

Λαμπρινή Καλαντζή

---

ΜΟΝΑΔΙΑΙΑ ΔΕΥΤΕΡΟΒΑΘΜΙΑ ΛΟΓΙΚΗ  
ΚΑΙ ΠΑΡΑΜΕΤΡΙΚΗ ΠΟΛΥΠΛΟΚΟΤΗΤΑ  
ΣΕ ΔΟΜΕΣ ΦΡΑΓΜΕΝΟΥ ΔΕΝΤΡΟΠΛΑΤΟΥΣ

Διδακτορική Διατριβή



---

Διαπανεπιστημιακό πρόγραμμα μεταπτυχιακών σπουδών στη  
Λογική και Θεωρία Αλγορίθμων και Υπολογισμού  
Τμήμα Μαθηματικών  
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών



# Ευχαριστίες

---

Θα ήθελα να ευχαριστήσω θερμά την επιβλέπουσα καθηγήτριά μου Ευγενία Φουστούκου, όπως επίσης και τους καθηγητές Σταύρο Κοσμάδακη, Bruno Courcelle και Irène Guessarian για την πολύτιμη βοήθεια που μου προσέφεραν ώστε να διεκπεραιωθεί αυτή η δουλειά.

Η διατριβή υποστηρίχθηκε οικονομικά από το πρόγραμμα «Μαθηματική Λογική, Θεωρία Αναδρομής και Εφαρμογές» που συγχρηματοδοτήθηκε από το Ευρωπαϊκό Κοινωνικό Ταμείο και Εθνικούς Πόρους – (ΕΠΕΑΕΚ ΙΙ) ΠΥΘΑΓΟΡΑΣ ΙΙ, και από το Πανεπιστημίου Πατρών στα πλαίσια του προγράμματος «Κ. Καραθεοδωρή – Βασική Έρευνα 2007-10: Λογική και Θεωρία Αλγορίθμων».



# Περιεχόμενα

---

<b>1</b>	<b>Εισαγωγή</b>	<b>1</b>
1.1	Το Πρόβλημα της Αποτίμησης για τη Μοναδιαία Δευτεροβάθμια Λογική . . .	1
1.2	Αναλυτική επισκόπηση της διατριβής . . . . .	9
<b>2</b>	<b>MSO αποτιμήσεις σε Δέντρα</b>	<b>17</b>
2.1	Βασικές Έννοιες . . . . .	18
2.2	Επαναπροσδιορισμός της σχέσης MSO–Αυτομάτων σε δέντρα . . . . .	21
2.2.1	Κωδικοποιήσεις αναθέσεων & αυτόματα αναθέσεων . . . . .	22
2.2.2	Η σχέση της MSO με τα Αυτόματα Αναθέσεων . . . . .	26
2.3	Επίλυση μέσω της Σχεσιακής Άλγεβρας . . . . .	32
2.3.1	Προκαταρκτικά της σχεσιακής άλγεβρας . . . . .	33
2.3.2	Εκφράζοντας το πρόβλημα μέσω ενός ακυκλικού συζευκτικού ερωτήματος $Q_t$ . . . . .	38
2.3.3	Αποτίμηση του $Q_t$ με τον αλγόριθμο του Yannakakis . . . . .	42
2.3.4	Αποτελέσματα πολυπλοκότητας & σύνδεση με τον FFG αλγόριθμο . .	47
2.4	Η λύση της datalog για το πρόβλημα αποτίμησης της MSO . . . . .	53
2.4.1	Προκατακτικά της datalog . . . . .	54
2.4.2	Εκφράζοντας το πρόβλημα στη datalog μέσω του $\text{Assign}_\phi$ . . . . .	57
2.4.3	Βελτιστοποιημένες εκδοχές του datalog προγράμματος $\Pi_{\phi,m}$ . . . . .	60
2.4.4	Επίλυση μέσω βελτιστοποιημένης $(k + 1)$ -datalog & μοναδιαίας datalog	66
2.4.5	Δύο ειδικές περιπτώσεις: το πρόβλημα ελέγχου μοντέλου & η αποτίμηση μονομελών ερωτημάτων . . . . .	71
2.4.6	Ορίζοντας datalog ερωτήματα μέσω του $\phi$ . . . . .	74
<b>3</b>	<b>MSO αποτιμήσεις σε Δομές Φραγμένου Δεντροπλάτους</b>	<b>85</b>
3.1	Προκαταρκτικά . . . . .	87
3.2	Η σχέση MSO–αυτομάτων στις δομές φραγμένου δεντροπλάτους . . . . .	89
3.2.1	Κωδικοποίηση αναθέσεων: διαίσθηση & τυπικοί ορισμοί . . . . .	89
3.2.2	Συνδυάζοντας δομές & δεντροδιασπάσεις . . . . .	93
3.2.3	Μία άμεση αναγωγή του προβλήματος αποτίμησης της MSO σε πρόβλημα αυτομάτων: τα αυτόματα αποδόμησης . . . . .	96

3.3	Η datalog λύση στα προβλήματα MSO αποτίμησης . . . . .	105
3.3.1	Από την MSO στην datalog: η κεντρική ιδέα της αναγωγής . . . . .	106
3.3.2	Η κατασκευή των δέντρων δομών . . . . .	107
3.3.3	Αποτίμηση μονομελών MSO-ορίσιμων ερωτημάτων μέσω της μοναδιαίας datalog . . . . .	109
3.3.4	Αποτίμηση $k$ -μελών MSO ερωτημάτων μέσω της $(k + 1)$ -datalog . . . . .	111
3.4	MSO-ορίσιμα προβλήματα βελτιστοποίησης & οι datalog λύσεις τους . . . . .	116
3.4.1	Το $k$ -card-MSO πρόβλημα ελέγχου μοντέλου . . . . .	116
3.4.2	Το min-card-MSO (min-card-MSO) πρόβλημα αποτίμησης . . . . .	118
3.4.3	Το min-MSO (max-MSO) πρόβλημα αποτίμησης . . . . .	122
3.5	Μία περίπτωση μελέτης: υπολογισμός καλύμματος κορυφών . . . . .	123
3.5.1	Η κατασκευή του αυτομάτου αποδόμησης που υπολογίζει καλύμματα κορυφών . . . . .	124
3.5.2	Επίλυση μέσω της datalog . . . . .	127
<b>4</b>	<b>Επίλογος</b>	<b>139</b>
	<b>Παραρτήματα</b>	<b>143</b>
	<b>Α΄ Επαγωγικοί ορισμοί αυτομάτων</b>	<b>143</b>
	<b>Β΄ Ο αλγόριθμος των Flum Frick Grohe</b>	<b>147</b>
	<b>Γ΄ Αυτόματα Αποδόμησης</b>	<b>149</b>
	<b>Ευρετήριο Συμβόλων</b>	<b>151</b>
	<b>Ευρετήριο Όρων</b>	<b>155</b>
	<b>Βιβλιογραφία</b>	<b>159</b>

# 1.

---

## Εισαγωγή

Το βασικό αντικείμενο της παρούσας διατριβής είναι η μελέτη παραμετρικών προβλημάτων αποτίμησης για τη μοναδιαία δευτεροβάθμια λογική σε πεπερασμένες δομές φραγμένου δεντροπλάτους. Σε αυτό το εισαγωγικό κεφάλαιο, ορίζονται αρχικά οι κεντρικές έννοιες που συνθέτουν τα προβλήματα που μελετάμε, και παρουσιάζεται μια αναλυτική ανασκόπηση των αποτελεσμάτων της σχετικής βιβλιογραφίας. Στη συνέχεια, παρατίθενται αναλυτικά τα βασικά αποτελέσματα της δουλειάς μας.

### 1.1 Το Πρόβλημα της Αποτίμησης για τη Μοναδιαία Δευτεροβάθμια Λογική

Παραθέτουμε μια επισκόπηση των βασικών αποτελεσμάτων που αφορούν στην αποτίμηση της μοναδιαίας δευτεροβάθμιας λογικής σε δομές φραγμένου δεντροπλάτους.

**Μοναδιαία Δευτεροβάθμια Λογική & Παραμετρική πολυπλοκότητα.** Το αντικείμενο της παρούσας διατριβής είναι η μοναδιαία δευτεροβάθμια λογική (Monadic Second-Order logic ή MSO) που αποτελεί επέκταση της πρωτοβάθμιας λογικής (First-Order logic ή FO) η οποία επιτρέπει την ποσοδείκτηση σε πεπερασμένα σύνολα. Πιο συγκεκριμένα, μελετάμε παραμετρικά προβλήματα αποτίμησης για τη μοναδιαία δευτεροβάθμια λογική σε πεπερασμένες δομές φραγμένου δεντροπλάτους. Η θεωρία παραμετρικής πολυπλοκότητας (parameterized complexity theory) [DF99, FG06, Nie06] είναι ένας σχετικά πρόσφατος κλάδος της υπολογιστικής πολυπλοκότητας ο οποίος παρέχει ένα πλαίσιο για μία ακριβέστερη ανάλυση υπολογιστικά δύσκολων (hard) αλγοριθμικών προβλημάτων, όπως είναι τα προβλήματα αποτίμησης για την FO και την MSO.

Το πρόβλημα ελέγχου μοντέλου  $MC(L, C)$  (model checking problem) για μία λογική  $L$  και μία κλάση  $C$  δομών ορίζεται ως το ακόλουθο πρόβλημα απόφασης: για μία δεδομένη πρόταση  $\phi$  της

$L$  και μία δεδομένη δομή  $\mathcal{A}$  της κλάσης  $\mathcal{C}$ , αληθεύει ότι  $\mathcal{A} \models \phi$ ; Το πρόβλημα της αποτίμησης  $\text{Eval}(L, \mathcal{C})$  (evaluation problem) για μία λογική  $L$  και μία κλάση  $\mathcal{C}$  δομών συνίσταται στον υπολογισμό του συνόλου  $\phi(\mathcal{A})$  των αναθέσεων που ικανοποιούν έναν δεδομένο τύπο  $\phi$  της  $L$  σε μία δεδομένη δομή  $\mathcal{A} \in \mathcal{C}$ . Είναι γνωστό αποτέλεσμα ότι η συνολική πολυπλοκότητα (combined complexity) του προβλήματος ελέγχου μοντέλου τόσο για την FO όσο και για την MSO είναι PSPACE-πλήρης (PSPACE-complete) [Sto74, Var82]. Ο συνήθης τρόπος να διαχειριστεί κανείς τέτοιου είδους αποτελέσματα «κακής» πολυπλοκότητας είναι να περιορίσει τα στιγμιότυπα εισόδου, δηλαδή στη συγκεκριμένη περίπτωση να θεωρήσει υποκλάσεις είτε των δεδομένων δομών είτε των δεδομένων τύπων λογικής. Δυστυχώς, ο περιορισμός των δομών που θεωρούμε ως είσοδο δεν συντελεί στη βελτίωση της πολυπλοκότητας, καθώς αυτή εξακολουθεί να είναι PSPACE-πλήρης ακόμα και όταν το πρόβλημα ελέγχου μοντέλου, τόσο για την FO όσο και για την MSO, σχετίζεται με μια συγκεκριμένη δομή (expression complexity). Από την άλλη, θεωρώντας το πρόβλημα ελέγχου μοντέλου για μια συγκεκριμένη λογική πρόταση (data complexity) έχουμε βελτίωση της πολυπλοκότητας μόνο για την περίπτωση της FO όπου η πολυπλοκότητα αποδεικνύεται τώρα πολυωνυμικού χρόνου (PTIME). Αξίζει να σημειωθεί ότι οι παραπάνω προσεγγίσεις δε λαμβάνουν υπόψη το γεγονός ότι στις περισσότερες περιπτώσεις το μέγεθος της πρότασης που θεωρούμε ως είσοδο είναι σημαντικά μικρότερο από το μέγεθος της δομής που αποτελεί μέρος της εισόδου.

Μία ακριβέστερη ανάλυση πολυπλοκότητας τέτοιων προβλημάτων απόφασης, η οποία επιπρόσθετα λαμβάνει υπόψη τις ανάλογες παραμέτρους των δεδομένων εισόδου, μπορεί να επιτευχθεί όταν αυτά μελετηθούν από την σκοπιά της *παραμετρικής θεωρίας πολυπλοκότητας* (parameterized complexity theory). Η κεντρική ιδέα αυτής της προσέγγισης είναι η απομόνωση της εξάρτησης που έχει ο χρόνος εκτέλεσης (running time) του αλγορίθμου από ένα συγκεκριμένο μέρος της εισόδου, το οποίο ονομάζουμε παράμετρο. Πιο συγκεκριμένα, αποτελεί βασικό ενδιαφέρον της παραμετρικής πολυπλοκότητας η διάκριση μεταξύ χρόνων εκτέλεσης της μορφής  $O(2^p \cdot n)$ , και της μορφής  $O(n^p)$ . Όπως θα δούμε στη συνέχεια, αν υποθέσουμε ότι το  $p$  είναι η παράμετρος, τότε ο πρώτος χρόνος αντιστοιχεί στην πολυωνυμική πολυπλοκότητα της κλασικής θεωρίας πολυπλοκότητας ενώ ο δεύτερος όχι.

Ένα παραμετρικό πρόβλημα ορίζεται ως ένα ζευγάρι  $(K, p)$  όπου το  $K$  είναι ένα πρόβλημα απόφασης και το  $p$  είναι μία υπολογίσιμη σε πολυωνυμικό χρόνο συνάρτηση η οποία αντιστοιχεί έναν φυσικό αριθμό, την παράμετρο, σε κάθε στιγμιότυπο. Ένα τέτοιο πρόβλημα ονομάζεται FPT (fixed-parameter tractable) εάν υπάρχει ένας αλγόριθμος που επιλύει το πρόβλημα, για ένα στιγμιότυπο εισόδου  $x$ , σε χρόνο

$$f(p(x)) \cdot |x|^c$$

όπου η  $f$  είναι κάποια υπολογίσιμη συνάρτηση και η  $c$  είναι κάποια σταθερά,  $|x|$  είναι το μέγεθος του στιγμιότυπου εισόδου  $x$ . Η συνάρτηση  $f$  ονομάζεται η *παραμετρική εξάρτηση* (parametric dependence) του αλγορίθμου. Ένας αλγόριθμος ονομάζεται FPT ως προς την παράμετρο  $p$  εάν επιτυγχάνει ένα τέτοιο χρόνο εκτέλεσης για κάθε είσοδο. Ένας FPT αλγόριθμος με  $c = 1$  ονομάζεται γραμμικός σταθερής παραμέτρου (fixed parameter linear). Εκτός αν αναφέρεται διαφορετικά, θεωρούμε ότι τόσο το πρόβλημα ελέγχου μοντέλου όσο και το πρόβλημα



αποτίμησης παραμετροποιούνται ως προς το μήκος  $|\phi|$  του δεδομένου τύπου.

