumaric acid.

Στο βήμα αυτό λαμβάνουν μέρος οι εξής διεργασίες:

$$P_{FA} = \text{exit fa. exit fa. } P_{ActiveFA}$$

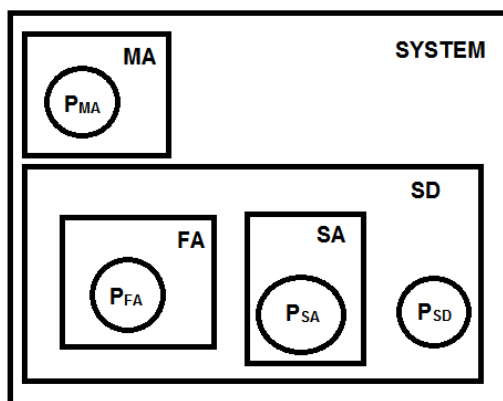
$$P_{SD} = \text{expel fa. expel sa. } P_{SD}$$

$$P_{SA} = \text{expel fa. exit sa. } 0$$

και η διαδικασία ολοκληρώνεται σε τρία υποβήματα.

Αρχικά έχουμε μια (exit/expel)επικοινωνία μεταξύ των ambients FA και SA πάνω στο κανάλι fa. Συγκεκριμένα το ambient FA ζητά από το ambient SA να του επιτρέψει την έξοδο και το ambient SA αποδέχεται την αίτηση.

Η επικοινωνία αυτή αλλάζει τη δομή του συστήματος ως εξής:



ενώ η λειτουργία των διεργασιών αλλάζει ως εξής:

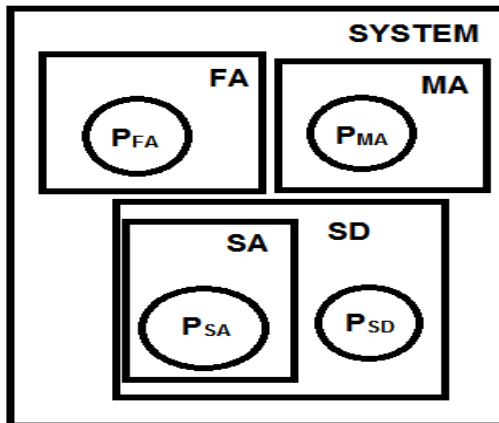
$$P_{SD} = \text{expel fa. expel sa. } P_{SD}$$

$$P_{FA} = \text{exit fa. } P_{ActiveFA}$$

$$P_{SA} = \text{exit sa. } 0$$

Συνεχίζουμε με μία (exit/expel)επικοινωνία μεταξύ των ambients FA και SD πάνω στο κανάλι fa. Συγκεκριμένα το ambient FA ζητά από το ambient SD να του επιτρέψει την έξοδο και το ambient SD αποδέχεται την αίτηση.

Η επικοινωνία αυτή αλλάζει τη δομή του συστήματος ως εξής:



ενώ η λειτουργία των διεργασιών αλλάζει ως εξής:

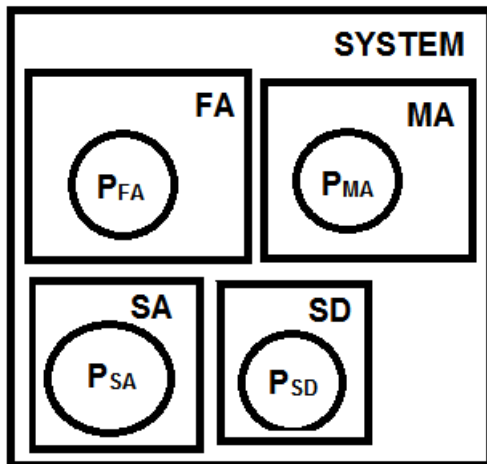
$$P_{SD} = \text{expel sa}.P_{SD}$$

$$P_{FA} = P_{\text{ActiveFA}}$$

$$P_{SA} = \text{exit sa}.0$$

Τέλος έχουμε μια (exit/expel)επικοινωνία μεταξύ των ambients SA και SD πάνω στο κανάλι sa. Συγκεκριμένα το ambient SA ζητά από το ambient SD να του επιτρέψει την έξοδο και το ambient SD αποδέχεται την αίτηση.

Η επικοινωνία αυτή αλλάζει τη δομή του συστήματος ως εξής:



ενώ η λειτουργία των διεργασιών έχει αλλάξει ως εξής:

$$P_{SD} = (P_{SD1} + P_{SD2})$$

$$P_{FA} = P_{ActiveFA}$$

$$P_{SA} = 0$$

Τώρα το ambient SD είναι διαθέσιμο να δεσμεύσει κάποιο άλλο SA και να το οξειδώσει ή κάποιο MA.

Δεύτερο σενάριο: Δέσμευση malonic acid από το ένζυμο succinic dehydrogenase

Και εδώ όπως και σε όλα τα σενάρια δέσμευσης του MA από το SD η διαδικασία ολοκληρώνεται σε δύο βήματα:

α) δέσμευση του malonic acid από το succinic dehydrogenase, που προσομοιώνεται με μια (enter/accept) επικοινωνία.

β) διαχωρισμός του succinic dehydrogenase και του malonic acid.

α) Δέσμευση του malonic acid από το succinic dehydrogenase, που προσομοιώνεται με μια (enter/accept) επικοινωνία.

Στο βήμα αυτό λαμβάνουν μέρος οι εξής διεργασίες:

$$P_{SD1} = \text{accept sa.p2c sa !}\{x1\}.\text{expel fa. expel sa.P}_{SD}$$

$$P_{SD2} = \text{accept ma. expel ma. P}_{SD}$$

$$P_{SD} = (P_{SD1} + P_{SD2})$$

$$P_{MA} = \text{enter ma. exit ma.P}_{MA}$$

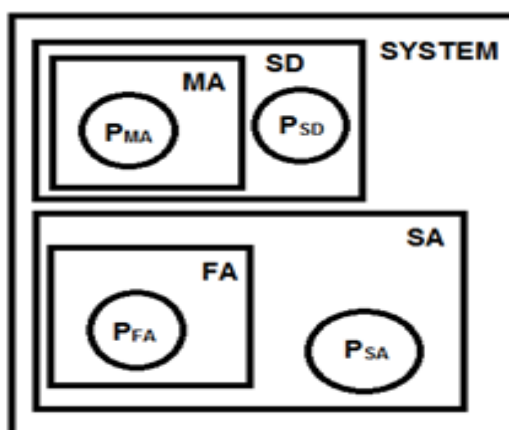
και προσομοιώνεται με μια (enter/accept) επικοινωνία των ambients MA και SD αντίστοιχα πάνω στο κανάλι ma. Συγκεκριμένα το ambient MA ζητά από το SD, να του επιστρέψει την είσοδο, και το SD αποδέχεται την αίτηση. Αποδεχόμενος την αίτηση δεν μπορεί να γίνει ξανά δέσμευση άλλου SA ή MA μέχρι να μεσολαβήσει το βήμα του διαχωρισμού.

Μετά την ολοκλήρωση αυτής της επικοινωνίας οι διεργασίες έχουν τη εξής λειτουργία:

$$P_{SD} = \text{expel ma. P}_{SD}$$

$$P_{MA} = \text{exit ma.P}_{MA}$$

Και η δομή του συστήματος είναι η εξής:



β) Διαχωρισμός του succinic dehydrogenase και του malonic acid.

Στο βήμα αυτό λαμβάνουν μέρος οι εξής διεργασίες:

$$P_{SD} = \text{expeI ma. } P_{SD}$$

$$P_{MA} = \text{exit ma. } P_{MA}$$

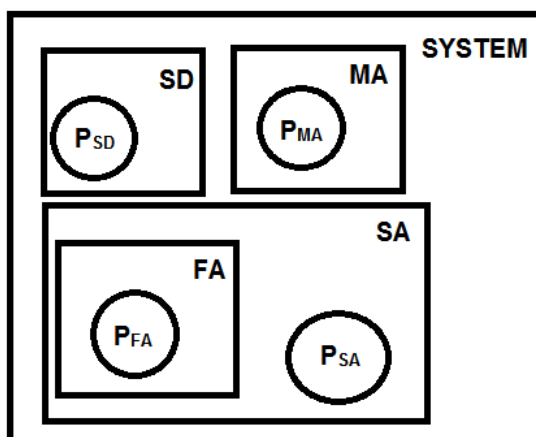
και η διαδικασία ολοκληρώνεται με μια (exit/expeI)επικοινωνία μεταξύ των ambients MA και SD πάνω στο κανάλι ma. Συγκεκριμένα το ambient MA ζητά από το ambient SD να του επιτρέψει την έξοδο και το ambient SD αποδέχεται την αίτηση.

Μετά την ολοκλήρωση αυτής της επικοινωνίας οι διεργασίες έχουν τη εξής λειτουργία:

$$P_{SD} = (P_{SD1} + P_{SD2})$$

$$P_{MA} = \text{enter ma. exit ma. } P_{MA}$$

Και η δομή του συστήματος είναι η εξής:



Τώρα το ambient SD είναι διαθέσιμο να δεσμεύσει κάποιο άλλο SA και να το οξειδώσει ή κάποιο MA.

Κεφάλαιο 4

Επιλογή της άλγεβρας διεργασιών που θα αναγνωρίζει το λογισμικό.

4.1 Κριτήρια επιλογής άλγεβρας διεργασιών.

4.2 Επιπλέον ανάλυση της επιλεγόμενης άλγεβρας διεργασιών.

4.1 Κριτήρια επιλογής άλγεβρας διεργασιών.

Αρχικά να τονίσουμε ότι και οι τρεις τεχνικές δεν αντιπροσωπεύουν ρεαλιστικά βιολογικά συστήματα, καθώς εκεί οι αντιδράσεις γίνονται παράλληλα ενώ με τις τεχνικές αυτές τις προσομοιώνουμε σειριακά.

Για την επιλογή της κατάλληλης άλγεβρας διεργασιών έλαβα σοβαρά υπόψη μου τις δυνατότητες κάθε μιας από τις άλγεβρες διεργασιών που μελέτησα, την προοπτική τους, και τον τρόπο που χειρίζονται τα δεδομένα ενός βιολογικού συστήματος.

Κατέληξα στο συμπέρασμα ότι οι άλγεβρες και οι τεχνικές που μελέτησα έχουν τις ίδιες δυνατότητες και αρκετά καλή προοπτική. Αυτό που επηρέασε την επιλογή μου είναι ο τρόπος που οργανώνονται και κατανέμονται οι πληροφορίες που παίρνουμε από ένα βιολογικό οργανισμό. Η καλύτερη δουλειά σε αυτό τον τομέα έχει γίνει με την άλγεβρα διεργασιών BioAmbients, μιας και λαμβάνει σοβαρά υπόψη τη δομή ενός βιολογικού οργανισμού. Αφού λοιπόν η δομή ενός βιολογικού οργανισμού παίζει τόσο σημαντικό ρόλο στην πραγματικότητα, πρέπει όλη αυτή την πληροφορία να την εκμεταλλευτούμε κατάλληλα.

Για να γίνει κατανοητή η σημαντικότητα της δομής ενός βιολογικού συστήματος θα αναφέρω ένα παράδειγμα. Έστω ότι έχουμε δύο μόρια A και B που ανήκουν σε ένα κύτταρο K, και επικοινωνούν συνεχώς μεταξύ τους,

ανταλλάζοντας πληροφορίες για την ομαλή λειτουργία του κυττάρου. Στο βιολογικό σύστημα που μοντελοποιούμε υπάρχουν πολλά τέτοια κύτταρα και κατ'επέκταση πολλά μόρια A και B. Σε ένα μη δομημένο μοντέλο κάποιο μόριο A μπορεί να επικοινωνήσει με κάποιο τυχαίο μόριο B, εκτός και αν γίνει ρητή αλλαγή του καναλιού, πάνω στο οποίο θα επικοινωνήσουν τα δύο μόρια, δημιουργώντας ένα κανάλι για κάθε κύτταρο K. Στο δομημένο μοντέλο που δημιουργείται με τη χρήση BioAmbients, ένα κύτταρο μπορεί να αναπαρασταθεί σαν ένα ambient K, και τα δύο μόρια ως δύο ambient A και B αντίστοιχα, παιδιά του ambient K. Η επικοινωνία αυτών των μορίων μπορεί να γίνει με μία sibling επικοινωνία σε κάποιο κανάλι, χωρίς να χρειάζεται να αλλάξουμε το κανάλι επικοινωνίας και να δώσουμε τόσα ονόματα όσα και τα κύτταρα K που υπάρχουν στο σύστημα. Η επικοινωνία αυτή είναι εσωτερική στο κύτταρο, και η ύπαρξη του ambient K ως πατέρας των δύο μορίων, περιορίζει την επικοινωνία τους με αντίστοιχα μόρια διαφορετικών κυττάρων.

Κάτι άλλο που με επηρέασε θετικά στην επιλογή των BioAmbients ως άλγεβρα διεργασιών ήταν ότι η δυνατότητα δόμησης που προσφέρει αυτή η άλγεβρα βοηθά κάποιον που δεν έχει και τόση σχέση με τις άλγεβρες διεργασιών, να δημιουργήσει στο μυαλό του την εικόνα του βιολογικού οργανισμού και αμέσως να ξεχωρίσει τα κύρια ambients που πρέπει να δημιουργηθούν στην μοντελοποίηση του συστήματος, καθώς και τις κύριες διεργασίες του κάθε ambient.

4.2 Επιπλέον ανάλυση της επιλεγόμενης άλγεβρας διεργασιών.

Χρειάζεται τώρα να αναλύσουμε εκτενέστερα τόσο την σύνταξη όσο και τους κανόνες απλοποίησης που εφαρμόζονται κατά την επικοινωνία δύο διεργασιών.

Όπως είδαμε και προηγουμένως η γραφή των διεργασιών ακολουθεί την εξής σύνταξη:

Σύνταξη		
m,n,p	Names	$P, Q \equiv$
$\pi \equiv$	Actions	Restriction $(\text{new } n) P$
$\$n!\{m\}$	Output action	Composition $P Q$
$\$n?\{m\}$	Input action	$!P$ Replication
$\$ \equiv$	Directions	$[P]$ Ambient(membrane)
local	Intra-ambient	$\pi.P$ Communication prefix P
s2s	Inter-siblings	Capability prefix $M.P$
p2c	Parent to child	$\sum_{i \in I} \pi_i.P_i$ Communication choice
c2p	Child to parent	$\sum_{i \in I} \pi_i.P_i$ Capability choice
M,N \equiv		
enter n	Synch entry	
accept n	Synch accept	
exit n	Synch exit	
expel n	Synch expel	
merge + n	Synch merge with	
merge – n	Synch merge into	

Ενώ οι αντιδράσεις-επικοινωνίες ακολουθούν τους πιο κάτω κανόνες απλοποίησης:

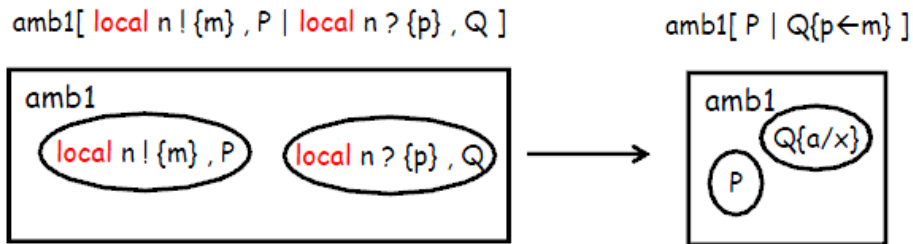
Κανόνες απλοποίησης	
$[(T + \text{enter } n.P) \mid Q] \parallel [(T' + \text{accept } n.R) \mid S] \rightarrow [[P \mid Q] \mid R \mid S]$	Red in
$[(T + \text{exit } n.P) \mid Q] \parallel [(T' + \text{expel } n.R) \mid S] \rightarrow [P \mid Q] \parallel [R \mid S]$	Red out
$[(T + \text{merge } + n.P) \mid Q] \parallel [(T' + \text{merge- } n.R) \mid S] \rightarrow [P \mid Q \mid R \mid S]$	Red merge
$(T + \text{local } n! \{m\}.P) \mid \text{local } n? \{p\}.Q + T' \rightarrow P \mid Q \{p \leftarrow m\}$	Red local
$(T + p2c n! \{m\}.P) \parallel [(c2p n? \{p\}.Q + T') \mid R] \rightarrow P \mid Q \{p \leftarrow m\} \mid R$	Red parent output
$[R \mid (T + c2p n! \{m\}.P)] \parallel (c2p n? \{p\}.Q + T') \rightarrow [R \mid P] \mid Q \{p \leftarrow m\}$	Red parent input
$[R \mid (T + s2s n! \{m\}.P)] \parallel [(s2s n? \{p\}.Q + T') \mid S] \rightarrow [R \mid P] \mid [Q \{p \leftarrow m\} \mid S]$	Red sibling
$P \rightarrow Q \Rightarrow (\text{new } n)P \rightarrow (\text{new } n)Q$	Red res
$P \rightarrow Q \Rightarrow [P] \rightarrow [Q]$	Red amb
$P \rightarrow Q \Rightarrow P \mid R \rightarrow Q \mid R$	Red par

Red \equiv (struct)

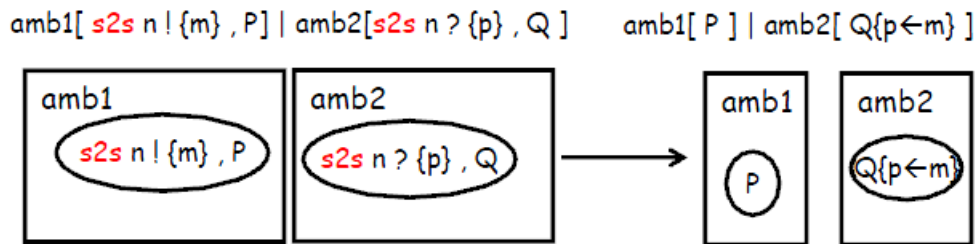
$P \equiv P', P \rightarrow Q, Q \equiv Q' \Rightarrow P' \rightarrow Q'$

Υπάρχουν τέσσερα είδη επικοινωνίας μεταξύ των διεργασιών των ambients.

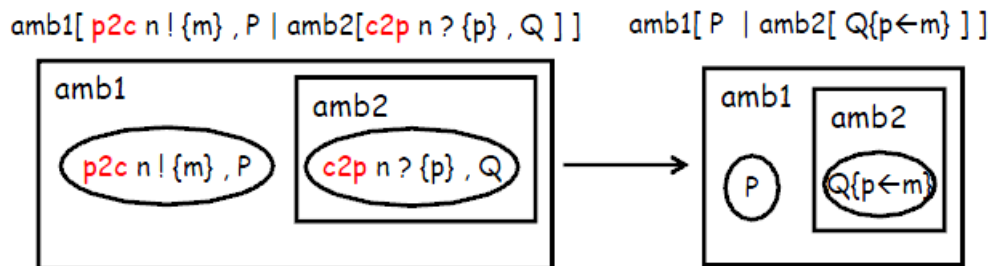
- local επικοινωνία, που μπορεί να γίνει μόνο μεταξύ δύο διεργασιών που βρίσκονται στο ίδιο ambient



- s2s επικοινωνία, που συμβολίζει μια επικοινωνία μεταξύ δύο διεργασιών, που βρίσκονται μέσα σε διαφορετικά ambients, και τα δύο αυτά ambient να είναι αδέρφια*.

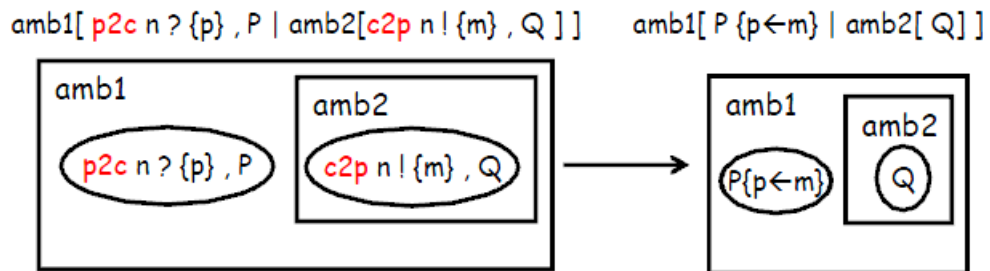


- p2c/c2p επικοινωνία, όπου μιας διεργασίας που βρίσκεται μέσα στον πατέρα επικοινωνεί με μια διεργασία που βρίσκεται σε ένα παιδί ambient.



*αδέρφια είναι δύο ambients που βρίσκονται στο ίδιο ύψος σε ιεραρχία, έχουν δηλαδή τον ίδιο πατέρα.

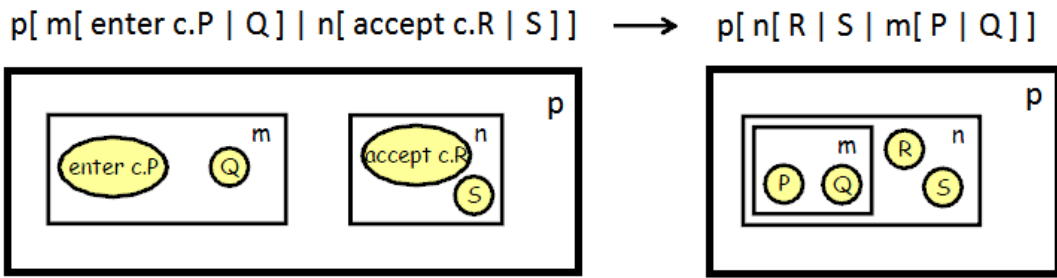
- c_2p/p_2c επικοινωνία, όπου μιας διεργασίας που βρίσκεται μέσα σε ένα παιδί επικοινωνεί με μια διεργασία που βρίσκεται μέσα στον πατέρα ambient.



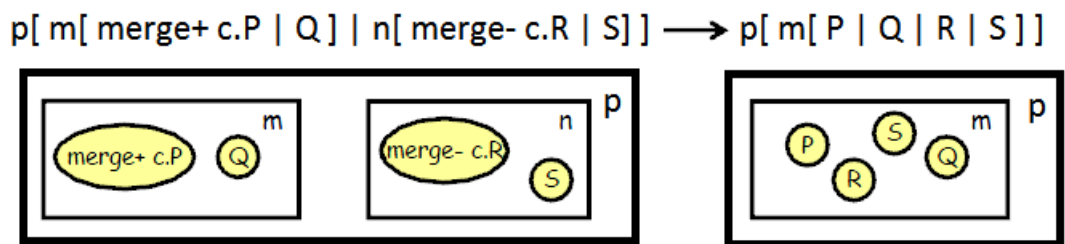
Υπάρχουν επίσης και κάποιες δυνατότητες αλλαγής της δομής ενός ambient, που επιτυγχάνεται με την επικοινωνία μεταξύ διεργασιών που βρίσκονται σε διαφορετικά ambients.