Το *δεντροπλάτος* (*treewidth*) είναι μία δομική έννοια για την ανάκτηση FPT αποτελεσμάτων που σχετίζονται με προβλήματα ελέγχου μοντέλου, καθώς επιτρέπει τη διάκριση ανάμεσα σε «διαχειρίσιμα» και σε «μη-διαχειρίσιμα» στιγμιότυπα (βλέπε τα [Flu01, Gro07, Kre09]). Ειδικότερα, το δεντροπλάτος είναι μια έννοια που εισήχθη από τους Robertson και Seymour [RS86] στο πλαίσιο της θεωρίας γραφημάτων και αποτελεί ένα μέτρο για την ομοιότητα μιας δεδομένης δομής με ένα δέντρο. Έτσι, τα δέντρα και τα δάση έχουν δεντροπλάτος 1, οι κύκλοι έχουν δεντροπλάτος 2, μία κλίκα  $k$  κόμβων έχει δεντροπλάτος  $k - 1$  (βλέπε επίσης τα [Bod93, Bod07]).

**Αποτίμηση σε δέντρα.** Ένα πρώτο FPT αποτέλεσμα που αφορά στην MSO πάνω σε δέντρα είναι το ακόλουθο.

**Θεώρημα 1.1.1 ([TW68]).** Το παραμετρικό πρόβλημα ελέγχου μοντέλου για την MSO πάνω σε δέντρα είναι επιλύσιμο από ένα *fixed-parameter linear* αλγόριθμο.

Το βασικό κομμάτι της απόδειξης του παραπάνω θεωρήματος αποδεικνύει ότι οι κλάσεις δέντρων που είναι ορίσιμες (definable) στην MSO είναι ακριβώς εκείνες που είναι αναγνωρίσιμες (recognizable) από ένα ντετερμινιστικό αυτόματο για δέντρα<sup>1</sup>. Πιο συγκεκριμένα, από το *θεώρημα των Thatcher & Wright*, έχουμε ότι για κάθε MSO πρόταση  $\phi$  μπορούμε να κατασκευάσουμε αποτελεσματικά ένα ντετερμινιστικό bottom-up αυτόματο δέντρου  $A_\phi$ , τέτοιο ώστε  $\mathcal{T} \models \phi$  αν και μόνο αν το  $A_\phi$  αποδέχεται το δέντρο  $\mathcal{T}$ . Συνεπώς, μέσω του υπολογισμού της εκτέλεσης (run) του  $A_\phi$  στο  $\mathcal{T}$ , μπορεί να αποφασιστεί σε γραμμικό χρόνο εάν ισχύει ότι  $\mathcal{T} \models \phi$ . Αναλυτικότερα, ο χρόνος εκτέλεσης αυτού του FPT αλγορίθμου αυτομάτων είναι

$$f(k) \cdot |T|$$

όπου  $k$  είναι το μήκος της  $\phi$  και  $|T|$  είναι το μέγεθος του συνόλου κόμβων  $T$  του δέντρου  $\mathcal{T}$ . Η παραμετρική εξάρτηση  $f$  αυτού του αλγορίθμου είναι μία μη-στοιχειώδης (non-elementary) συνάρτηση, γεγονός που οφείλεται στο ότι ο αριθμός των καταστάσεων του minimal αυτομάτου  $A_\phi$  δε φράσσεται από καμία στοιχειώδη συνάρτηση του μήκους  $k$  της πρότασης  $\phi$ .<sup>2</sup>

Μια εναλλακτική απόδειξη, η οποία βασίζεται στη *μέθοδο σύνθεσης των Feferman-Vaught* (*Feferman-Vaught composition method*) [Mak04, Tho97a], είναι η εξής: έστω ότι το  $\mathcal{T}$  είναι ένα δυαδικό δέντρο με σύνολο κόμβων  $T$  και έστω ότι η  $\phi$  είναι μία πρόταση της MSO με βαθμό ποσοδείκτη (quantifier rank)  $q$ . Ξεκινώντας από τα φύλλα και συνεχίζοντας μέχρι και τη

<sup>1</sup>Θυμίζουμε ότι μία κλάση  $\mathcal{C}$  δομών είναι ορίσιμη σε μία λογική  $L$  εάν υπάρχει μία πρόταση  $\phi$  της  $L$  τέτοια ώστε  $\mathcal{A} \in \mathcal{C}$  ανν  $\mathcal{A} \models \phi$ . Η  $\mathcal{C}$  είναι αναγνωρίσιμη από ένα αυτόματο ανν υπάρχει ένα αυτόματο που αποδέχεται (accepts) ακριβώς εκείνες τις δομές που ανήκουν στην  $\mathcal{C}$ .

<sup>2</sup>Μία συνάρτηση  $f : \mathbb{N} \mapsto \mathbb{N}$  είναι στοιχειώδης (elementary) εάν υπάρχει ένα  $h \geq 0$  τέτοιο ώστε  $f(n) \geq \exp^{(h)}(n)$  για κάθε  $n \in \mathbb{N}$ , όπου η συνάρτηση  $\exp^{(h)}$  ορίζεται ως εξής:  $\exp^{(0)}(n) = n$  και  $\exp^{(h)}(n) = 2^{\exp^{(h-1)}(n)}$  για κάθε  $n, h \in \mathbb{N}$ .

ρίζα, ο αλγόριθμος υπολογίζει σε κάθε κόμβο  $n \in T$  τον  $\text{MSO}_q\text{-type}^3 \text{tr}_q^{\text{MSO}}(\mathcal{T}, n)$  του  $n$  στο υποδέντρο  $\mathcal{T}_n$  του  $\mathcal{T}$  που έχει ρίζα τον  $n$ . Στη ρίζα  $r$ , μπορούμε μέσω του  $\text{tr}_q^{\text{MSO}}(\mathcal{T}, r)$  να αποφανθούμε αν  $\mathcal{T} \models \phi$ . Όπως και στην περίπτωση του παραπάνω αλγορίθμου αυτομάτων έτσι και εδώ, ο FPT αλγόριθμος που βασίζεται στα *type* έχει μη στοιχειώδη παραμετρική εξάρτηση καθώς ο αριθμός των  $\text{MSO}_q\text{-type}$  δε φράσσεται από καμία στοιχειώδη συνάρτηση του  $q$ .

Η μη-στοιχειώδης παραμετρική εξάρτηση των δύο παραπάνω FPT αλγορίθμων είναι γενικά αναπόφευκτη. Πιο συγκεκριμένα, κάτω από λογικές προϋποθέσεις, το αποτέλεσμα αυτό ισχύει για κάθε FPT αλγόριθμο που επιλύει το ίδιο πρόβλημα, όπως διατυπώνεται στο θεώρημα που ακολουθεί.

**Θεώρημα 1.1.2 ([FG04]).** *Εκτός αν  $\text{P} = \text{NP}$ , δεν υπάρχει FPT αλγόριθμος με στοιχειώδη παραμετρική εξάρτηση ο οποίος να επιλύει το παραμετρικό πρόβλημα ελέγχου μοντέλου (parameterized model checking problem) για την MSO σε δομές που είναι δέντρα.*

**Αποτίμηση σε δομές φραγμένου δεντροπλάτους.** Το FPT αποτέλεσμα του Θεωρήματος 1.1.1 γενικεύεται από τα δέντρα σε δομές που μοιάζουν με δέντρα μέσω της έννοιας του δεντροπλάτους, και διατυπώνεται στο θεώρημα που ακολουθεί. Λέμε ότι μία κλάση  $\mathcal{C}$  δομών έχει φραγμένο δεντροπλάτος, αν υπάρχει ένα  $w \geq 1$  τέτοιο ώστε για κάθε δομή  $\mathcal{A} \in \mathcal{C}$  το δεντροπλάτος  $\text{tw}(\mathcal{A})$  της  $\mathcal{A}$  να είναι το πολύ  $w$ . Σε αυτές τις κλάσεις, θεωρούμε ότι το πρόβλημα ελέγχου μοντέλου για μία MSO πρόταση  $\phi$  και μία δομή  $\mathcal{A}$  παραμετροποιείται ως προς το  $|\phi| + \text{tw}(\mathcal{A})$  ή ισοδύναμα ως προς το ζευγάρι  $(|\phi|, \text{tw}(\mathcal{A}))$  [CE]. Σημειώνουμε ότι στην περίπτωση αυτή, το τμήμα  $\text{tw}(\mathcal{A})$  της παραμέτρου δεν είναι πολυωνυμικά υπολογίσιμη συνάρτηση, και για το λόγο αυτό ο αλγόριθμος θεωρείται ασθενώς FPT.

**Θεώρημα 1.1.3 (Θεώρημα του Courcelle [CE, Cou90]).** *Το παραμετρικό πρόβλημα ελέγχου μοντέλου (parameterized model checking problem) για την MSO σε δομές φραγμένου δεντροπλάτους επιλύεται από έναν ασθενώς fixed-parameter linear αλγόριθμο, δηλαδή σε χρόνο*

$$f(|\phi|, \text{tw}(\mathcal{A})) \cdot |\mathcal{A}|.$$

Υπάρχουν αποδείξεις του παραπάνω θεωρήματος που βασίζονται είτε στα αυτόματα είτε στους *type*, και αντιστοιχούν πλήρως στις αποδείξεις του Θεωρήματος 1.1.1. Πράγματι, αυτές οι αποδείξεις αποτελούν κατάλληλες γενικεύσεις των αντίστοιχων που παρουσιάστηκαν για τα δέντρα, δηλαδή τόσο της μεθοδολογίας που βασίζεται στα αυτόματα όσο και εκείνης που χρησιμοποιεί τη μέθοδο σύνθεσης. Πιο συγκεκριμένα, και οι δύο αυτές προσεγγίσεις χρησιμοποιούν μία δεντροδιάσπαση (tree-decomposition) των δομών εισόδου και με αυτό τον τρόπο ανάγουν το αρχικό πρόβλημα —που οριζόταν πάνω σε δομές που δεν ήταν δέντρα—

<sup>3</sup>Ο  $\text{MSO}_q\text{-type}$  μιας δομής  $\mathcal{A}$ , ο οποίος συμβολίζεται με  $\text{tr}_q^{\text{MSO}}(\mathcal{A})$ , ορίζεται ως η κλάση των MSO προτάσεων με βαθμό ποσοδείκτη το πολύ  $q$  που αληθεύουν στη δομή  $\mathcal{A}$ . Εάν  $\bar{a} \in A^k$ , τότε ο  $\text{MSO}_q\text{-type}$  του  $\bar{a}$  στη δομή  $\mathcal{A}$ , ο οποίος συμβολίζεται με  $\text{tr}_q^{\text{MSO}}(\mathcal{A}, \bar{a})$ , ορίζεται ως η κλάση των MSO τύπων  $\phi(x_1, \dots, x_k)$  με βαθμό ποσοδείκτη το πολύ  $q$  για τους οποίους ισχύει ότι  $\mathcal{A} \models \phi(\bar{a})$ .

σε ένα ισοδύναμο πρόβλημα για δέντρα. Αξίζει να σημειωθεί, όπως διατυπώνεται και στο ακόλουθο θεώρημα, ότι οι δεντροδιασπάσεις δομών φραγμένου δεντροπλάτους μπορούν να υπολογιστούν αποτελεσματικά. Το γεγονός αυτό αποτελεί σημαντικό σημείο στην απόδειξη της τελικής γραμμικής πολυπλοκότητας του προβλήματος που θεωρούμε.

**Θεώρημα 1.1.4 ([Bod96]).** Υπάρχει ένας αλγόριθμος ο οποίος, δεδομένης μιας δομής  $\mathcal{A}$  και ενός  $w \geq 1$ , υπολογίζει μία δεντροδιάσπαση της  $\mathcal{A}$  με πλάτος το πολύ  $w$  εάν  $\text{tw}(\mathcal{A}) \leq w$ , και επιστρέφει αρνητική απάντηση εάν  $\text{tw}(\mathcal{A}) > w$ .

Ο χρόνος εκτέλεσης του αλγορίθμου είναι  $2^{O(w^3)} \cdot |\mathcal{A}|$ .

Το παραπάνω αποτέλεσμα του Courcelle επεκτάθηκε στο [ALS91] έτσι ώστε να περιλαμβάνει προβλήματα μέτρησης (counting problems) που ορίζονται μέσω τύπων της MSO. Η προσέγγιση αυτή επεκτάθηκε ακόμα περισσότερο στο [FFG02] όπου αποδεικνύεται το ακόλουθο αποτέλεσμα πολυπλοκότητας για το πρόβλημα αποτίμησης για την MSO (MSO evaluation problem).

**Θεώρημα 1.1.5 ([FFG02]).** Το πρόβλημα της αποτίμησης για την MSO σε δομές φραγμένου δεντροπλάτους επιλύεται από έναν αλγόριθμο ο οποίος είναι γραμμικός ως προς το μέγεθος της δομής εισόδου (input structure) και το μέγεθος της εξόδου (output). Πιο συγκεκριμένα, υπάρχει ένας αλγόριθμος που υπολογίζει το  $\phi(\mathcal{A})$  σε χρόνο

$$f(|\phi|, \text{tw}(\mathcal{A})) \cdot (|\mathcal{A}| + |\phi(\mathcal{A})|).$$

Η απόδειξη που δίνεται στο [FFG02] βασίζεται στην αναγωγή του προβλήματος αποτίμησης για την MSO σε δομές φραγμένου δεντροπλάτους σε ένα ισοδύναμο πρόβλημα αποτίμησης για την MSO σε δέντρα. Πιο συγκεκριμένα, για έναν δεδομένο τύπο  $\phi(X_1, \dots, X_k)$  και μία δομή  $\mathcal{A}$  με δεντροπλάτος  $w$ , ορίζεται ένας νέος τύπος  $\phi^*(\bar{Y}_1, \dots, \bar{Y}_k)$  όπου  $\bar{Y}_i = X_1^1, \dots, X_1^{w+1}$  για  $1 \leq i \leq w+1$  πάνω σε ένα δέντρο  $\mathcal{T}^*$ . Το δέντρο  $\mathcal{T}^*$  κωδικοποιεί την  $\mathcal{A}$  καθώς και μία δεντροδιάσπαση της  $\mathcal{A}$  πλάτους  $w$ . Οι αναθέσεις που ικανοποιούν τον  $\phi^*$  στο  $\mathcal{T}^*$  αποτελούν κωδικοποιημένη μορφή των αναθέσεων που ικανοποιούν τον  $\phi$  στην  $\mathcal{A}$ . Πιο συγκεκριμένα, οι τελευταίες υπολογίζονται ως  $((w+1) \cdot k)$ -διανύσματα υποσυνόλων των κόμβων του  $\mathcal{T}^*$ . Η απόδειξη της συνολικής τελικής πολυπλοκότητας βασίζεται στο γεγονός ότι ο υπολογισμός του  $\phi^*(\mathcal{T}^*)$  μέσω του αλγορίθμου αυτομάτων πραγματοποιείται σε χρόνο  $g(|\phi^*|) \cdot (|\mathcal{T}^*| + |\phi^*(\mathcal{T}^*)|)$ .

**Η συμβολή των βάσεων δεδομένων στην επίλυση προβλημάτων αποτίμησης για την MSO.** Η διατριβή αυτή αποτελεί μία συστηματική database-oriented προσέγγιση στην επίλυση προβλημάτων αποτίμησης για την MSO σε δομές φραγμένου δεντροπλάτους. Προτού προβούμε σε μία αναλυτική παράθεση των αποτελεσμάτων μας, που αποτελούν λύσεις που βασίζονται στη σχεσιακή άλγεβρα και στην datalog, αναφέρουμε ανάλογα αποτελέσματα της σχετικής βιβλιογραφίας. Τα αποτελέσματα αυτά αφορούν αποκλειστικά στην datalog καθώς με βάση όσα είμαστε σε θέση να γνωρίζουμε δεν υπάρχουν στην βιβλιογραφία άλλες δουλειές που να συσχετίζουν την MSO με τη σχεσιακή άλγεβρα.

Στο [GK04], οι Gottlob και Koch μελετούν πάνω σε δέντρα μή φραγμένης τάξης (unranked trees) MSO τύπους της μορφής  $\phi(x)$ , δηλαδή τύπους που έχουν μία μόνο πρωτοβάθμια ελεύθερη μεταβλητή. Η επιλογή αυτή βασίζεται στο γεγονός ότι το συγκεκριμένο τμήμα της MSO διαθέτει επαρκή εκφραστικότητα για την ανάκτηση πληροφορίας από τον παγκόσμιο ιστό. Παρακινούμενοι από το γεγονός αυτό, αποδεικνύουν ότι αυτό το τμήμα της MSO πάνω σε δέντρα είναι ισοδύναμο με την μοναδιαία datalog σε δέντρα ως προς την ικανότητά του να εκφράζει μονομελή ερωτήματα (unary queries).

Προτού δούμε τις λεπτομέρειες αυτής της προσέγγισης παραθέτουμε κάποιους βασικούς ορισμούς. Ένα μονομελές ερώτημα (unary query)  $q$  σε μία κλάση  $\mathcal{C}$  από δέντρα είναι μία απεικόνιση  $\mathcal{T} \mapsto \mathcal{P}(T)$  που αντιστοιχεί κάθε δέντρο  $\mathcal{T}$  της  $\mathcal{C}$  σε ένα υποσύνολο του συνόλου  $T$  των κόμβων του. Ένα μονομελές MSO-ορίσιμο ερώτημα (unary MSO-definable query)  $q$  σε δέντρα είναι ένα ερώτημα που ορίζεται από έναν MSO τύπο  $\phi(x)$ , δηλαδή

$$q = q_\phi : \mathcal{T} \mapsto \phi(\mathcal{T})$$

όπου  $\phi(\mathcal{T}) = \{n \in T \mid \mathcal{T} \models \phi(n)\}$ . Η μοναδιαία datalog (monadic datalog) είναι εκείνο το τμήμα της datalog που αποτελείται από κανόνες που έχουν την εξής ιδιαιτερότητα: κάθε σύμβολο κατηγορήματος που εμφανίζεται στην κεφαλή ενός κανόνα έχει το πολύ μία μεταβλητή. Ένα ερώτημα της μοναδιαίας datalog (monadic datalog query) είναι ένα ερώτημα που ορίζεται από ένα πρόγραμμα της μοναδιαίας datalog και κάποιο σύμβολο κατηγορήματος που εμφανίζεται στην κεφαλή ενός κανόνα του προγράμματος. Λέμε ότι ένα ερώτημα είναι ορίσιμο στη (μοναδιαία) datalog αν υπάρχει ένα (μοναδιαίο) datalog πρόγραμμα που το υπολογίζει.

Είμαστε τώρα έτοιμοι να σκιαγραφήσουμε την απόδειξη που δίνεται στο [GK04]: η πρώτη κατεύθυνση, δηλαδή το γεγονός ότι κάθε ερώτημα της μοναδιαίας datalog είναι MSO-ορίσιμο μέσω ενός τύπου  $\phi(x)$ , αποτελεί γνωστό αποτέλεσμα των βάσεων δεδομένων. Η απόδειξη για την αντίστροφη κατεύθυνση, δηλαδή για το ότι κάθε μονομελές MSO-ορίσιμο ερώτημα (σε δέντρα) μπορεί να εκφραστεί στην μοναδιαία datalog, βασίζεται στην μέθοδο σύνθεσης των Feferman-Vaught (βλέπε το σχολιασμό που έπεται του Θεωρήματος 1.1.1). Πιο συγκεκριμένα, το πρόβλημα επιλύεται μέσω κανόνων της datalog που εκφράζουν τον υπολογισμό των  $\text{MSO}_q$ -type των κόμβων του δέντρου εισόδου. Τα datalog προγράμματα που προκύπτουν με αυτό τον τρόπο έχουν γραμμική πολυπλοκότητα ίση με  $O(|\Pi| \cdot |T|)$  όπου το  $|\Pi|$  δηλώνει το μέγεθος του προγράμματος και το  $|T|$  το μέγεθος του δέντρου εισόδου.

Αναλυτικότερα, αν υποθέσουμε ότι ο βαθμός ποσοδείκτη του MSO τύπου  $\phi(x)$  είναι  $q$ , τότε το αντίστοιχο datalog πρόγραμμα υπολογίζει τους  $\text{MSO}_q$ -type για τα υποδέντρα  $\mathcal{T}_n$  και για τα συμπληρώματά τους  $\overline{\mathcal{T}}_n$ , διατρέχοντας το δέντρο εισόδου με κατεύθυνση από τα φύλλα προς τη ρίζα και με κατεύθυνση από τη ρίζα προς τα φύλλα αντίστοιχα. Πιο συγκεκριμένα, σε κάθε κόμβο  $n \in T$  υπολογίζονται οι type  $\text{tr}_q^{\text{MSO}}(\mathcal{T}_n, n)$  και  $\text{tr}_q^{\text{MSO}}(\overline{\mathcal{T}}_n, n)$ , οι οποίοι και αποτελούν τα ονόματα των κατηγορημάτων του datalog προγράμματος που ορίζεται με αυτόν τον τρόπο. Από την κατάλληλη σύνθεση των type που υπολογίζονται σε κάθε κόμβο, προκύπτει η απάντηση στο ερώτημα. Το datalog πρόγραμμα που προκύπτει με αυτό τον τρόπο είναι ισοδύναμο με εκείνο που προκύπτει από την προσομοίωση των αυτομάτων ερωτημάτων (query automata) του [NS02] στη μοναδιαία datalog. Τα αυτόματα ερωτημάτων είναι ντετερμινιστικά

διπλής κατεύθυνσης (two-way) αυτόματα μαζί με μία συνάρτηση επιλογής η οποία τα καθιστά ικανά να ορίζουν μονομελή ερωτήματα σε δέντρα. Ο ορισμός των αυτομάτων αυτών βασίζεται στη μέθοδο σύνθεσης ενώ έχει αποδειχθεί ότι ένα μονομελές ερώτημα σε δέντρα είναι MSO-ορίσιμο αν υπάρχει ένα αυτόματο ερωτημάτων που το υπολογίζει. Έτσι, το πρόγραμμα που προσομοιώνει τα αυτόματα ερωτημάτων, και δίνεται επίσης στο [GK04], αποτελεί μία διαφορετική απόδειξη της ορισιμότητας (definability) των μονομελών MSO-ορίσιμων ερωτημάτων στη μοναδιαία datalog.

Στο [GPW10], οι Gottlob, Pichler και Wei αποδεικνύουν ότι κάθε μονομελές MSO-ορίσιμο ερώτημα σε πεπερασμένες δομές φραγμένου δεντροπλάτους είναι επίσης ορίσιμο στη μοναδιαία datalog. Γενικεύουν έτσι το αντίστοιχο αποτέλεσμα εκφρασιμότητας του [GK04] που αφορούσε σε πεπερασμένα δέντρα. Όπως και στην περίπτωση των δέντρων, η προσέγγισή τους βασίζεται στον υπολογισμό των  $MSO_q$ -type των υποδομών που προκύπτουν από τα υποδέντρα των δεντροδιασπάσεων. Πιο συγκεκριμένα, γενικεύουν αρχικά τα αποτελέσματα που αφορούν στους  $q$ -type από τα δέντρα στις δομές φραγμένου δεντροπλάτους και στη συνέχεια εκφράζουν τα νέα θεωρήματα σύνθεσης που προκύπτουν στην datalog. Το datalog πρόγραμμα που προκύπτει είναι ένα μοναδιαίο πρόγραμμα που αποτιμάται σε χρόνο  $O(|\Pi| \cdot |\mathcal{A}|)$ . Έτσι, αποδεικνύουν ότι το πρόβλημα της αποτίμησης για το τμήμα της MSO που αποτελείται από τύπους  $\phi(x)$  σε μία δομή  $\mathcal{A}$  επιλύεται σε χρόνο  $O(f(|\phi|, tw(\mathcal{A})) \cdot |\mathcal{A}|)$ . Αξίζει να σημειωθεί ότι η  $f(|\phi|, tw(\mathcal{A}))$  φράσσει τόσο την κατασκευή όσο και το μέγεθος  $|\Pi|$  του μοναδιαίου datalog προγράμματος  $\Pi$  που προκύπτει τελικά. Καθώς ο αριθμός των  $MSO_q$ -type δεν φράσσεται από καμία στοιχειώδη συνάρτηση του  $q$ , κάθε αλγόριθμος που βασίζεται σε αυτή την κατασκευή έχει μη-στοιχειώδη παραμετρική εξάρτηση (όπως προκύπτει άλλωστε και από το Θεώρημα 1.1.2). Η προσέγγιση αυτή υλοποιείται για συγκεκριμένα προβλήματα όπως τα 3-COLORABILITY και PRIMALITY για τα οποία γράφονται προγράμματα στη C++ που βασίζονται σε datalog αλγόριθμους.

Στο [PRW10] επεκτείνεται η datalog προσέγγιση του [GPW10] στην επίλυση *προβλημάτων μέτρησης (counting problems)* που ορίζονται μέσω της MSO σε δομές φραγμένου δεντροπλάτους. Τα προβλήματα αυτά συνίστανται στον υπολογισμό του αριθμού των αναθέσεων που ικανοποιούν έναν MSO τύπο  $\phi$  σε μία δεδομένη δομή. Βασιζόμενοι στους MSO  $q$ -type, αποδεικνύουν ότι το πρόβλημα της μέτρησης για την MSO είναι ορίσιμο σε μία επέκταση της datalog η οποία διαθέτει μετρητές (counters) και τη συναθροιστική συνάρτηση (aggregate function) SUM. Επιπρόσθετα, δείχνουν πώς η datalog επίλυση του προβλήματος μέτρησης μπορεί να χρησιμοποιηθεί για να ανακτηθεί η λύση στο αντίστοιχο πρόβλημα απαρίθμησης για την MSO (MSO enumeration problem). Ωστόσο, δεν δίνεται datalog λύση για αυτό το τελευταίο πρόβλημα. Η προσέγγισή τους επεξηγείται μέσω της εφαρμογή της για την επίλυση των αντίστοιχων παραλλαγών του 3-COLORABILITY.

Στην παρούσα διατριβή, παρέχουμε μία αναλυτική μελέτη των λύσεων που βασίζονται στην datalog και επιλύουν το πρόβλημα της αποτίμησης για *ολόκληρη* την MSO, θεωρώντας αρχικά το πρόβλημα σε πεπερασμένα δέντρα και έπειτα ανεξάρτητα σε δομές φραγμένου δεντροπλάτους. Επεκτείνουμε κατά συνέπεια τις υπάρχουσες λύσεις που βασίζονται στην datalog και αφορούν στην επίλυση του προβλήματος της αποτίμησης για MSO τύπους  $\phi(x)$ , σε ολόκληρη την κλάση των MSO τύπων  $\phi(x_1, \dots, x_\ell, X_1, \dots, X_k)$ . Σε αντίθεση με τις δουλειές

της υπάρχουσας βιβλιογραφίας που αναφέρθηκαν παραπάνω, οι οποίες βασίζονται στη μεταφορά της ιδέας της μεθόδου σύνθεσης των Feferman-Vaught στο πλαίσιο της datalog, η δουλειά μας βασίζεται στη μεταφορά της προσέγγισης που βασίζεται στη θεωρία αυτομάτων (παραπέμπουμε στις σκιαγραφήσεις των αποδείξεων του Θεωρήματος 1.1.1 για μία σύντομη παρουσίαση αυτών των δύο διαφορετικών προσεγγίσεων στην περίπτωση των δέντρων). Πιο συγκεκριμένα, εκφράζουμε κατασκευές κατάλληλων αυτομάτων για δέντρα στην datalog και στη συνέχεια επιλύουμε τα αντίστοιχα προβλήματα υπολογισμού αυτομάτων μέσω *μεθόδων βελτιστοποίησης της datalog*. Στην περίπτωση των δέντρων, τα αυτόματα που κατασκευάζουμε μέσω της datalog είναι στην ουσία τα Thatcher-Wright αυτόματα (οι καταστάσεις και οι μεταβάσεις τους αποτελούν ονόματα κατηγορημάτων). Στην περίπτωση των δομών φραγμένου δεντροπλάτους, τα αυτόματά μας αντιστοιχούν σε Thatcher-Wright αυτόματα που αποδέχονται επεκτάσεις δεντροδιασπάσεων που κωδικοποιούν τις αναθέσεις που ικανοποιούν τον  $\phi$  στη δεδομένη δομή  $\mathcal{A}$ . Μάλιστα, τα αυτόματα αυτά είναι ισοδύναμα με τα αυτόματα που αντιστοιχούν στον μετασχηματισμένο τύπο  $\phi^*$  που δίνεται στην απόδειξη του Θεωρήματος 1.1.5 του [FFG02]. Εντούτοις, είναι ιδιαίτερα σημαντικό το γεγονός ότι η δουλειά μας παρέχει μία απευθείας κατασκευή του αυτομάτου αυτού μέσω του αρχικού τύπου  $\phi$ , η οποία δεν προϋποθέτει την κατασκευή ενός πολύ πιο σύνθετου τύπου  $\phi^*$  (βλέπε τα σχόλια μετά το Θεώρημα 1.1.5).