Συγκεκριμένα υπάρχουν τρεις τρόποι να αλλάξει η δομή ενός ambient. Ο δύο πρώτοι τρόποι που θα αναφέρω επιτυγχάνονται με επικοινωνία δύο διεργασιών, που βρίσκονται μέσα σε αδέρφια ambients.

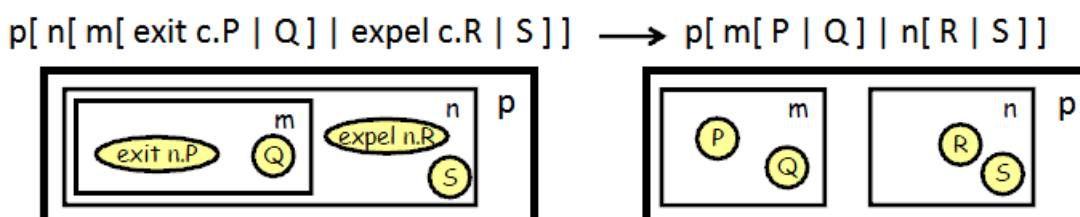
Ο πρώτος τρόπος είναι με μια (enter/accept) επικοινωνία, κατά την οποία ένα ambient m ζητά από το ambient n , να του επιτρέψει την είσοδο, και το n του επιτρέψει την είσοδο. Με αυτή την είσοδο τόσο η δομή του πατέρα του m όσο και η δομή του n αλλάζουν. Ο πατέρας του m , δεν έχει το m πλέον ως παιδί, ενώ το A έχει αποκτήσει ένα νέο παιδί, το m .



Ο δεύτερος τρόπος είναι με μια (merge+/merge-) επικοινωνία, κατά την οποία ένα ambient m ζητά από το ambient n να συνενωθούν και το ambient n, αποδέχεται την αίτηση. Με αυτή την συνένωση, δημιουργείται ένα νέο παιδί ambient που έχει το όνομα m και περιέχει τις διεργασίες του m και του n. Ο πατέρας των δύο ambients δεν έχει πλέον τα m και n ως παιδιά, αλλά ένα νέο παιδί, την συνένωση των δύο προηγούμενων παιδιών του, με όνομα m.



Ο τρίτος και τελευταίος τρόπος να αλλάξει η δομή ενός ambient είναι με μια (exit/expel) επικοινωνία, κατά την οποία ένα παιδί ambient m ζητά από τον πατέρα του n να βγει από αυτόν. Ο πατέρας απαντά θετικά στην αίτηση αυτή και έτσι το παιδί m βγαίνει από τον πατέρα του, με αποτέλεσμα η δομή του πατέρα να αλλάξει, ενώ ταυτόχρονα ο πατέρας του αποκτά ένα νέο παιδί, το m.



Κεφάλαιο 5

Ανάπτυξη λογισμικού.

5.1 Ανάλυση απαιτήσεων και προδιαγραφών του λογισμικού.

5.2 Επιλογή γλώσσας προγραμματισμού.

5.3 Χαρακτηριστικά λογισμικού και δυνατότητες χρήσης.

5.1 Ανάλυση απαιτήσεων και προδιαγραφών του λογισμικού.

Ακολουθούν κάποια χαρακτηριστικά που πρέπει να έχει το λογισμικό που θα αναπτυχθεί, υπό μορφή λίστας.

1. Το λογισμικό χρειάζεται να αναγνωρίζει την σύνταξη της άλγεβρας διεργασιών BioAmbients.
2. Το βιολογικό σύστημα θα δίνεται στο λογισμικό υπό μορφή κειμένου, μέσα σε ένα text file.
3. Το λογισμικό πρέπει να παρουσιάζει όλες τις πιθανές αντιδράσεις του βιολογικού συστήματος, και να δίνει τη δυνατότητα στον χρήστη να προσομοιώσει μια, δείχνοντας του τις αλλαγές που προκαλεί αυτή η αντίδραση στο σύστημα.
4. Το λογισμικό πρέπει να παρουσιάζει όλες τις πιθανές αντιδράσεις του βιολογικού συστήματος, και να δίνει τη δυνατότητα στον χρήστη να προσομοιώσει μια σειρά από τυχαίες αντιδράσεις πατώντας ένα κουμπί ,

δείχνοντας του τις αλλαγές που προκαλεί αυτή η αντίδραση στο σύστημα. Η διαδικασία πρέπει να τερματίζει πατώντας ένα κουμπί.

5. Το λογισμικό πρέπει να παρουσιάζει τη δυσδιάστατη αναπαράσταση του βιολογικού συστήματος, καθώς και μια αναπαράσταση της αντίδρασης που προσομοιώνεται.

6. Το λογισμικό πρέπει να παρουσιάζει όλες τις διεργασίες ενός ambient.

7. Το λογισμικό πρέπει να παρουσιάζει όλα τα ambient του συστήματος σε μια λίστα και να δίνει την δυνατότητα στον χρήστη να τα βλέπει πιο αναλυτικά.

8. Το λογισμικό πρέπει να εμφανίζει μηνύματα λάθους στη σύνταξη του κειμένου που δίνεται σαν είσοδο, και περιέχει το βιολογικό σύστημα γραμμένο σε BioAmbients.

9. Η σύνταξη που θα δέχεται εξής:

(όνομα σταθεράς):=(κείμενο), για τη δήλωση των σταθερών του συστήματος

(όνομα διεργασίας):(κείμενο), για τη δήλωση των διεργασιών

(όνομα ambient)['process₁'|...| 'process_p' | 'ambient₁' |...|'ambient_n'], για τα ambients

5.2 Επιλογή γλώσσας προγραμματισμού.

Για την επιλογή της γλώσσα προγραμματισμού με επηρέασε πολύ ο τρόπος που διατηρεί τη δομή ενός βιολογικού συστήματος η άλγεβρα διεργασιών BioAmbients. Κάθε ambient είναι ένας οριοθετημένος χώρος, μέσα στον οποίο περιορίζεται το εύρος των επικοινωνιών τόσο των παιδιών του όσο και των δικών του. Η δομή αυτή με οδήγησε στην απόφαση να χρησιμοποιήσω την έννοια της αντικειμενοστρέφιας, θεωρώντας ένα ambient ως ένα αντικείμενο, και ως ένα μικρόκοσμο, μέσα στον οποίο γίνονται οι επικοινωνίες. Τα βασικά χαρακτηριστικά κάθε αντικειμένου θα είναι τα παιδιά του και οι διεργασίες του. Τα παιδιά του περιορίζονται σε επικοινωνία με τον πατέρα, τα αδέρφια και τα παιδιά τους.

Κάτι άλλο πολύ σημαντικό είναι να υπάρχει ευκολία στη διαχείριση κειμένου, καθώς ο χρήστης θα δίνει σαν είσοδο το βιολογικό σύστημα υπό μορφή κειμένου.

Σημαντικό επίσης παράγοντα θεώρησα την ευκολία στην αποσφαλμάτωση, κάτι που με οδήγησε στο συμπέρασμα ότι πρέπει να χρησιμοποιήσω μια Interpreted γλώσσα προγραμματισμού.

Μια τέτοια γλώσσα είναι η java, με την οποία είχα και προηγούμενη εμπειρία, οπότε δεν θα αντιμετώπιζα δυσκολίες στην εκμάθηση της γλώσσας ούτε στην εκμάθηση της έννοιας της αντικειμενοστρέφιας.

Κάποια άλλα πλεονεκτήματα της java είναι ο μεγάλος αριθμός έτοιμων βιβλιοθηκών που θα είχα στη διάθεση μου, η ευκολία στη ανάπτυξη γραφικής διαπροσωπίας, η εύκολη διαχείριση της μνήμης και το γεγονός ότι είναι ανεξάρτητη πλατφόρμας.

Τέλος αποφάσισα να χρησιμοποιήσω το εργαλείο NetBeans IDE έκδοση 6.8 για την ανάπτυξη του λογισμικού, ένα εργαλείο που είχα χρησιμοποιήσει στο παρελθόν και γνώριζα τις ευκολίες που προσφέρει στη δημιουργία φορμών.

5.3 Χαρακτηρίστηκα λογισμικού και δυνατότητες χρήσης.

Ακολουθούν κάποια χαρακτηριστικά που έχει το λογισμικό που θα αναπτυχθεί, υπό μορφή λίστας, υποδεικνύοντας και την προδιαγραφή που ζητά το συγκεκριμένο χαρακτηριστικό αν υπάρχει.

- 1.(Προδιαγραφή 2).Με το πάτημα ενός κουμπιού θα εμφανίζεται ένα παράθυρο πλοήγησης στα αρχεία του υπολογιστή του χρήστη, μέσω του οποίου ο χρήστης θα επιλέγει το αρχείο κειμένου που περιέχει το βιολογικό σύστημα.
- 2.(Προδιαγραφή 3).Για τη παρουσίαση όλων των πιθανών αντιδράσεων χρειάζεται να έχουμε ένα πίνακα , μέσα στον οποίο θα τις παρουσιάζουμε. Ο χρήστης με double click πάνω σε μια από αυτές τις αντιδράσεις μέσα στον πίνακα θα μπορεί να την προσομοιώσει και να παρακολουθήσει τις αλλαγές που προκαλεί στο σύστημα.
- 3.(Προδιαγραφή 4).Με το πάτημα ενός κουμπιού ο χρήστης θα ενεργοποιεί μια διαδικασία τυχαίας επιλογής αντιδράσεως, από το σύνολο των αντιδράσεως του συστήματος. Η διαδικασία αυτή θα συνεχίζεται μέχρι ο χρήστης να πατήσει ένα κουμπί που τερματίζει την διαδικασία, ή μέχρι να μην υπάρχουν άλλες δυνατές αντιδράσεις.

- 4.(Προδιαγραφή 5).Για την παρουσίαση τόσο της δυσδιάστατης αναπαράστασης του βιολογικού συστήματος όσο και της αναπαράστασης των αντιδράσεων χρειάζεται να έχουμε κάποιο χώρο πάνω στη φόρμα, πάνω στον οποίο θα ζωγραφίζεται αυτή η αναπαράσταση.
- 5.(Προδιαγραφή 6). Για τη παρουσίαση όλων των διεργασιών ενός ambient χρειάζεται να έχουμε ένα πίνακα , μέσα στον οποίο θα τις παρουσιάζονται.
- 6.(Προδιαγραφή 7). Για τη παρουσίαση όλων των ambient χρειάζεται να έχουμε ένα πίνακα , μέσα στον οποίο θα τα παρουσιάζουμε. Ο χρήστης με double click πάνω σε ένα ambient μέσα στον πίνακα θα μπορεί να βλέπει το ambient πιο αναλυτικά.
- 7.(Προδιαγραφή 8.) Για την παρουσίαση των μηνυμάτων λάθους στην σύνταξη του κειμένου εισόδου, θα υπάρχει κάποιος χώρος όπου θα αναγράφονται τα μηνύματα.
- 8.(Προδιαγραφή 9.) Το λογισμικό θα αναγνωρίζει γραμμές που έχουν την ακόλουθη σύνταξη:
(όνομα σταθεράς):=(κείμενο)
(όνομα διεργασίας):(κείμενο)
(όνομα ambient)['process₁'|...| 'process_p' | 'ambient₁' |...|'ambient_n'].
Στην ίδια γραμμή μπορούν να μπου πάνω από μια δηλώσεις, φτάνει να διαχωρίζονται με μια κάθετη γραμμή '|'.
9. Πάνω στη φόρμα θα υπάρχει ένας πίνακας, μέσα στον οποίο θα εμφανίζονται οι σταθερές του βιολογικού συστήματος, όπως ορίστηκαν στο κείμενο εισόδου.
10. Ο χρήστης θα έχει τη δυνατότητα να αποθηκεύει σε ένα αρχείο κειμένου τις αντιδράσεις που έχουν προσομοιωθεί μέχρι στιγμής από το λογισμικό, με την σειρά που έχουν προσομοιωθεί. Συγκεκριμένα κάθε γραμμή στο αρχείο αυτό αντιστοιχεί σε ένα ζεύγος από ambients, και ένα αντίστοιχο ζεύγος διεργασιών που έλαβαν μέρος στην αντίδραση. Η κάθε γραμμή που αντιστοιχεί σε μια αντίδραση θα έχει την ακόλουθη δομή:
ambient₁[process₁] ambient₂[process₂]
11. Ο χρήστης θα έχει τη δυνατότητα πατώντας δεξί click σε κάποια από τις γραμμές του πίνακα που παρουσιάζει όλες τις δυνατές αντιδράσεις να

παρατηρήσει τις διεργασίες που λαμβάνουμε μέρος στη συγκεκριμένη αντίδραση, χρωματίζοντάς τις μαύρες.

12.Ο χρήστης θα έχει την δυνατότητα με το πάτημα ενός κουμπιού να επαναφέρει το σύστημα στην κατάσταση που ήταν πριν από την τελευταία αντίδραση.

Η φόρμα του λογισμικού θα έχει την εξής δομή κατά την εκκίνηση:

Simulation in BioAmbients

New Simulation

Possible reactions:

Reaction Type	First Ambient	Second Ambient

Ambients

Processes

Constants

ενώ μετά την εισαγωγή του αρχείου κειμένου θα έχει την ακόλουθη δομή:

Simulation in BioAmbients

New Simulation

Possible reactions:

Reaction Type	First Ambient	Second Ambient
Sibling	TRANSCR(S2S9ASAL(0),X8)	GENE(S2S6ASAL(X2),EXPELA...
Sibling	TRANSCR(S2S9ASAL(0),X8)	GENE(S2S6ASAL(X2),EXPELA...

Ambients

- GENEA
- GENEF
- TRANSCR
- TRANSL
- RNADeg
- PROTEINDeg
- RNA

Processes

Constants

```

X1:=S2S9ASAL(X2),EXPELA,X1+S2S9A(X1),EXPELA,X1
X2:=EXIT(S5UTR(X4),EXPELA,X2+S2S9EM(X3),0)
X3:=P2CBB1(0),(EXPELG,X3+X3)
X4:=S2S9DEEP(X6),P2CBB3(0),P2CBB3(0),0
X5:=S2S9DEEP(X7),P2CBB3(0),0
X6:=ACCEPTIF(((GAA+X8+X3G)))
X7:=S2S9B2(0),X4+C2PBB3(X5),0
    
```

Up

Undo

Save Simulation

Start Random Simulation

Stop Random Simulation

Κεφάλαιο 6

Μοντελοποίηση δημιουργίας και ενεργοποίησης πρωτεϊνών σε BioAmbients.

6.1 Περιγραφή της διαδικασίας στον πραγματικό κόσμο.

6.2 Περιγραφή της διαδικασίας στον κόσμο των BioAmbients.

6.3 Μοντελοποίηση του βιολογικού συστήματος σε BioAmbients.

Η επιλογή του case study έγινε με τρόπο ώστε αυτό που θα παρουσιαστεί να αντιπροσωπεύει αντιδράσεις που συναντούμε σε πραγματικούς οργανισμού, να είναι εύκολο στην κατανόηση και να δείχνει όσο το δυνατό περισσότερες αντιδράσεις τις οποίες μπορεί μέχρι σήμερα να προσομοιώσει η άλγεβρα διεργασιών BioAmbients. Το case study αυτό παρουσιάζει τη δημιουργία δύο πρωτεϊνών, της A και της TF, και στη συνέχεια την ενεργοποίηση της πρωτεΐνης TF από την A.

6.1 Περιγραφή της διαδικασίας στον πραγματικό κόσμο.