Επιπρόσθετα, επεκτείνουμε την προσέγγισή μας για να επιλύσουμε *προβλήματα βελτιστοποίησης* που ορίζονται μέσω τύπων της MSO σε δομές φραγμένου δεντροπλάτους, εξακολουθώντας να έχουμε ως βάση τη θεωρία αυτομάτων. Σημειώνουμε ότι για την επίλυση σε αυτές τις περιπτώσεις, ορίζουμε κατάλληλες επεκτάσεις της datalog αλλά και των βάσεων δεδομένων που αποτελούν την είσοδο στα προγράμματά μας. Ανεξάρτητα από τις λύσεις μέσω της datalog, δίνουμε μία λύση που βασίζεται στη σχεσιακή άλγεβρα και αφορά στο πρόβλημα της αποτίμησης της MSO σε δέντρα. Ένα ιδιαίτερα ενδιαφέρον σημείο αυτής της προσέγγισης είναι ότι δια φωτίζει τη στενή σύνδεση ανάμεσα σε διαφορετικούς αλγορίθμους βελτιστοποίησης των βάσεων δεδομένων όπως ο αλγόριθμος του Yannakakis και οι μέθοδοι αναγραφής της datalog (datalog rewriting methods).

Μέσω των διαφορετικών αναγωγών των προβλημάτων αποτίμησης της MSO σε προβλήματα αποτίμησης της datalog που παρέχουμε, αποδεικνύουμε *ενδιαφέροντα αποτελέσματα εκφρασιμότητας* για την datalog και την MSO. Πιο συγκεκριμένα, ξανά-αποδεικνύουμε τα αποτελέσματα ορισιμότητας των [GK04, GPW10] που αφορούν στα μονομελή MSO-ορίσιμα ερωτήματα και τα γενικεύουμε για κάθε ερώτημα που ορίζεται μέσω οποιουδήποτε τύπου της MSO πάνω σε δέντρα και σε δομές φραγμένου δεντροπλάτους. Ειδικότερα, ανάμεσα σε άλλα, αποδεικνύουμε ότι τα  $k$ -μελή MSO-ορίσιμα ερωτήματα είναι ορίσιμα στην datalog που έχει intentional κατηγορήματα με αριθμό μεταβλητών το πολύ  $k + 1$ .

Εκτός από τα παραπάνω αναφερόμενα αποτελέσματα εκφρασιμότητας, πορίσματα των προσεγγίσεών μας, που βασίζονται στην datalog και τη σχεσιακή άλγεβρα, είναι *εναλλακτικές αποδείξεις πολυπλοκότητας* —μέσω εργαλείων που βασίζονται στις βάσεις δεδομένων— για τα προβλήματα αποτίμησης της MSO. Δηλαδή ξανά-αποδεικνύουμε τα αποτελέσματα πολυπλοκότητας των Θεωρημάτων 1.1.1, 1.1.3 και 1.1.5. Θα πρέπει να τονίσουμε ότι το γεγονός

ότι έχουμε τα ίδια φράγματα πολυπλοκότητας (complexity bounds) βασίζεται στη βελτιστοποίηση που εφαρμόζουμε στον υπολογισμό μας, δηλαδή στην εφαρμογή του αλγορίθμου του Yannakakis και στις μεθόδους αναγραφής της datalog.

Αξίζει ωστόσο να σημειωθεί ότι η FPT πολυπλοκότητα των datalog αλγορίθμων δε συνεπάγεται απαραίτητα αλγορίθμους που είναι πρακτικά χρήσιμοι, καθώς δεν μπορούμε να αποφύγουμε τη μη-στοιχειώδη παραμετρική εξάρτηση που επιβάλλεται από το μέγεθος του αυτομάτου (βλέπε επίσης το Θεώρημα 1.1.2). Πιο συγκεκριμένα, το μη-στοιχειώδες μέγεθος των αυτομάτων που προκύπτουν καθορίζει το μέγεθος των datalog προγραμμάτων μας, κατά τον ίδιο τρόπο που ο αριθμός των MSO-type καθορίζει το μέγεθος των datalog προγραμμάτων στο [GPW10].

Παρόλα αυτά, λόγω της λειτουργικής σημασιολογίας (operational semantics) που διαθέτει η datalog, η ύπαρξη μιας γενικής λύσης για τα προβλήματα αποτίμησης της MSO η οποία βασίζεται αποκλειστικά στην datalog δε θα πρέπει να υποτιμάται, καθώς αποτελεί ένα βασικό βήμα προς την κατεύθυνση παραγωγής υλοποιήσιμων αλγορίθμων. Για παράδειγμα, θέτει τις κατευθυντήριες γραμμές για περαιτέρω βελτιστοποιήσεις στις περιπτώσεις που ασχολούμαστε με συγκεκριμένα προβλήματα για τα οποία μία αποτελεσματική κατασκευή αυτομάτου είναι εφικτή. Παρουσιάζουμε τέτοιες κατευθύνσεις μέσω της εφαρμογής της datalog προσέγγισής μας στην περίπτωση του τύπου που εκφράζει το VERTEX-COVER, από την οποία προκύπτουν νέοι αλγόριθμοι για τα προβλήματα που σχετίζονται με το VERTEX-COVER. Η εφαρμογή αυτή λειτουργεί ως μέσο έκθεσης του τρόπου που αυτή η προσέγγιση μπορεί να χρησιμοποιηθεί ώστε να παραχθούν αποτελεσματικοί αλγόριθμοι. Σημειώνουμε ότι δεν παρουσιάζουμε κάποια υλοποίηση αυτού του αλγορίθμου σε προγραμματιστικό περιβάλλον.

Μια πιο αναλυτική έκθεση των αποτελεσμάτων μας δίνεται στη επόμενη ενότητα.

## 1.2 Αναλυτική επισκόπηση της διατριβής

Το κύριο θέμα αυτής της διατριβής είναι η ανάπτυξη *βελτιστοποιημένων database-oriented λύσεων* για προβλήματα αποτίμησης της MSO σε πεπερασμένα δέντρα και δομές που έχουν δεντρική μορφή. Και στις δυο αυτές περιπτώσεις, δηλαδή τόσο στην περίπτωση των δέντρων όσο και στις δομές φραγμένου δεντροπλάτους, η θεωρητική βάση της προσέγγισής μας είναι η κλασική ισοδυναμία ανάμεσα στην MSO με τα αυτόματα σε δέντρα που διατυπώνεται στο θεώρημα των Doner [Don70], Thatcher & Wright [TW68]. Πιο συγκεκριμένα, επαναπροσδιορίζουμε αυτή τη σχέση μέσω ενός νέου φορμαλισμού αυτομάτων, που ονομάζονται αυτόματα αναθέσεων, και τη γενικεύουμε στις δομές φραγμένου δεντροπλάτους. Με αυτό τον τρόπο, ανάγουμε τα αρχικά προβλήματα αποτίμησης της MSO σε ισοδύναμα προβλήματα που αφορούν στα αυτόματα αναθέσεων. Μέσω της μοντελοποίησης αυτών των τελευταίων προβλημάτων αποτίμησης αυτομάτων σε ένα πλαίσιο βάσεων δεδομένων, επιλύουμε τα αρχικά προβλήματα αποτίμησης της MSO μέσω βελτιστοποιημένων αλγορίθμων από το χώρο των βάσεων δεδομένων.

Τα αποτελέσματά μας παρουσιάζονται σε δυο κεφάλαια: η περίπτωση των δέντρων εξετάζεται στο Κεφάλαιο 2 και η περίπτωση των δομών φραγμένου δεντροπλάτους στο Κεφάλαιο

3. Κάθε κεφάλαιο είναι αυτόνομο και μπορεί να διαβαστεί ανεξάρτητα από το άλλο. Το περιεχόμενο του Κεφαλαίου 2 έχει δημοσιευτεί στο [FK09c] (επεκτεταμένη σύνοψη του οποίου αποτελεί το [FK09a]), ενώ σχετικά θέματα βρίσκονται στα [FK05], [FK09b]. Το μεγαλύτερο μέρος του Κεφαλαίου 3 αποτελεί το [FK11a] του οποίου περιληπτική μορφή είναι το [FK11b], ενώ μια πρώιμη μορφή των αντίστοιχων αποτελεσμάτων έχει παρουσιαστεί στο [FK10].

Στη συνέχεια παρουσιάζεται μία αναλυτική επισκόπηση του περιεχομένου και της δομής των κεφαλαίων αυτών.

**Επισκόπηση του Κεφαλαίου 2.** Στο κεφάλαιο αυτό παρουσιάζονται αναγωγές του προβλήματος της αποτίμησης της MSO πάνω σε *πεπερασμένα δέντρα*, σε προβλήματα αποτίμησης από το χώρο των βάσεων δεδομένων. Τα προβλήματα αυτά εκφράζονται τόσο στη σχεσιακή άλγεβρα όσο και στην datalog. Πιο συγκεκριμένα, έχοντας ως βάση την ισοδυναμία MSO–αυτομάτων για δέντρα, ορίζουμε στην Ενότητα 2.2 ένα νέο αυτόματο που συμβολίζουμε  $\text{Assign}_\phi$  και το οποίο ονομάζουμε *αυτόματο αναθέσεων*. Το  $\text{Assign}_\phi$  δεν είναι ένα συμβατικό αυτόματο, δηλαδή ένα αυτόματο που αναγνωρίζει. Αντίθετα, μέσω της προσθήκης ενός διανύσματος αναθέσεων στις καταστάσεις του, μετατρέπεται σε ένα *μηχανισμό διατύπωσης ερωτημάτων*, δηλαδή αποτελεί ένα νέο φορμαλισμό που *εκφράζει MSO-ορίσιμα ερωτήματα*. Αποδεικνύουμε ότι για κάθε τύπο  $\phi$  της MSO μπορεί να κατασκευαστεί ένα αυτόματο αναθέσεων το οποίο υπολογίζει τις αναθέσεις που ικανοποιούν τον  $\phi$  σε δέντρα. Σημειώνουμε ότι τα αυτόματα αναθέσεων γενικεύουν σε όλη την MSO την ιδέα των αυτομάτων ερωτημάτων (query automata ή QA) του [NS02] και των αυτομάτων επιλογής σε δέντρα (selecting tree automata ή STA) του [FGK03] τα οποία υπολογίζουν τις αναθέσεις που ικανοποιούν τύπους της MSO της μορφής  $\phi(x)$ , δηλαδή με μία πρωτοβάθμια ελεύθερη μεταβλητή. Μάλιστα, όταν περιοριστούμε σε μονομελή MSO-ορίσιμα ερωτήματα, δηλαδή ερωτήματα που ορίζονται από MSO τύπους  $\phi(x)$ , οι καταστάσεις του  $\text{Assign}_\phi$  μετατρέπονται σε καταστάσεις που επιλέγουν όπως συμβαίνει με τις καταστάσεις των QA και STA.

Στην Ενότητα 2.3, προτείνουμε μία φυσική αναγωγή του προβλήματος της αποτίμησης της MSO σε ένα πρόβλημα αποτίμησης ενός ακυκλικού συζευκτικού ερωτήματος  $\mathcal{Q}$ . Η ιδέα της αναγωγής είναι η αναπαράσταση των εκτελέσεων του  $\text{Assign}_\phi$  μέσω της συνένωσης (join) των αντίστοιχων σχέσεων μετάβασης (transition relations). Ο *αλγόριθμος του Γιαννακάκης* [Yan81] είναι ένας αποδοτικός (efficient) αλγόριθμος για την αποτίμηση ακυκλικών συζευκτικών ερωτημάτων. Μέσω της εφαρμογής του για την αποτίμηση του  $\mathcal{Q}$ , δίνουμε μία *βέλτιστη λύση*, που βασίζεται στη σχεσιακή άλγεβρα, για το πρόβλημα της αποτίμησης της MSO σε πεπερασμένα δέντρα. Η ύπαρξη συναρτησιακών εξαρτήσεων (functional dependencies) στις σχέσεις εισόδου των στιγμιοτύπων που εξετάζουμε –αποδεικνύονται συνεκτιμώντας κάποιες ιδιαιτερότητές τους που κληρονομούνται από τα χαρακτηριστικά των μεταβάσεων που περιγράφουν– συνεπάγεται βελτιωμένα, σε σχέση με τη γενική περίπτωση, αποτελέσματα πολυπλοκότητας για τον αλγόριθμο του Γιαννακάκη. Η αντίστοιχη απόδειξη υποδεικνύει μία σαφή και απλή ανάλυση πολυπλοκότητας για το πρόβλημα της αποτίμησης της MSO και παρέχει μία *εναλλακτική απόδειξη*, μέσω της *σχεσιακής άλγεβρας*, της *πολυπλοκότητας που δίνεται στο* [FFG02] και αναφέρεται στο ίδιο πρόβλημα.



Αυτή είναι η πρώτη φορά που το πρόβλημα της αποτίμησης ενός τύπου της MSO εκφράζεται στη σχεσιακή άλγεβρα και επιλύεται από το βελτιστοποιημένο αλγόριθμο του Yannakakis. Ένα ιδιαίτερα ενδιαφέρον σημείο, που αποδεικνύεται μέσω της αναγωγής αυτής, είναι ότι ο αλγόριθμος του Yannakakis, εφαρμοσμένος σε ένα συγκεκριμένο συζευκτικό ερώτημα και συγκεκριμένη είσοδο, ουσιαστικά συμπίπτει με τον αλγόριθμο αυτομάτων των Flum, Frick, Grohe που δίνεται στο [FFG02] –θα τον αποκαλούμε «FFG αλγόριθμο» στη συνέχεια της διατριβής–. Ο FFG αλγόριθμος έχει ως είσοδο ένα αυτόματο για τον  $\phi$  και ένα δέντρο  $\mathcal{T}$ , και υπολογίζει το σύνολο  $\phi(\mathcal{T})$  των αναθέσεων που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$ . Η συγκριτική μελέτη αυτών των δύο αλγορίθμων, οδηγεί σε μία βαθύτερη κατανόηση των εγγενών ομοιοτήτων τους που απορρέουν από το γεγονός ότι ο κεντρικός τους πυρήνας είναι η απαλοιφή των άχρηστων τμημάτων της εισόδου και ο υπολογισμός της εξόδου μέσω της σύνθεσης ακριβώς εκείνων των τμημάτων από τα οποία αποτελείται. Ο συσχετισμός των δύο αλγορίθμων επιστημονήθηκε αρχικά στο [FFG02], εντούτοις –χάρη στην αναγωγή μας– καταφέραμε να προσδιορίσουμε την ακριβή φύση της σχέσης τους που συνοψίζεται στην επόμενη φράση. Ο αλγόριθμος του Yannakakis περιλαμβάνει τον FFG αλγόριθμο, με την έννοια ότι ο τελευταίος αποτελεί ένα συγκεκριμένο «στιγμιότυπο» του πρώτου.

Στην Ενότητα 2.4, προτείνουμε διάφορες αναγωγές του προβλήματος αποτίμησης της MSO σε προβλήματα αποτίμησης για την datalog. Όπως και στην περίπτωση της σχεσιακής άλγεβρας, η ιδέα των αναγωγών αυτών είναι η προσομοίωση των εκτελέσεων του  $\text{Assign}_\phi$  στην datalog. Η μελέτη των αντίστοιχων προβλημάτων πραγματοποιείται σε δύο ανεξάρτητα μέρη: στο πρώτο μέρος όλα τα datalog ερωτήματα χρησιμοποιούν το  $\text{Assign}_\phi$  για να οριστούν, ενώ στο δεύτερο μέρος ορίζουμε τα ίδια datalog ερωτήματα έχοντας ως είσοδο τον  $\phi$  χωρίς να προϋποθέτουμε την κατασκευή του  $\text{Assign}_\phi$ . Μέσω της τελευταίας προσέγγισης, επιτυγχάνουμε μία πλήρη αναγωγή του προβλήματος αποτίμησης της MSO πάνω σε δέντρα σε ένα πρόβλημα αποτίμησης ενός datalog ερωτήματος, δηλαδή παρέχουμε μία λύση που βασίζεται πλήρως στις βάσεις δεδομένων.

Πιο συγκεκριμένα, δίνουμε αρχικά τρία ισοδύναμα datalog ερωτήματα  $\mathcal{Q}_{\phi,m}$  τα οποία υπολογίζουν –σε κωδικοποιημένη μορφή– τις αναθέσεις που ικανοποιούν τον  $\phi$  σε δέντρα μεγέθους το πολύ  $m$ . Το πρώτο από αυτά τα ερωτήματα είναι μη-βελτιστοποιημένο, ενώ τα άλλα δύο αποτελούν βελτιστοποιημένες εκδοχές του πρώτου οι οποίες προκύπτουν μέσω της εφαρμογής των ιδεών του *φιλτράρισματος που βασίζεται στην επίλυση (resolution-based filtering)*.<sup>4</sup> Το φιλτράρισμα επίλυσης είναι μία τεχνική βελτιστοποίησης που βασίζεται στην αναγραφή, και συνίσταται στον υπολογισμό γεγονότων (facts) τα οποία, όταν εισάγονται στο σώμα των κανόνων, απαλείφουν κάποιες από τις στιγμιοτυποποιήσεις κανόνων (rule instantiations) που δε συμμετέχουν στον υπολογισμό του δεδομένου ερωτήματος. Το πρόγραμμα που παράγει τα γεγονότα μέσω των οποίων επιτυγχάνεται το φιλτράρισμα ονομάζεται *φίλτρο*. Λέμε ότι ένα φίλτρο είναι *βέλτιστο* όταν τα γεγονότα που παράγει κατορθώνουν να απαλείψουν όλα τα «άχρηστα» στιγμιότυπα κανόνων. Το φίλτρο που συμμετέχει στη βελτιστοποίηση

<sup>4</sup>Το «φιλτράρισμα που βασίζεται στην επίλυση» (ή «φιλτράρισμα επίλυσης» για συντομία) είναι η μέθοδος αναγραφής (rewriting method) για την datalog η οποία αναφέρεται συχνά στη βιβλιογραφία με το όνομα «magic sets» (βλέπε επίσης το [BMSU86]).

του τρίτου από τα ερωτήματα που αναφέραμε παραπάνω είναι βέλτιστο. Πράγματι, αυτό το τρίτο ερώτημα σχετίζεται στενά με τον αλγόριθμο του Yannakakis, και έτσι η σχέση του τελευταίου με τον FFG αλγόριθμο, που αναδείχθηκε στο πλαίσιο της σχεσιακής άλγεβρας, ισχυροποιείται σε μία ισοδυναμία που συμπεριλαμβάνει ένα αλγόριθμο αναγραφής που αντιστοιχεί στη «βελτιστοποιημένη» εκδοχή της μεθόδου του φιλτραρίσματος επίλυσης.

Συνεχίζουμε με τις αναγωγές μας ορίζοντας ένα νέο βελτιστοποιημένο datalog ερώτημα το οποίο διαθέτει βέλτιστο φίλτρο. Για την ακρίβεια, αυτό το ερώτημα συνιστά μία παραλλαγή του τρίτου ερωτήματος που ορίστηκε προηγούμενα αλλά, σε αντίθεση με το τελευταίο, αυτό υπολογίζει απευθείας τις αναθέσεις –και όχι μια κωδικοποιημένη εκδοχή τους– και επιπρόσθετα δεν παρουσιάζει καμία εξάρτηση από τον μέγεθος του δέντρου εισόδου. Πιο συγκεκριμένα, αποδεικνύουμε ότι για κάθε MSO τύπο  $\phi(X_1, \dots, X_k)$  μπορεί να κατασκευαστεί ένα datalog ερώτημα  $\mathcal{Q}_\phi$ , με πλήθος μεταβλητών  $(k+1)$ , το οποίο υπολογίζει τις αναθέσεις που ικανοποιούν τον  $\phi$  σε κάθε δέντρο. Συνεπώς αποδεικνύουμε ότι κάθε  $k$ -μελές MSO-ορίσιμο ερώτημα είναι ορίσιμο στην  $(k+1)$ -datalog, δηλαδή στην datalog με κανόνες που έχουν την εξής ιδιαιτερότητα: τα κατηγορήματα που εμφανίζονται στις κεφαλές των κανόνων έχουν αριθμό μεταβλητών το πολύ  $k+1$ . Αυτό το αποτέλεσμα εκφρασιμότητας επεκτείνει το αντίστοιχο του [GK04] ότι δηλαδή μονομελή MSO-ορίσιμα ερωτήματα είναι ορίσιμα στη μοναδιαία datalog.

Επιπλέον, δίνουμε μία λύση για το πρόβλημα της αποτίμησης της MSO μέσω ενός μοναδιαίου datalog προγράμματος. Αυτό το μοναδιαίο πρόγραμμα ορίζει μία οικογένεια από ερωτήματα της μοναδιαίας datalog τα οποία υπολογίζουν τα στοιχεία των αναθέσεων που ικανοποιούν τον δεδομένο τύπο. Τέλος, ολοκληρώνουμε το κεφάλαιο εξετάζοντας ξεχωριστά ειδικές περιπτώσεις MSO τύπων όπως είναι οι προτάσεις, οι τύποι που ορίζουν μονομελή ερωτήματα και οι τύποι με  $n$  πρωτοβάθμιες ελεύθερες μεταβλητές. Για κάθε μία από αυτές τις περιπτώσεις, δίνεται ένα datalog πρόγραμμα που επιλύει το αντίστοιχο πρόβλημα αποτίμησης και πραγματοποιείται εκτίμηση της πολυπλοκότητάς του.

Σημειώνουμε ότι οι προσεγγίσεις της βιβλιογραφίας που συσχετίζουν το πρόβλημα της αποτίμησης της MSO με την datalog (βλέπε τα [GK04, GS03]), δεν αναφέρονται σε ολόκληρη την MSO. Από όσο είμαστε σε θέση να γνωρίζουμε, αυτή είναι η πρώτη φορά που i) δίνεται μία datalog λύση που αναφέρεται σε ολόκληρη την MSO και επιπρόσθετα βασίζεται σε μεθόδους βελτιστοποίησης της datalog, και ii) πραγματοποιείται μία ανάλυση πολυπλοκότητας για το πρόβλημα της αποτίμησης της MSO σε πεπερασμένα δέντρα χρησιμοποιώντας datalog.

**Επισκόπηση του Κεφαλαίου 3.** Στο κεφάλαιο αυτό γενικεύουμε τις λύσεις που παρουσιάζονται στο Κεφάλαιο 2, και αφορούν αποκλειστικά σε δομές που είναι δέντρα, στην περίπτωση των δομών φραγμένου δεντροπλάτους. Επιπρόσθετα, παράλληλα με το πρόβλημα της αποτίμησης της MSO, εξετάζουμε τρία προβλήματα βελτιστοποίησης που ορίζονται μέσω MSO τύπων της μορφής  $\phi(X)$  στις ίδιες κλάσεις δομών. Η αποτίμηση αυτών των τριών προβλημάτων, τα οποία ονομάζουμε MSO-ορίσιμα προβλήματα βελτιστοποίησης (MSO-definable optimization problems), συνίσταται i) στον υπολογισμό της απάντησης στο ερώτημα εάν υπάρχει ένα σύνολο  $X$  μεγέθους το πολύ  $k$  που ικανοποιεί τον MSO τύπο  $\phi(X)$ , ii) στον υπολογισμό του πληθάριθμου των αναθέσεων που ικανοποιούν τον  $\phi(X)$  και έχουν ελάχιστο (αντίστοιχα μέ-

γιστο) μέγεθος, και iii) στον υπολογισμό του συνόλου των αναθέσεων ελάχιστου (αντίστοιχα μέγιστου) μεγέθους που ικανοποιούν τον  $\phi(X)$ . Σε όλα τα προβλήματα που εξετάζουμε, εκτός από έναν MSO τύπο  $\phi$  και μία πεπερασμένη δομή  $\mathcal{A}$ , θεωρούμε ως μέρος της εισόδου μία δεντροδιάσπαση  $\mathcal{I}$  της  $\mathcal{A}$ . Για κάθε ένα από τα προβλήματα αυτά, δίνουμε λύσεις που βασίζονται τόσο στα αυτόματα όσο και στην datalog.

Στο πρώτο μέρος αυτού του κεφαλαίου, παρουσιάζουμε μία απευθείας αναγωγή του αρχικού προβλήματος, δηλαδή του προβλήματος αποτίμησης της MSO σε δομές φραγμένου δεντροπλάτους, σε ένα ισοδύναμο πρόβλημα αποτίμησης αυτομάτων σε δέντρα. Πιο συγκεκριμένα, ορίζουμε ένα νέο φορμαλισμό αυτομάτων, με την ονομασία *αυτόματα αποδόμησης* (*decomposition automata*), τα οποία τρέχουν σε δέντρα που αποδομούν τις αρχικές δομές. Μία τέτοια δεντρική αποδόμηση της αρχικής δομής, ορίζεται ως προς τη δεντροδιάσπαση που είναι μέρος της εισόδου των προβλημάτων που εξετάζουμε, και κατασκευάζεται μέσω της επέκτασης της δεδομένης δεντροδιάσπασης με κατάλληλες κωδικοποιήσεις των σχέσεων της αρχικής δομής. Αποτελεί δηλαδή μία δεντρική κωδικοποίηση της δομής εισόδου και της αντίστοιχης δεντροδιάσπασης. Ο ορισμός των αυτομάτων αποδόμησης βασίζεται στην *ανάπτυξη μιας ειδικής κωδικοποίησης των αναθέσεων πάνω στις δεντροδιασπάσεις*. Πιο συγκεκριμένα, ένα αυτόματο αποδόμησης για έναν MSO τύπο  $\phi$ , καθώς εκτελείται πάνω σε μία δεντρική αποδόμηση της αρχικής δομής, παράγει κωδικοποιημένες τις αναθέσεις που ικανοποιούν τον  $\phi$  στη δεδομένη δομή. Οι αναθέσεις παράγονται μέσω των επιτυχημένων εκτελέσεων του αυτομάτου. Για να κατανοήσουμε διαισθητικά αυτά τα αυτόματα, επισημαίνουμε τη σχέση τους με τα αυτόματα που δέχονται δεντροδιασπάσεις που κωδικοποιούν αναθέσεις: τα τελευταία μπορούν να θεωρηθούν ως μία διαφορετική αναπαράσταση των πρώτων. Αποδεικνύουμε ότι για κάθε MSO τύπο  $\phi$  υπάρχει ένα αυτόματο αποδόμησης που παράγει τις αναθέσεις που ικανοποιούν τον  $\phi$  σε κάθε δομή φραγμένου δεντροπλάτους. Η απόδειξη είναι κατασκευαστική —με την έννοια ότι δίνονται αναλυτικοί ορισμοί των αυτομάτων αποδόμησης— και αποτελεί μία διαφορετική απόδειξη της επέκτασης του θεωρήματος του Courcelle που δίνεται στο [FFG02].

Ένα πλεονέκτημα της προτεινόμενης προσέγγισης, δηλαδή της επίλυσης του προβλήματος αποτίμησης της MSO σε δομές φραγμένου δεντροπλάτους μέσω των αυτομάτων αποδόμησης, είναι ότι δεν απαιτείται κανένας μετασχηματισμός του αρχικού MSO τύπου σε κάποιον νέο MSO τύπο που να ορίζεται σε δέντρα. Στις περιπτώσεις που συμβαίνει κάτι τέτοιο (ανάλογο παράδειγμα είναι η προσέγγιση του [FFG02]), η κατασκευή του αυτομάτου για αυτόν τον νέο τύπο, ο οποίος είναι σημαντικά πολύ πιο σύνθετος από τον αρχικό τύπο, απαιτεί πολύ περισσότερα βήματα και πολύ πιο σύνθετες κατασκευές. Ωστόσο, καθώς στην προσέγγισή μας θεωρούμε όλους του τύπους της MSO, η «έκρηξη καταστάσεων» του τελικού αυτομάτου είναι γενικά και εδώ αναπόφευκτη [FG04]. Έτσι, ο απαιτούμενος χώρος μπορεί να αποβεί σημαντικότερο πρόβλημα στις πρακτικές υλοποιήσεις αυτής της προσέγγισης. Οι Courcelle και Durand αντιμετωπίζουν αυτό το πρόβλημα στο [CD10a] προτείνοντας, μεταξύ άλλων, τμήματα της MSO τα οποία έχουν ενδιαφέρουσα εκφραστική δύναμη και για τα οποία οι κατασκευές των αυτομάτων είναι πρακτικά εφικτές. Αργότερα στο [CD10b], επανεξετάζουν το ίδιο πρόβλημα μέσω μιας μη συστηματικής προσέγγισης. Πιο συγκεκριμένα, θεωρώντας μία συγκεκριμένη ιδιότητα που ορίζεται από έναν MSO τύπο, προτείνουν i) την απευθείας

κατασκευή του τελικού αυτομάτου, και ii) την αναπαράσταση των μεταβάσεων του αυτομάτου με συναρτήσεις. Παρόλο που οι προσεγγίσεις των [CD10a, CD10b] αφορούν σε γράφους φραγμένου δεντροπλάτους, οι ίδιες ιδέες μπορούν κάλλιστα να εφαρμοστούν στο δικό μας πλαίσιο, οδηγώντας σε πιθανές λύσεις για τα αντίστοιχα προβλήματα περιορισμένου χώρου κατά τις υλοποιήσεις.