Στην πραγματικότητα η παραγωγή των δύο πρωτεϊνών και η ενεργοποίηση της TF από την A ακολουθεί την εξής ροή: Αρχικά το κομμάτι του DNA που μεταφράζεται στην πρωτεΐνη, μεταγράφεται σε mRNA, με τη βοήθεια της πολυμεράσης. Η πολυμεράση διαβάζει το DNA και δημιουργεί ένα πιστό αντίγραφο του DNA αντικαθιστώντας τη θυμίνη με ουρακίλη. Στη συνέχεια το mRNA μεταφέρεται στο ριβόσωμα, όπου και γίνεται η μετάφραση και κατ'επέκταση η δημιουργία της πρωτεΐνης. Στη συνέχεια η πρωτεΐνη A ενεργοποιεί την πρωτεΐνη TF. Η πρωτεΐνη A όπως θα δούμε και στη συνέχεια είναι μια πρωτεΐνη cyclin η οποία δεσμεύεται από την πρωτεΐνη κινάση. Η κινάση είναι ένα ένζυμο που αναλαμβάνει τη φωσφορυλίωση συγκεκριμένων πρωτεϊνών, με τη μεταφορά φωσφορικών ομάδων σε αυτές. Με την δράση της έχει τη δυνατότητα να ενεργοποιεί ή να απενεργοποιεί πρωτεΐνες, να αλλάζει τη λειτουργία τους, την τοποθεσία τους ή ακόμα και τη σχέση τους με τις άλλες πρωτεΐνες. Η κινάση φωσφορυλιώνει πρωτεΐνες που έχουν να κάνουν κάποια κρίσιμη δουλειά σε ένα οργανισμό, όπως η αντιγραφή του DNA. Για τη διαδικασία αυτή πρέπει να συνεργαστεί με μια πρωτεΐνη, την cyclin, η οποία την ενεργοποιεί και τις υποδεικνύει τις πρωτεΐνες που χρειάζονται φωσφορικές ομάδες.

6.2 Περιγραφή της διαδικασίας στον κόσμο των BioAmbients.

Η μοντελοποίηση αυτού του σεναρίου με τη χρήση BioAmbients είναι η ακόλουθη:

- Δύο γονίδια, το γονίδιο A και το γονίδιο TF αναπαριστούνται ως δύο ambients, Τα $Gene_A$ και $Gene_{TF}$ αντίστοιχα.
- Τα μεταγραφόμενα DNA των γονιδίων (mRna), παρουσιάζονται ως δύο ambients RNA_A και RNA_{TF} μέσα στα δύο γονίδια (ambients) $Gene_A$ και $Gene_{TF}$ αντίστοιχα.

- Η πρωτεΐνη στην οποία μεταφράζεται κάθε mRNA, παρουσιάζεται και πάλι ως ambient και συγκεκριμένα η πρωτεΐνη A αναπαριστάτε με ένα ambient με όνομα $Protein_A$ και περιέχεται στο $ambient RNA_A$, ενώ η πρωτεϊνική TF αναπαριστάτε με ένα ambient με όνομα $Protein_{TF}$ και περιέχεται στο $ambient RNA_{TF}$.
- Ένα ambient που συμβάλει στην μεταγραφή του DNA, με όνομα $Transcr$.
- Ένα ambient που συμβάλει στη μετάφραση του mRNA σε πρωτεΐνη, με όνομα $Transl$.
- Η μη ενεργή υπόσταση της πρωτεΐνης TF, αναπαριστάται ως ένα ambient με όνομα $Bount_{TF}$ μέσα στο $ambient Protein_{TF}$.
- Η ενεργή υπόσταση της πρωτεΐνης TF, αναπαριστάται ως ένα ambient με όνομα $Active_{TF}$ μέσα στο $ambient Bound_{TF}$.
- Ένα ambient που συμβάλουν στην καταστροφή του mRNA με όνομα RNA_{deg} .
- Ένα ambient που συμβάλουν στην καταστροφή των πρωτεϊνών με όνομα $Protein_{deg}$.
- Η κινάση που αναλαμβάνει να ενεργοποιήσει την πρωτεΐνη TF, αναπαριστάται ως ένα ambient με όνομα $Kinase$ και βρίσκεται μέσα στο $ambient Protein_A$.

Η διαδικασία δημιουργίας της πρωτεΐνης $Protein_A$ είναι η εξής: Αρχικά έχουμε την μεταγραφή του DNA σε mRNA του γονιδίου $Gene_A$. Η μεταγραφή αυτή έχει δύο υποβήματα. Προηγείται μια (sibling)επικοινωνία στο κανάλι basal μεταξύ των ambients $Gene_A$ και $Transcr$ και στη συνέχεια ακολουθεί μια (exit/expel)επικοινωνία στο κανάλι α, μεταξύ του $Gene_A$ και του RNA_A , που έχει σαν αποτέλεσμα την αποβολή του RNA_A από το $Gene_A$. Σε αυτό το σημείο το $ambient RNA_A$, που αντιπροσωπεύει το μεταγραφόμενο DNA του γονιδίου $Gene_A$, είναι σε θέση να αντιδράσει με το $ambient Transl$, το οποίο συμβάλει στην μετάφραση του mRNA σε πρωτεΐνη. Είναι σημαντικό εδώ να τονίσουμε ότι όπως και στην πραγματικότητα, χρειάζεται κάποιον να μεταγράψει το DNA για να πάρουμε το mRNA. Αυτή τη δουλειά της μεταγραφής την κάνει συμβολικά η (sibling) επικοινωνία μεταξύ των ambients $Gene_A$ και $Transcr$, που ξεκλειδώνει

στην ουσία την δυνατότητα του $Gene_A$ να αποβάλει το $ambient RNA_A$, και να ολοκληρωθεί έτσι η διαδικασία της μεταγραφής.

Στη συνέχεια έχουμε την μετάφραση του $mRNA(RNA_A)$ σε πρωτεΐνη ($Protein_A$). Και αυτή η διαδικασία ολοκληρώνεται σε δύο υποβήματα. Προηγείται μια (sibling)επικοινωνία στο κανάλι utr , μεταξύ των $ambients RNA_A$ και $Transl$ και ακολουθεί μια (exit/expel)επικοινωνία στο κανάλι b , μεταξύ των $ambients RNA_A$ και $Protein_A$, που έχει σαν αποτέλεσμα την αποβολή του $ambient Protein_A$ από το RNA_A , που αντιπροσωπεύει τη δημιουργία της πρωτεΐνης A μετά την μετάφραση του RNA_A . Σε αυτό το σημείο όπως και πριν βλέπουμε την συμβολή του $Transl$, για να καταστεί δυνατή η δημιουργία της πρωτεΐνης, αφού στην ουσία μόνο μετά από αυτή την επικοινωνία μπορεί το $ambient RNA_A$ να αποβάλει την $Protein_A$. Εδώ ολοκληρώνεται η διαδικασία της δημιουργία της πρωτεΐνης A .

Αντίστοιχη όμως είναι και η διαδικασία παραγωγής της πρωτεΐνης TF . Δηλαδή:

Αρχικά έχουμε την μεταγραφή του DNA σε $mRNA$ του γονιδίου $Gene_{TF}$. Η μεταγραφή αυτή έχει δύο υποβήματα. Προηγείται μια (sibling)επικοινωνία στο κανάλι $basal$ μεταξύ των $ambients Gene_{TF}$ και $Transcr$ και στη συνέχεια ακολουθεί μια (exit/expel)επικοινωνία στο κανάλι c , μεταξύ του $Gene_{TF}$ και του RNA_{TF} , που έχει σαν αποτέλεσμα την αποβολή του RNA_{TF} από το $Gene_{TF}$. Σε αυτό το σημείο το $ambients RNA_{TF}$, που αντιπροσωπεύει το μεταγραφόμενο DNA του γονιδίου $Gene_{TF}$, είναι σε θέση να αντιδράσει με το $ambient Transl$, το οποίο συμβάλει στην μετάφραση του $mRNA$ σε πρωτεΐνη. Στη συνέχεια έχουμε την μετάφραση του $mRNA(RNA_{TF})$ σε πρωτεΐνη ($Protein_T$). Και αυτή η διαδικασία ολοκληρώνεται σε δύο υποβήματα. Προηγείται μια (sibling)επικοινωνία στο κανάλι utr , μεταξύ των $ambients RNA_{TF}$ και $Transl$ και ακολουθεί μια (exit/expel)επικοινωνία στο κανάλι e , μεταξύ των $ambients RNA_{TF}$ και $Protein_{TF}$, που έχει σαν αποτέλεσμα την αποβολή του $ambient Protein_T$ από το RNA_{TF} , που αντιπροσωπεύει τη δημιουργία της πρωτεΐνης TF μετά την μετάφραση του RNA_{TF} .

Οι διαδικασίες παραγωγής των δύο πρωτεϊνών είναι ανεξάρτητες μεταξύ τους.

Μετά τη δημιουργία των δύο πρωτεϊνών A και TF. Θα δούμε την διαδικασία ενεργοποίησης της πρωτεΐνης TF από την πρωτεΐνη A.

Η διαδικασία της ενεργοποίησης της πρωτεΐνης TF ακολουθεί τα εξής βήματα: Αρχικά η πρωτεΐνη A δεσμεύει την πρωτεΐνη TF με μια (enter/accept)επικοινωνία στο κανάλι tf, μεταξύ των ambients $Protein_A$ και $Protein_{TF}$. Στη συνέχεια η πρωτεΐνη TF ($Protein_{TF}$) αποβάλλει το ambient $Bound_{TF}$ με μια (exit/expel)επικοινωνία στο κανάλι atf. Το ambient $Bound_{TF}$ αντιπροσωπεύει τη μη ενεργοποιημένη μορφή της πρωτεΐνης TF. Στη συνέχεια με μια (sibling)επικοινωνία στο κανάλι bb2 η κινάση(Kinase) αλλάζει την λειτουργία του $Bound_{TF}$. Ακολουθεί μια (parent to child)επικοινωνία στο κανάλι bb1 μεταξύ, του ambient $Protein_A$ και του $Bound_{TF}$. Τέλος με μια (exit/expel) επικοινωνία στο κανάλι f μεταξύ των ambients $Active_{TF}$ και $Bound_{TF}$ και μια (exit/expel) επικοινωνία στο κανάλι g μεταξύ των ambients $Active_{TF}$ και $Protein_A$, έχουμε στην επιφάνεια το ambient $Active_{TF}$ που αντιπροσωπεύει την ενεργή μορφή της πρωτεΐνης TF.

6.3 Μοντελοποίηση του βιολογικού συστήματος σε BioAmbients.

Η μοντελοποίηση του βιολογικού αυτού συστήματος είναι η ακόλουθη:

```
X1:=(s2s basal?{x2}.expel a.X1 + s2s pa?{x1}.expel a.X1)
X2:=exit a.(s2s utr?{x4}.expel b.X2 + s2s degm?{x3}.0)
X3a:=p2c bb1!{d}.(expel g.X3 + X3)
X3b:=s2s degp?{x6}.p2c bb3!{d}.p2c bb3!{d}.0
X3c:=s2s degp?{x7}.p2c bb3!{d}.0
X3:=accept tf.( X3a + X3b + X3c)
X4:=(s2s bb2!{d}.X4 + c2p bb3?{x5}.0)
X5:=(s2s basal?{y2}.expel c.X5+ s2s pa?{y1}.expel c.X5)
X6:=exit c.(s2s utr?{y4}.expel e.X6 + s2s degm?{y3}.0)
X7:=(s2s ptail!{d}.X7 + s2s degp?{y10}.0)
X8:=(s2s basal!{d}.X8+s2s ptail?{z1}.s2s pa!{d}.X8)
X9:=s2s utr!{d}.X9
X10:=s2s degm!{d}.X10
X11:=s2s degp!{d}.X10
X12a:= c2p bb1?{y9}.enter atf.0
X12b:=c2p bb3?{y8}.0
X12c:=s2s bb2?{y7}.(c2p bb1?{y6}.expel f.0+ c2p bb3?{y5}.0)
X12:=(X12a+X12b+X12c)
GeneA[GA1:X1| RnaA[RA1:X2|ProteinA[PA1:exit b.X3|Kinase[K1:X4]]]]
GeneTF[GTF1:X5|RnaTF[RTF1:X6|ProteinTF[PTF1:exit e. enter tf. expel atf.
accept atf.0|BoundTF[BTF1:exit atf.X12|ActiveTF[ATF1:exit f.exit g.X7]]]]]
Transcr[Tr1:X8]
Transl[TI1:X9]
RnaDEG[RDEG1:X10]
ProteinDEG[PDEG1:X11]
```

Κεφάλαιο 7

Χρήση και Προοπτικές του λογισμικού.

7.1 Χρήση του λογισμικού στο χώρο των βιολογικών επιστημών.

7.2 Μελλοντικές βελτιστοποιήσεις του λογισμικού.

7.1 Χρήση του λογισμικού στο χώρο των βιολογικών επιστημών.

Ο κλάδος των βιολογικών επιστημών καλείται να οργανώσει και να εκμεταλλευτεί ένα τεράστιο όγκο πληροφοριών που αντλεί μελετώντας τα βιολογικά συστήματα. Ο κλάδος της επιστήμης της πληροφορικής με τις τεράστιες υπολογίσιμες δυνατότητες έχει κληθεί να βοηθήσει στην οργάνωση αυτών των πληροφοριών. Το λογισμικό που έχει αναπτυχθεί στα πλαίσια αυτής της διπλωματικής εργασίας αναλαμβάνει να οργανώσει και να διαχειριστεί τις πληροφορίες που υπάρχουν σε ένα βιολογικό σύστημα μοντελοποιημένο σε BioAmbients. Επομένως θα μπορούσε να χρησιμοποιηθεί τόσο για την προσομοίωση βιολογικών συστημάτων σε ερευνητικά εργαστήρια βιολογίας όσο και για εκπαιδευτικούς σκοπούς.

Θα ήταν χρήσιμο σε ένα ερευνητικό εργαστήριο καθώς δίνει την δυνατότητα να παρακολουθεί κανείς την κατάσταση στην οποία βρίσκεται το βιολογικό σύστημα ανά πάσα στιγμή και να προσομοιώσει αντιδράσεις επιλέγοντάς τις ή αφήνοντας το λογισμικό να επιλέξει τυχαία.

Σε θέματα εκπαίδευσης θα μπορούσε να χρησιμοποιηθεί κατά την εκμάθηση μοντελοποίηση των βιολογικών συστημάτων στην άλγεβρα διεργασιών BioAmbients καθώς θα δίνει την δυνατότητα στο εκπαιδευόμενο να παρακολουθήσει και να μελετήσει το μοντέλο που έχει δημιουργήσει εύκολα με την χρήση του λογισμικού, και να εξετάσει την ορθότητα του μοντέλου που έχει δημιουργήσει.

7.2 Μελλοντικές βελτιστοποιήσεις του λογισμικού.

Τη δεδομένη στιγμή το λογισμικό προσομοιώνει τυχαίες αντιδράσεις σε ένα βιολογικό σύστημα, ή αντιδράσεις που επιλέγει ο χρήστης. Αυτές οι τυχαίες επιλογές αντιδράσεων γίνονται χωρίς βάρη μέσα από μια λίστα από πιθανές αντιδράσεις. Στο μέλλον είναι δυνατόν να προστεθούν βάρη που να επηρεάζουν την επιλογή των τυχαίων αντιδράσεων, φέρνοντας έτσι την προσομοίωση πιο κοντά στα πραγματικά δεδομένα, καθώς ξέρουμε ότι κάποιες αντιδράσεις είναι πιο πιθανόν να συμβούν από κάποιες άλλες.