Παρόλα αυτά, η σημασία μιας γενικής μεθόδου που βασίζεται στα αυτόματα δεν πρέπει σε καμία περίπτωση να υποτιμηθεί. Και αυτό γιατί, πέρα από την ύπαρξη τύπων που δεν προκαλούν έκρηξη καταστάσεων, η γενική μέθοδος παρέχει λύση στις περιπτώσεις εκείνες που δεν υπάρχουν άμεσοι αλγόριθμοι για ένα συγκεκριμένο πρόβλημα. Άλλωστε, η ύπαρξη μιας γενικής λύσης για ένα πρόβλημα μπορεί να λειτουργήσει ως σημείο αναφοράς, θέτοντας τις βασικές κατευθυντήριες γραμμές για την παραγωγή ενός αποδοτικού αλγορίθμου. Για παράδειγμα, μπορούμε να εφαρμόσουμε μεθόδους βελτιστοποίησης κατά τις επαγωγικές κατασκευές των αυτομάτων και να αποκτήσουμε χρήσιμη διαίσθηση για τα χαρακτηριστικά της αντίστοιχης λύσης. Προτείνουμε κάποιες ιδέες βελτιστοποίησης για την περίπτωση του αυτομάτου που επιλύει το πρόβλημα κάλυψης κορυφών (vertex cover). Η προσέγγισή μας συνθέτει πληροφορίες που προκύπτουν τόσο από τον επαγωγικό ορισμό των αυτομάτων όσο και από τη σημασία των λύσεων του συγκεκριμένου προβλήματος. Ωστόσο, μία συστηματική μελέτη των βελτιστοποιήσεων κατά την κατασκευή των αυτομάτων είναι πέρα από τις επιδιώξεις της παρούσας δουλειάς.

Τα αυτόματα αποδόμησης παρέχουν ένα νέο φορμαλισμό αυτομάτων, ο οποίος ορίζει τα ερωτήματα που προσδιορίζουν τα προβλήματα αποτίμησης της MSO που δόθηκαν αναλυτικά παραπάνω. Ωστόσο, εξακολουθούμε να έχουμε ανάγκη από αποδοτικούς αλγορίθμους που να υλοποιούν τη διαδικασία αποτίμησης αυτών των αυτομάτων, που να υπολογίζουν δηλαδή τις απαντήσεις στα ερωτήματα που ορίζονται από αυτά τα αυτόματα.

Στο δεύτερο μέρος αυτού του κεφαλαίου, παρουσιάζουμε βελτιστοποιημένους αλγορίθμους που επιλύουν το πρόβλημα αποτίμησης των αυτομάτων αποδόμησης. Οι αλγόριθμοι αυτοί βασίζονται στην datalog που αποτελεί μία γλώσσα ερωτημάτων για τις επαγωγικές βάσεις δεδομένων. Αναλυτικότερα, πραγματοποιούμε μία συστηματική γενίκευση της προσέγγισης του προηγούμενου κεφαλαίου –για την περίπτωση των δέντρων– σε προβλήματα αποτίμησης που ορίζονται μέσω MSO τύπων πάνω σε δομές φραγμένου δεντροπλάτους. Πιο συγκεκριμένα, εκφράζουμε στην datalog το πρόβλημα αποτίμησης για τα αυτόματα αποδόμησης, και στη συνέχεια χρησιμοποιούμε μεθόδους βελτιστοποίησης των βάσεων δεδομένων για να αποτιμήσουμε τα αντίστοιχα datalog ερωτήματα. Οι αλγόριθμοι που προκύπτουν αποτελούν αποδοτικές διαδικασίες επίλυσης για το πρόβλημα αποτίμησης των αυτομάτων αποδόμησης, και κατά συνέπεια και για τα αρχικά προβλήματα αποτίμησης της MSO που θεωρούμε. Αξίζει να σημειωθεί ότι, επειδή η κατασκευή ενός αυτομάτου μπορεί να εκφραστεί πλήρως στην datalog, μπορούμε δεδομένου ενός τύπου της MSO να ορίσουμε απευθείας datalog ερωτήματα που υπολογίζουν τις λύσεις των προβλημάτων αποτίμησης που θεωρούμε. Εκτός από αυτή την αλγοριθμική χαρακτηρισία σημασία του συγκεκριμένου αποτελέσματος, που είναι άμεση συνέπεια της λειτουργικής σημασιολογίας (operational semantics) που διαθέτει η datalog, μέσω της προσέγγισής μας αποδεικνύουμε ενδιαφέροντα αποτελέσματα εκφρασιμότητας για την datalog.

Πιο συγκεκριμένα, αποδεικνύουμε ότι κάθε μονομελές (αντίστοιχα  $k$ -μελές) MSO-ορίσιμο ερώτημα σε δομές φραγμένου δεντροπλάτους είναι ορίσιμο στη μοναδιαία datalog (αντίστοιχα στην  $(k + 1)$ -datalog). Το αποτέλεσμα που αφορά στην ορισιμότητα των μονομελών MSO-ορίσιμων ερωτημάτων στη μοναδιαία datalog έχει ήδη αποδειχθεί στο [GPW10] μέσω τη θεωρίας σύνθεσης των type. Η γενίκευση του αποτελέσματος αυτού για την περίπτωση των  $k$ -μελών ερωτημάτων, την οποία αποδεικνύουμε εδώ μέσω της θεωρίας αυτομάτων, είναι μη τετριμμένη. Ενδιαφέροντα αποτελέσματα εκφρασιμότητας αποδεικνύονται και για τα προβλήματα βελτιστοποίησης που μελετάμε. Στις αντίστοιχες περιπτώσεις, η επίλυση των προβλημάτων πραγματοποιείται μέσω της επέκτασης της datalog με συναθροιστικές συναρτήσεις (aggregate functions)

Η προσέγγιση που παρέχουμε αποτελεί μία πλήρη και γενική αλγοριθμική διαδικασία για την επίλυση οποιουδήποτε προβλήματος μπορεί να μοντελοποιηθεί ως πρόβλημα αποτίμησης για την MSO σε δομές φραγμένου δεντροπλάτους. Επιλέγουμε να δώσουμε την πλήρη παρουσίαση της προσέγγισής μας για την επίλυση των εξής προβλημάτων που σχετίζονται με τον υπολογισμό καλύμματος κορυφών (vertex cover) σε γράφους: VERTEX COVER, MINIMUM VERTEX COVER, MINIMUM VERTEX COVER CARDINALITY και  $k$ -VERTEX COVER. Πιο συγκεκριμένα, δίνουμε αρχικά τον ορισμό του αυτομάτου αποδόμησης για έναν MSO τύπο  $\phi_{vc}(X)$  ο οποίος εκφράζει ότι «το σύνολο  $X$  αποτελεί κάλυμμα κορυφών». Στη συνέχεια, χρησιμοποιούμε το αυτόματο αυτό για να επιλύσουμε όλα τα προαναφερθέντα προβλήματα που σχετίζονται με το πρόβλημα κάλυψης κορυφών.



## 2.

---

# Προβλήματα Αποτίμησης για τη Μοναδιαία Δευτεροβάθμια Λογική σε Πεπερασμένα Χρωματισμένα Δέντρα

Στο κεφάλαιο αυτό επιλύουμε το πρόβλημα της αποτίμησης για τη μοναδιαία δευτεροβάθμια λογική (MSO) σε χρωματισμένα δυαδικά δέντρα, με τη βοήθεια δύο αποτελεσματικών μεθόδων της θεωρίας βάσεων δεδομένων. Η βάση της προσέγγισής μας είναι η κλασική ισοδυναμία MSO–αυτομάτων για δέντρα.

Αρχικά, παρουσιάζουμε μια αναγωγή του προβλήματος της αποτίμησης για την MSO σε ένα πρόβλημα αποτίμησης ενός ερωτήματος της σχεσιακής άλγεβρας. Πιο συγκεκριμένα, ορίζουμε ένα ακυκλικό συζευκτικό ερώτημα  $\mathcal{Q}$  που υπολογίζει τις αναθέσεις που ικανοποιούν ένα δεδομένο MSO τύπο  $\phi$  σε ένα δεδομένο δέντρο, και αποτιμούμε το  $\mathcal{Q}$  με τον αλγόριθμο του Yannakakis [Yan81]. Μέσω της προσέγγισής μας, αποδεικνύουμε τελικά ότι τα βήματα υπολογισμού που προκύπτουν από την εφαρμογή του αλγορίθμου του Yannakakis για την αποτίμηση του  $\mathcal{Q}$  αντιστοιχούν πλήρως στον αλγόριθμο αυτομάτων του [FFG02]. Παράλληλα, αποδεικνύεται η πολυπλοκότητα του αλγορίθμου που στην περίπτωσή μας είναι βελτιωμένη σε σχέση με τη γενική πολυπλοκότητα του αλγορίθμου του Yannakakis.

Στη συνέχεια, παρουσιάζουμε μια δεύτερη αναγωγή για το ίδιο πρόβλημα, η οποία όμως βασίζεται στην datalog. Πιο συγκεκριμένα, κατασκευάζουμε ένα datalog πρόγραμμα που παράγει τις αναθέσεις που ικανοποιούν ένα δεδομένο MSO τύπο  $\phi$  σε ένα δεδομένο δέντρο, και στη συνέχεια δίνουμε δυο βελτιστοποιημένες εκδοχές αυτού του προγράμματος μέσω της μεθόδου της αναγραφής που βασίζεται στην επίλυση και αναφέρεται συχνά στη βιβλιογραφία ως «magic sets» [BMSU86]. Ο πυρήνας του δεύτερου βελτιστοποιημένου προγράμματος χρησιμοποιείται ώστε να εκφράσουμε τη λύση και στη μοναδιαία datalog. Έτσι, αποδεικνύουμε ότι το πρόβλημα της αποτίμησης για την MSO σε χρωματισμένα δέντρα είναι ορίσιμο τόσο στην datalog όσο και στη μοναδιαία datalog. Τα αντίστοιχα αποτελέσματα εκφρασιμότητας επεκτεί-

νουν το αποτέλεσμα του [GK04] ότι τα μονομελή MSO-ορίσιμα ερωτήματα είναι ορίσιμα στη μοναδιαία datalog. Για όλα τα προγράμματα πραγματοποιείται ανάλυση πολυπλοκότητας.

Για λόγους απλοποίησης, παρουσιάζουμε όλα τα αποτελέσματα για την περίπτωση των πεπερασμένων δυαδικών δέντρων. Το γεγονός αυτό δε συνεπάγεται απώλεια γενικότητας καθώς όλα τα αποτελέσματα εφαρμόζονται κατά τον ίδιο τρόπο και στην περίπτωση των δέντρων που έχουν το πολύ  $k$  παιδιά.

## 2.1 Βασικές Έννοιες

Στην ενότητα αυτή δίνουμε τους τυπικούς ορισμούς που είναι απαραίτητοι για την παρουσίαση των αποτελεσμάτων αυτού του κεφαλαίου.

**Δομές & δέντρα, MSO.** Μία (σχεσιακή) υπογραφή  $\tau$  είναι ένα πεπερασμένο σύνολο από σύμβολα κατηγορημάτων  $R_1, \dots, R_n$ . Σε κάθε σύμβολο κατηγορήματος  $R_i$ ,  $i \in \{1, \dots, n\}$ , αντιστοιχεί ένας φυσικός αριθμός  $a_i \geq 0$  που ονομάζεται βαθμός του. Μία  $\tau$ -δομή  $\mathcal{A}$  είναι ένα ζευγάρι

$$(A, (R_i^{\mathcal{A}})_{i \in \{1, \dots, n\}})$$

όπου το  $A$  είναι ένα μη κενό σύνολο που ονομάζεται *σύμπαν* της  $\mathcal{A}$ , και κάθε  $R_i^{\mathcal{A}}$  είναι μία σχέση με  $a_i$  μεταβλητές στο  $A$ . Μία δομή  $\mathcal{A}$  είναι πεπερασμένη εάν το  $A$  είναι ένα πεπερασμένο σύνολο. Στο κεφάλαιο αυτό μελετάμε δομές που αναπαριστούν πεπερασμένα χρωματισμένα πλήρη δυαδικά δέντρα, δηλαδή χρωματισμένα δέντρα όπου κάθε εσωτερικός κόμβος έχει ακριβώς δύο παιδιά. Ας υποθέσουμε ότι το  $\Gamma$  είναι ένα πεπερασμένο σύνολο συμβόλων που ονομάζεται το *αλφάβητο χρωμάτων* και έστω ότι το  $\tau_\Gamma$  συμβολίζει την υπογραφή

$$\tau_\Gamma = \{S_1, S_2, (P_\gamma)_{\gamma \in \Gamma}\}$$

όπου τα  $S_1, S_2$  είναι διμελή σύμβολα κατηγορημάτων και τα  $(P_\gamma)_{\gamma \in \Gamma}$  μονομελή σύμβολα κατηγορημάτων. Ένα  $\Gamma$ -χρωματισμένο δέντρο (ή  $\Gamma$ -δέντρο)  $\mathcal{T}$  είναι μία πεπερασμένη σχεσιακή  $\tau_\Gamma$ -δομή  $(T, S_1^{\mathcal{T}}, S_2^{\mathcal{T}}, (P_\gamma^{\mathcal{T}})_{\gamma \in \Gamma})$  τέτοια ώστε  $(n, n_i) \in S_i^{\mathcal{T}}$  αν  $n_i$  είναι το  $i$ -οστό παιδί του  $n$ ,  $i = 1, 2$ , και  $n \in P_\gamma^{\mathcal{T}}$  αν το  $\gamma$  είναι το χρώμα του  $n$ . Κάθε στοιχείο του  $T$  ονομάζεται κόμβος του δέντρου  $\mathcal{T}$ . Μπορούμε ισοδύναμα να αναπαραστήσουμε ένα δέντρο  $\mathcal{T}$  ως ένα ζευγάρι

$$(t, c)$$

όπου το  $t = (T, s_1, s_2)$  είναι το *υποκείμενο δέντρο* του  $\mathcal{T}$  και τα  $s_1, s_2$  είναι οι συναρτήσεις αριστερού και δεξιού παιδιού αντίστοιχα, ενώ η  $c : T \rightarrow \Gamma$  είναι η *συνάρτηση χρωματισμού* για το  $t$  τέτοια ώστε  $c(n) = \gamma$  αν  $n \in P_\gamma^{\mathcal{T}}$ . Ένα *υποδέντρο*  $\mathcal{T}'$  του  $\mathcal{T}$  είναι ένας συνδεδεμένος υπογράφος του  $\mathcal{T}$  με σύνολο κόμβων  $T' \subseteq T$ :

$$\mathcal{T}' = (T', S_1^{\mathcal{T}} \cap T'^2, S_2^{\mathcal{T}} \cap T'^2, (P_\gamma^{\mathcal{T}} \cap T')_{\gamma \in \Gamma}).$$

Συμβολίζουμε με  $\mathcal{T}_n$  το υποδέντρο του  $\mathcal{T}$  με ρίζα το  $n$  και σύνολο κόμβων  $T_n$  που περιλαμβάνει από όλα τα φύλλα που είναι προσβάσιμα από τον  $n$ . Τέτοια υποδέντρα ονομάζονται



πλήρη. Η προδιατεταγμένη διάσχιση (*preorder traversal*) ενός δέντρου συναντά τους κόμβους σύμφωνα με τον αλγόριθμο της διερεύνησης κατά βάθος (*depth-first search*) που στην περίπτωση των δυαδικών δέντρων ορίζεται αναδρομικά ως εξής: i) επισκέψου αρχικά τη ρίζα, ii) πραγματοποιήσε μία προδιατεταγμένη διάσχιση στο αριστερό υποδέντρο, iii) πραγματοποιήσε μία προδιατεταγμένη διάσχιση στο δεξιό υποδέντρο. Σε όλο το κεφάλαιο, κάθε φορά που αναφερόμαστε σε μία διάταξη των κόμβων του συνόλου  $T$ , θεωρούμε ότι αυτή αντιστοιχεί στην προδιατεταγμένη διάσχιση.

Η μοναδιαία δευτεροβάθμια λογική (*Monadic Second-Order logic* ή MSO για συντομία) αποτελεί επέκταση της πρωτοβάθμιας λογικής (*First-Order logic* ή FO για συντομία) η οποία επιτρέπει την ποσοδείκτηση πάνω σε σύνολα. Στο κεφάλαιο αυτό μελετάμε την  $MSO[\tau_T]$  που είναι το τμήμα εκείνο της MSO που αντιστοιχεί στο σύνολο όλων των MSO τύπων που γράφονται με σύμβολα κατηγορημάτων από το  $\tau_T$  και δεν εμφανίζουν πρωτοβάθμιες μεταβλητές. Η απαλοιφή των πρωτοβάθμιων μεταβλητών είναι κοινή πρακτική (βλέπε για παράδειγμα την  $MSO_0$  στο [Tho97b]) η οποία υιοθετείται για λόγους ομοιογένειας και δεν συνεπάγεται απώλεια της γενικότητας. Πράγματι, η  $MSO[\tau_T]$  είναι ισοδύναμη ως προς τα μοντέλα με την κλάση των MSO τύπων που έχουν την ίδια υπογραφή και γράφονται τόσο με δευτεροβάθμιες όσο και με πρωτοβάθμιες μεταβλητές. Οι τελευταίοι μετατρέπονται στους πρώτους αν κάθε πρωτοβάθμια μεταβλητή  $x$  αντικατασταθεί από μία δευτεροβάθμια μεταβλητή  $X$  έτσι ώστε αν η  $x$  δηλώνει το στοιχείο  $a$  του σύμπαντος, τότε η  $X$  να δηλώνει το μονοσύνολο  $\{a\}$ . Συνεπώς, οι ατομικοί τύποι που συνήθως γράφονται ως  $Z(x)$  (ερμηνεύεται ως «το στοιχείο  $x$  ανήκει στο σύνολο  $Z$ »),  $S_1(x, y)$  (ερμηνεύεται ως «το στοιχείο  $y$  είναι το πρώτο παιδί του στοιχείου  $x$ »),  $S_2(x, y)$  (ανάλογα για το δεύτερο παιδί),  $P_\gamma(x)$  (ερμηνεύεται ως «το χρώμα του στοιχείου  $x$  είναι  $\gamma$ ») γράφονται τώρα αντίστοιχα ως

$$E(X, Z), S_1(X, Y), S_2(X, Y), P_\gamma(X)$$

όπου οι μεταβλητές  $X, Y$  ερμηνεύονται ως μονοσύνολα. Οι μη ατομικοί τύποι έχουν τη μορφή

$$\exists X \phi, \neg \phi, \phi \wedge \psi$$

όπου  $\phi$  και  $\psi$  είναι  $MSO[\tau_T]$  τύποι.

Σημειώνουμε ότι ο ατομικός τύπος  $x = y$  αναπαρίστανται από τον  $MSO[\tau_T]$  τύπο

$$E(Y, X) \wedge E(X, Y)$$

και ο τύπος  $\exists y(S_1(x, y) \wedge Z(y) \wedge P_\gamma(x))$  αναγράφεται στον

$$\exists Y(S_1(X, Y) \wedge E(Y, Z) \wedge P_\gamma(X)).$$

Ονομάζουμε  $MSO[\tau_T]$  τύπους χωρίς άρνηση, τους τύπους εκείνους στους οποίους ο σύνδεσμος  $\neg$  δεν εμφανίζεται καθόλου ή εμφανίζεται μόνο μπροστά από ατομικούς τύπους. Το σύνολο των τύπων χωρίς άρνηση, που αποτελεί την MSO χωρίς άρνηση, είναι το τμήμα της  $MSO[\tau_T]$  όπου οι μη ατομικοί τύποι έχουν είτε τη μορφή  $\neg a$ ,  $a$  ατομικός τύπος, ή τη μορφή  $\exists X \phi$  ή  $\phi \wedge \psi$  για  $\phi, \psi$  MSO τύπους χωρίς άρνηση.

**Η σύνδεση MSO & αυτομάτων για δέντρα.** Ένα μη ντετερμινιστικό *bottom-up* αυτόματο για  $\Gamma$ -χρωματισμένα δέντρα (ή για συντομία  $\Gamma$ -αυτόματο)  $\mathcal{A}$  είναι μία πλειάδα

$$(\Gamma, Q, \Delta_0, \Delta, F)$$

όπου το  $\Gamma$  είναι ένα πεπερασμένο αλφάβητο χρωμάτων, το  $Q$  είναι ένα πεπερασμένο σύνολο καταστάσεων, το  $F \subseteq Q$  είναι το σύνολο των τελικών καταστάσεων και οι  $\Delta_0, \Delta$  είναι οι σχέσεις μετάβασης:  $\Delta_0 \subseteq \Gamma \times Q$  και  $\Delta \subseteq Q \times Q \times \Gamma \times Q$ . Το αυτόματο  $\mathcal{A}$  είναι ντετερμινιστικό όταν οι σχέσεις  $\Delta_0, \Delta$  ορίζουν απεικονίσεις  $\delta_0 : \Gamma \rightarrow Q, \delta : Q \times Q \times \Gamma \rightarrow Q$  αντίστοιχα.

Η εκτέλεση (run)  $\rho$  ενός (είτε ντετερμινιστικού είτε μη)  $\Gamma$ -αυτομάτου  $\mathcal{A}$  σε ένα  $\Gamma$ -δέντρο  $\mathcal{T}$ , είναι μία απεικόνιση, η οποία αποδίδει καταστάσεις στους κόμβους του δέντρου έτσι ώστε:

- (i) αν το  $n$  είναι φύλλο χρώματος  $a$ , τότε  $\rho(n) = q$  εάν υπάρχει μία μετάβαση  $(a, q) \in \Delta_0$ ,
- (ii) αν το  $n$  είναι ένας εσωτερικός κόμβος χρώματος  $a$  με παιδιά τα  $n_1, n_2$ , η τιμή της  $\rho(n)$  είναι τέτοια ώστε  $(\rho(n_1), \rho(n_2), a, \rho(n)) \in \Delta$ .

Μία εκτέλεση είναι *επιτυχημένη* εάν απεικονίζει τη ρίζα σε μία τελική κατάσταση. Ένα  $\Gamma$ -δέντρο  $\mathcal{T}$  αναγνωρίζεται από ένα  $\Gamma$ -αυτόματο  $\mathcal{A}$  εάν υπάρχει μία επιτυχημένη εκτέλεση του  $\mathcal{A}$  στο  $\mathcal{T}$ . Μία κλάση  $\mathcal{C}$   $\Gamma$ -δέντρων είναι *αναγνωρίσιμη* εάν υπάρχει ένα αυτόματο τέτοιο ώστε να αναγνωρίζει ακριβώς όλα τα δέντρα της  $\mathcal{C}$  και κανένα άλλο.

Θυμίζουμε ότι μία κλάση  $\mathcal{C}$   $\Gamma$ -δέντρων είναι  $\text{MSO}[\tau_\Gamma]$ -ορίσιμη εάν υπάρχει μία πρόταση  $\phi$  της  $\text{MSO}[\tau_\Gamma]$  τέτοια ώστε  $\mathcal{T} \in \mathcal{C}$  ανν  $\mathcal{T} \models \phi$ . Το θεώρημα που ακολουθεί διατυπώνει την κλασσική ισοδυναμία αυτομάτων και MSO στην περίπτωση των δέντρων.

**Θεώρημα 2.1.1 (Doner [Don70], Thatcher & Wright [TW68]).** Μία κλάση  $\Gamma$ -δέντρων είναι αναγνωρίσιμη αν και μόνο αν είναι  $\text{MSO}[\tau_\Gamma]$ -ορίσιμη.

**Αποτίμηση MSO ερωτημάτων σε πεπερασμένα δέντρα: Το πρόβλημα της αποτίμησης της MSO & Το πρόβλημα ελέγχου μοντέλου της MSO.** Κάθε  $\text{MSO}[\tau_\Gamma]$  τύπος  $\phi(X_1, \dots, X_k)$  ορίζει μια απεικόνιση, η οποία επίσης συμβολίζεται με  $\phi$ , πάνω στην κλάση  $\mathcal{C}_\mathcal{T}$  των πεπερασμένων χρωματισμένων δέντρων: για κάθε δέντρο  $\mathcal{T}$ ,

$$\phi(\mathcal{T}) = \{(B_1, \dots, B_k) \mid \mathcal{T} \models \phi(B_1, \dots, B_k)\} \subseteq (\mathcal{P}(T))^k.$$

Η κλάση όλων των απεικονίσεων  $q$  πάνω στην  $\mathcal{C}_\mathcal{T}$  που ορίζονται από έναν MSO τύπο, δηλαδή εκείνων των απεικονίσεων για τις οποίες υπάρχει ένας MSO τύπος  $\phi(X_1, \dots, X_k)$  τέτοιος ώστε

$$q(\mathcal{T}) = \phi(\mathcal{T}) \text{ για κάθε δέντρο } \mathcal{T} \text{ της κλάσης } \mathcal{C}_\mathcal{T}$$

ονομάζεται η κλάση των  $k$ -μελών MSO-ορίσμων ερωτημάτων (ή απλά MSO ερωτημάτων) σε δέντρα. Το πρόβλημα της αποτίμησης της MSO (MSO evaluation problem) σε πεπερασμένα δέντρα ορίζεται ως το πρόβλημα υπολογισμού του συνόλου  $\phi(\mathcal{T})$  δεδομένων ενός τύπου  $\phi$  και ενός δέντρου  $\mathcal{T}$ . Προφανώς, η επίλυση αυτού του προβλήματος ισοδυναμεί με την επίλυση του προβλήματος αποτίμησης MSO ερωτημάτων σε δέντρα.

Εκτός από το πρόβλημα αποτίμησης της MSO, μελετάμε ανεξάρτητα τις εξής ειδικές περιπτώσεις:

1. το πρόβλημα ελέγχου μοντέλου της MSO (MSO model checking problem), δηλαδή το πρόβλημα απόφασης που ορίζεται μέσω της ερώτησης για το αν ισχύει ότι  $\mathcal{T} \models \phi$  δεδομένης μιας MSO πρότασης  $\phi$  και ενός δέντρου  $\mathcal{T}$ ,
2. το πρόβλημα αποτίμησης της MSO για τύπους  $\phi(X)$  όπου η  $X$  ερμηνεύεται ως μονοσύνολο (δηλαδή για εκείνους τους τύπους που αντιστοιχούν σε MSO τύπους  $\phi(x)$ ,  $x$  πρωτοβάθμια μεταβλητή) και ορίζουν την απεικόνιση

$$\phi(\mathcal{T}) = \{n \mid \mathcal{T} \models \phi(\{n\})\} \subseteq T,$$

3. το πρόβλημα αποτίμησης της MSO για τύπους  $\phi(X_1, \dots, X_k)$  όπου κάθε μεταβλητή  $X_i$  ερμηνεύεται ως μονοσύνολο (δηλαδή για τύπους που αντιστοιχούν σε MSO τύπους  $\phi(x_1, \dots, x_k)$ ,  $x_i$  πρωτοβάθμια μεταβλητή) και ορίζουν την απεικόνιση

$$\phi(\mathcal{T}) = \{(n_1, \dots, n_k) \mid \mathcal{T} \models \phi(\{n_1\}, \dots, \{n_k\})\} \subseteq T^k.$$

Οι απεικονίσεις πάνω στην κλάση  $\mathcal{C}_{\mathcal{T}}$  που ορίζονται από MSO προτάσεις (αντίστοιχα από MSO τύπους  $\phi(x)$  όπως στην περίπτωση (2) παραπάνω, και αντίστοιχα από τύπους  $\phi(x_1, \dots, x_k)$  όπως στην περίπτωση (3) παραπάνω) ονομάζονται στη βιβλιογραφία *boolean* (αντίστοιχα *μονομελή*, και αντίστοιχα *k-μελή επιλογής κόμβων* (node selection) MSO-ορίσματα ερωτήματα σε δέντρα (ή MSO ερωτήματα σε δέντρα).

## 2.2 Επαναπροσδιορισμός της σχέσης MSO–Αυτομάτων σε δέντρα

Στην ενότητα αυτή εισάγουμε τα αυτόματα αναθέσεων (assignment automata) και επαναπροσδιορίζουμε μέσω αυτών τη γνωστή σχέση MSO–αυτομάτων στα πεπερασμένα δέντρα. Με αυτό τον τρόπο, παρέχουμε μια αναγωγή του προβλήματος αποτίμησης της MSO πάνω σε δέντρα, σε ένα πρόβλημα αποτίμησης αυτομάτων αναθέσεων. Η εν λόγω αναγωγή αποτελεί τη θεωρητική βάση των *database-oriented* λύσεών μας. Οι τελευταίες, επιλύουν προβλήματα αποτίμησης της MSO επιλύοντας στην ουσία ισοδύναμα προβλήματα αποτίμησης αυτομάτων.