Επίσης βάση της άλγεβρας που χρησιμοποιείται δεν είναι δυνατόν να γνωρίζουμε την θέση των παιδιών *ambients* που υπάρχουν σε ένα σύστημα, οπότε και η τοποθέτησή τους στη δυσδιάστατη αναπαράσταση του συστήματος γίνεται τυχαία. Στην άλγεβρα αυτή δεν παίζει ρόλο η συγκεκριμένη τοποθεσία των *ambients* αλλά ο χώρος στον οποίο βρίσκονται καθώς όπως αναφέρθηκε και προηγουμένως η ιεραρχία και η δόμηση παίζουν σημαντικό ρόλο στα όρια επικοινωνίας που μπορεί να έχει κάποιο *ambient*. Παρόλα αυτά αν στο μέλλον υπάρξει αλλαγή που καθορίζει την συγκεκριμένη θέση ενός *ambient* στο σύστημα θα μπορούσε εύκολα να προσαρμοστεί το λογισμικό ώστε το κάθε *ambient* να βρίσκεται στην τοποθεσία που καθορίζεται από την άλγεβρα.

Τέλος είναι σημαντικό να αναφέρουμε ότι η άλγεβρα που χρησιμοποιήθηκε έχει την δύναμη να μοντελοποιεί πολύπλοκα σύστημα με την χρήση ελάχιστων μορφών επικοινωνίας, οι οποίες χρησιμοποιούνται για την αναπαράσταση πολύπλοκων αντιδράσεων. Εντούτοις υπάρχουν αντιδράσεις που δεν μπορούν να προσομοιωθούν όπως για παράδειγμα η μίτωση και η μείωση, όπου χρειάζεται να αντιγραφούν τα δεδομένα που έχουμε για ένα κύτταρο, δημιουργώντας έτσι 2 κύτταρα. Τέτοιες αντιδράσεις θα προστεθούν στο μέλλον στην άλγεβρα, καθιστώντας την έτσι πιο πλήρη. Μια τέτοια αλλαγή είναι εύκολο να προστεθεί στο σύστημα αν χρειαστεί.

Κεφάλαιο 8

Συμπεράσματα.

8.1 Συμπεράσματα για τη χρήση της άλγεβρας διεργασιών Bioambients.

8.2 Συμπεράσματα για τη χρήση του λογισμικού για την προσομοίωση βιολογικών συστημάτων.

8.1 Συμπεράσματα για τη χρήση της άλγεβρας διεργασιών Bioambients.

Με τη χρήση της άλγεβρας διεργασιών BioAmbients εκτός των άλλων πληροφοριών που αντλούμε από ένα βιολογικό σύστημα, λαμβάνουμε υπόψη μας και την ιεραρχία και την δόμηση που συναντούμε σε ένα σύστημα, κάτι που οι υπόλοιπες άλγεβρες διεργασιών δεν εκμεταλλεύονται. Αυτά τα δεδομένα μας δίνουν την δυνατότητα να μοντελοποιήσουμε ένα βιολογικό σύστημα εύκολα, δημιουργώντας ένα μοντέλο που δεν απέχει και πολύ από το πραγματικό σύστημα σε ιεραρχία και δομή ενώ ταυτόχρονα μπορεί να προσομοιώσει όλες τις αντιδράσεις του συστήματος. Ένα άλλο πλεονέκτημά της είναι ότι λόγω του τρόπου που μοντελοποιεί τα συστήματα είναι εύκολο κανείς να καταλάβει και την σύνταξη της και τους κανόνες απλοποίησης που εφαρμόζονται.

Συνεπώς προβλέπεται η άλγεβρα αυτή να χρησιμοποιηθεί σε μεγάλο βαθμό από το χώρο των βιολογικών επιστημών για την μοντελοποίηση βιολογικών συστημάτων.

8.2 Συμπεράσματα για τη χρήση του λογισμικού για την προσομοίωση βιολογικών συστημάτων.

Αφού προβλέπεται η χρήση της άλγεβρας διεργασιών BioAmbients να είναι μεγάλη στο χώρο των βιολογικών επιστημών, μεγαλώνει και η χρησιμότητα του λογισμικού που έχει αναπτυχθεί, μιας και αναγνωρίζει αυτή την άλγεβρα διεργασιών και επιτρέπει στο χρήστη να προσομοιώσει βιολογικά συστήματα μοντελοποιημένα στην άλγεβρα διεργασιών Bioambients και να παρακολουθήσει την συμπεριφορά τους.

Βιβλιογραφία

- [1.] BAETEN, J. C. M. 2005. A brief history of process algebra. *Theor. Comput. Sci.* 335, 2-3 (May), 131–146.
- [2.] BONZANNI, N., FEENSTRA, K. A., FOKKINK, W., AND KREPSKA, E. 2009. What Can Formal Methods Bring to Systems Biology? *Lecture Notes in Computer Science* 5850, 16–+.
- [3.] BONZANNI, N., FEENSTRA, K. A., FOKKINK, W., AND KREPSKA, E. 2009. What can formal methods bring to systems biology? In *FM '09: Proceedings of the 2nd World Congress on Formal Methods*, Springer-Verlag, Berlin, Heidelberg, 16–22.
- [4.] CIOCCHETTA, F., AND HILLSTON, J. 2008. Process algebras in systems biology. In *Formal Methods for Computational Systems Biology*, Springer Berlin / Heidelberg, vol. 5016 of *Lecture Notes in Computer Science*, 265–312.
- [5.] DANOS, V., AND KRIVINE, J. 2007. Formal molecular biology done in ccs-r. *Electronic Notes in Theoretical Computer Science* 180, 3, 31 – 49. Proceedings of the First Workshop on Concurrent Models in Molecular Biology (BioConcur 2003).
- [6.] DEGANO, P., PRANDI, D., PRIAMI, C., AND QUAGLIA, P. 2006. Beta-binders for biological quantitative experiments. *Electronic Notes in Theoretical Computer Science* 164, 3, 101 – 117. Proceedings of the 4th International Workshop on Quantitative Aspects of Programming Languages (QAPL 2006), Quantitative Aspects of Programming Languages 2006.
- [7.] KREPSKA, E., BONZANNI, N., FEENSTRA, A., FOKKINK, W., KIELMANN, T., BAL, H., AND HERINGA, J. 2008. Design issues for qualitative modelling of biological cells with petri nets. In *FMSB '08: Proceedings of the 1st international workshop on Formal Methods in Systems Biology*, Springer-Verlag, Berlin, Heidelberg, 48–62.
- [8.] NIELSON, F., RIIS NIELSON, H., PRIAMI, C., AND ROSA, D. 2007. Control flow analysis for BioAmbients. *Electronic Notes in Theoretical Computer Science* 180, 3, 65–79.
- [9.] PRIAMI, C., AND QUAGLIA, P. 2005. Beta binders for biological interactions. In *Computational Methods in Systems Biology*, Springer Berlin / Heidelberg, vol. 3082 of *Lecture Notes in Computer Science*, 20–33.

- [10.] PRIAMI, C., REGEV, A., SHAPIRO, E., AND SILVERMAN, W. 2001. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Inf. Process.Lett.* 80, 1, 25–31.
- [11.] REGEV, A., PANINA, E. M., SILVERMAN, W., CARDELLI, L., AND SHAPIRO, E. 2004. BioAmbients: an abstraction for biological compartments. *Theor. Comput. Sci.* 325, 1, 141–167.
- [12.] CIOCCHETTA, F., AND HILLSTON, J. 2008. Process algebras in systems biology. In *Formal Methods for Computational Systems Biology*, Springer Berlin / Heidelberg, vol. 5016 of *LectureNotes in Computer Science*, 265–312.