### 2.2.1 Κωδικοποιήσεις αναθέσεων & αυτόματα αναθέσεων

Τα αυτόματα αναθέσεων είναι αυτόματα τα οποία, όταν εκτελούνται σε ένα δέντρο  $\mathcal{T}$ , παράγουν αναθέσεις κωδικοποιημένες με τη μορφή κατάλληλων απεικονίσεων από το σύμπαν του  $\mathcal{T}$  σε διανύσματα πάνω στο  $\{0, 1\}$ . Πιο συγκεκριμένα, ας υποθέσουμε ότι το  $A$  είναι ένα πεπερασμένο σύνολο. Ονομάζουμε  $k$ -ανάθεση του  $A$  κάθε  $k$ -πλειάδα  $\bar{B} = (B_1, \dots, B_k)$  από υποσύνολα του  $A$ . Κάθε τέτοια  $k$ -πλειάδα  $\bar{B}$  μπορεί να κωδικοποιηθεί ως μία ολική συνάρτηση  $\varepsilon$  με πεδίο ορισμού το  $A$  και σύνολο τιμών το  $\{0, 1\}^k$  που ορίζεται ως εξής: για κάθε  $n \in A$ , η τιμή της  $\varepsilon$  στο  $n$  είναι ένα  $k$ -διάνυσμα από  $0, 1$  έτσι ώστε η εμφάνιση του  $1$  (αντίστοιχα του  $0$ ) στη θέση  $j$  του  $\varepsilon(n)$  δηλώνει ότι το  $n$  περιέχεται (αντίστοιχα δεν περιέχεται) στο  $B_j$ .

Αξίζει να σημειωθεί ότι κάθε ολική απεικόνιση  $\varepsilon$  με πεδίο ορισμού ένα πεπερασμένο σύνολο  $A$  μεγέθους  $m$  έχει μία φυσική αναπαράσταση ως διάνυσμα με τη μορφή ενός  $(k, m)$ -διανύσματος ανάθεσης, δηλαδή ως ένα στοιχείο του  $(\{0, 1\}^k)^m$ , με τον εξής τρόπο: υποθέτοντας μία ολική διάταξη  $o$  στο σύνολο  $A$ , το  $A$  μπορεί να ειδωθεί ως ένα  $m$ -διάνυσμα πάνω στα στοιχεία του, διατεταγμένα σύμφωνα με την  $o$ . Αρκεί σε αυτό το διάνυσμα να αντικατασταθεί κάθε στοιχείο του  $n$  με την τιμή  $\varepsilon(n)$ . Συμβολίζουμε με  $\varepsilon_o$  τη διανυσματική αναπαράσταση του  $\varepsilon$  ως προς τη διάταξη  $o$ . Όταν η  $o$  είναι ξεκάθαρη από τα συμφραζόμενα την παραλείπουμε και χρησιμοποιούμε τον ίδιο συμβολισμό  $\varepsilon$  τόσο για την απεικόνιση όσο και για τη διανυσματική της αναπαράσταση.

**Ορισμός 2.2.1.** Θεωρούμε ότι το  $T$  είναι ένα πεπερασμένο σύνολο και έστω ότι

$$\bar{B} = (B_1, \dots, B_k) \in (\mathcal{P}(T))^k.$$

Η απεικόνιση ανάθεσης που κωδικοποιεί την  $\bar{B}$  συμβολίζεται με  $\varepsilon_{\bar{B}}$  και ορίζεται ως εξής:

$$\varepsilon_{\bar{B}} : T \rightarrow \{0, 1\}^k$$

τέτοια ώστε για κάθε  $n \in A$ , έχουμε  $\varepsilon_{\bar{B}}(n) = (e_n^1, \dots, e_n^k)$  όπου

$$e_n^i = 1 \quad \text{ανν} \quad n \in B_i, \quad 1 \leq i \leq k.$$

Έστω ότι το  $\varepsilon$  αποτελεί την αναπαράσταση ως  $(k, m)$ -διάνυσμα ανάθεσης της απεικόνισης  $\varepsilon_{\bar{B}}$ . Το  $\varepsilon$  ονομάζεται το διάνυσμα ανάθεσης που κωδικοποιεί την  $\bar{B}$  και συμβολίζεται με  $\varepsilon_{\bar{B}}$ .

Η παραπάνω διαδικασία κωδικοποίησης ορίζει την εξής συνάρτηση κωδικοποίησης  $\text{enc}_k : \bar{B} \mapsto \varepsilon_{\bar{B}}$ . Ορίζουμε επίσης την συνάρτηση αποκωδικοποίησης  $\text{dec}_k$  ως την αντίστροφη  $\text{enc}_k^{-1}$  της συνάρτησης κωδικοποίησης. Αν  $\text{dec}_k(\varepsilon) = \bar{B}$ , τότε η  $\bar{B}$  ονομάζεται η  $k$ -ανάθεση που κωδικοποιείται από το  $\varepsilon$  και συμβολίζεται με  $\bar{B}_\varepsilon$ .

Ονομάζουμε κωδικοποιημένη εκδοχή του προβλήματος αποτίμησης της MSO σε πεπερασμένα δέντρα το πρόβλημα του υπολογισμού, με δεδομένα έναν MSO τύπο  $\phi$  και ένα δέντρο  $\mathcal{T}$ , του συνόλου

$$\phi_{\text{enc}}(\mathcal{T}) = \{\varepsilon_{\bar{B}} \mid \bar{B} \in \phi(\mathcal{T})\}$$

των διανυσμάτων που κωδικοποιούν τις αναθέσεις που ικανοποιούν τον  $\phi$ . Τα στοιχεία του  $\phi_{\text{enc}}(\mathcal{T})$  ονομάζονται κωδικοποιημένες αναθέσεις που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$ .

**Παράδειγμα 2.2.1.** Ας θεωρήσουμε το σύνολο  $S = \{n_1, n_2, n_3\}$  και τη διάταξη  $o = (n_1, n_2, n_3)$  (δηλαδή  $n_1 < n_2 < n_3$ ). Η ανάθεση  $\bar{B}_1 = (\{n_2\}, \{n_1, n_3\})$  πάνω στο  $S$  κωδικοποιείται από το διάνυσμα  $\varepsilon_1 = ((0, 1), (1, 0), (0, 1))$ , ενώ το διάνυσμα  $\varepsilon_2 = ((0, 0), (1, 0), (0, 1))$  κωδικοποιεί την ανάθεση  $\bar{B}_2 = (\{n_2\}, \{n_3\})$ .

Είμαστε λοιπόν έτοιμοι να ορίσουμε τα αυτόματα  $k$ -αναθέσεων τα οποία καθώς εκτελούνται σε ένα δέντρο  $\mathcal{T}$  με σύνολο κόμβων  $T$ , υπολογίζουν  $k$ -αναθέσεις του  $T$ . Αυτές οι αναθέσεις, οι οποίες είναι κωδικοποιημένες με τη μορφή απεικονίσεων, αποτελούν τμήμα των επιτυχημένων εκτελέσεων των αυτομάτων αναθέσεων.

**Ορισμός 2.2.2.** Ένα  $\Gamma$ -αυτόματο  $k$ -αναθέσεων  $\mathcal{A}$ ,  $k > 0$ , είναι ένα ζευγάρι

$$(A, \text{dec}_k)$$

όπου το

$$A = (\Gamma, Q \times \{0, 1\}^k, \Delta_0, \Delta, F \times \{0, 1\}^k)$$

είναι ένα μη-ντετερμινιστικό bottom-up αυτόματο για  $\Gamma$ -δέντρα: το  $Q$  είναι ένα πεπερασμένο σύνολο, το  $F \subseteq Q$ , το  $Q_k = Q \times \{0, 1\}^k$  αποτελεί το σύνολο καταστάσεων του  $\mathcal{A}$ . Κάθε κατάσταση είναι μία  $(k + 1)$ -πλειάδα  $(q, e_1, \dots, e_k)$ . Το  $F_k = F \times \{0, 1\}^k$  είναι το σύνολο των τελικών καταστάσεων του  $\mathcal{A}$ . Η  $\Delta_0 \subseteq \Gamma \times Q_k$  είναι η αρχική συνάρτηση μετάβασης και η  $\Delta \subseteq Q_k \times Q_k \times \Gamma \times Q_k$  είναι η συνάρτηση μετάβασης. Στη συνέχεια θα χρησιμοποιούμε συχνά τα  $q\bar{e}$  και  $q_j\bar{e}_j$  ως συντομεύσεις για τα  $(q, e_1, \dots, e_k)$  και  $(q_j, e_{j1}, \dots, e_{jk})$  αντίστοιχα, όπου τα  $\bar{e} = (e_1, \dots, e_k)$ ,  $\bar{e}_j = (e_{j1}, \dots, e_{jk})$  είναι στοιχεία του  $\{0, 1\}^k$ . Το δεύτερο σκέλος

$$\text{dec}_k$$

είναι η συνάρτηση αποκωδικοποίησης που ορίστηκε παραπάνω στον Ορισμό 2.2.1

Μία εκτέλεση  $\rho$  ενός  $\Gamma$ -αυτομάτου  $k$ -αναθέσεων  $\mathcal{A}$  σε ένα  $\Gamma$ -δέντρο ορίζεται με τον ίδιο τρόπο όπως η εκτέλεση ενός κανονικού μη-ντετερμινιστικού αυτομάτου για δέντρα δηλαδή είναι μία απεικόνιση που αντιστοιχεί καταστάσεις σε κόμβους έτσι ώστε i) αν το  $n$  είναι ένα φύλλο χρώματος  $a$ , τότε  $\rho(n) = q\bar{e}$  εάν υπάρχει μία μετάβαση  $(a, q\bar{e}) \in \Delta_0$ , και ii) αν το  $n$  είναι ένας εσωτερικός κόμβος χρώματος  $a$ , με παιδιά τα  $n_1, n_2$ , τότε η τιμή  $\rho(n)$  είναι τέτοια ώστε  $(\rho(n_1), \rho(n_2), a, \rho(n)) \in \Delta$ . Μία εκτέλεση ονομάζεται επιτυχημένη εάν αντιστοιχεί τη ρίζα σε μία τελική κατάσταση. Έστω ότι το  $\mathcal{T}'$  είναι ένα πλήρες υποδέντρο του  $\Gamma$ -δέντρου  $\mathcal{T}$ . Ονομάζουμε  $\mathcal{T}'$ -υποεκτέλεση (ή υποεκτέλεση στο  $\mathcal{T}'$ ) του  $\mathcal{A}$  τον περιορισμό στο (σύνολο κόμβων του)  $\mathcal{T}'$  της εκτέλεσης του  $\mathcal{A}$  στο  $\mathcal{T}$  – οι  $\mathcal{T}'$ -υποεκτελέσεις του  $\mathcal{A}$  αντιστοιχούν πλήρως στις εκτελέσεις του  $\mathcal{A}$  στο  $\mathcal{T}'$ .

Θεωρούμε την εκτέλεση

$$\rho : T \rightarrow Q \times \{0, 1\}^k.$$

Είναι ιδιαίτερα βολικό στη συνέχεια να σκεφτόμαστε την  $\rho$  ως τη συνένωση (concatenation)

$$\varsigma; \varepsilon$$

δύο τμημάτων, του μέρους των καταστάσεων  $\varsigma : T \rightarrow Q$  και του μέρους της ανάθεσης  $\varepsilon : T \rightarrow \{0, 1\}^k$ , που ορίζονται ως εξής:

$$\zeta(n) = q \text{ και } \varepsilon(n) = \bar{e} \text{ ανν } \rho(n) = (q, \bar{e}) \text{ για κάθε } n.$$

Ονομάζουμε *διάνυσμα καταστάσεων* και *διάνυσμα ανάθεσης* του  $\rho$  τη διανυσματική αναπαράσταση, ως προς την προδιατεταγμένη διάσχιση (preorder traversal) του δεδομένου δέντρου, του μέρους των καταστάσεων και του μέρους της ανάθεσης της  $\rho$  αντίστοιχα.

Εάν η  $\rho = \zeta; \varepsilon$  είναι μία επιτυχημένη εκτέλεση του  $\mathcal{A}$  στο  $\mathcal{T}$ , θα λέμε ότι το  $\mathcal{A}$  υπολογίζει την ανάθεση  $\text{dec}_k(\varepsilon)$  δηλαδή την ανάθεση  $\overline{B}_\varepsilon$  που κωδικοποιεί το  $\varepsilon$ . Συνεπώς το αυτόματο  $\mathcal{A}$  ορίζει μία απεικόνιση πάνω στα δέντρα που αντιστοιχεί ένα δέντρο  $\mathcal{T}$  στο σύνολο  $\mathcal{A}(\mathcal{T})$  των αναθέσεων που υπολογίζονται από το  $\mathcal{A}$  στο  $\mathcal{T}$ .

**Παρατήρηση 2.2.1.** Ας υποθέσουμε ότι το  $\mathcal{A} = (\Gamma, Q_1, \Delta_0, \Delta, F_1)$  είναι ένα αυτόματο 1-αναθέσεων το οποίο έχει την ιδιομορφία να αντιστοιχεί κάθε δέντρο  $\mathcal{T}$  σε ένα σύνολο από μονοσύνολα. Ισχύει τότε,

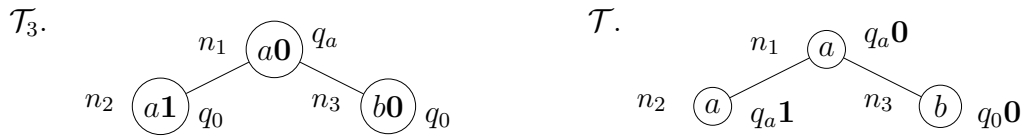
$$\{n\} \in \mathcal{A}(\mathcal{T}) \text{ ανν υπάρχει μία επιτυχημένη εκτέλεση } \rho \text{ του } \mathcal{A} \text{ στο } \mathcal{T} \text{ με διάνυσμα ανάθεσης } \varepsilon_{\{n\}}, \text{ δηλαδή τέτοια ώστε } \rho(n) = (q, 1) \text{ για κάποια } q \in Q.$$

Παρατηρούμε λοιπόν ότι το σύνολο καταστάσεων  $Q \times \{1\} \subset Q_1$  αυτών των αυτομάτων είναι ανάλογο του συνόλου  $S$  των καταστάσεων που επιλέγουν στην περίπτωση των αυτομάτων επιλογής σε δέντρα (selecting tree automata ή STA) του [FGK03], που αποτελούν αυτόματα τα οποία αντιστοιχούν ένα δέντρο  $\mathcal{T}$  σε ένα σύνολο κόμβων του  $\mathcal{T}$ . Θυμίζουμε ότι ένα STA  $\mathcal{A}'$  είναι ένα ζευγάρι  $(\mathcal{A}', S)$  όπου το  $\mathcal{A}'$  είναι ένα αυτόματο για χρωματισμένα δέντρα με σύνολο καταστάσεων  $Q$ , ενώ έχουμε ότι  $S \subseteq Q$ . Ισχύει  $n \in \mathcal{A}'(\mathcal{T})$  ανν κάθε επιτυχημένη εκτέλεση  $\rho$  του  $\mathcal{A}'$  στο  $\mathcal{T}$  είναι τέτοια ώστε  $\rho(n) \in S$ .

Το παράδειγμα που ακολουθεί αποσαφηνίζει την ιδέα της κωδικοποίησης μιας ανάθεσης πάνω στα χρώματα ενός δέντρου αλλά και την ιδέα του ορισμού των αυτομάτων αναθέσεων, ενώ αποτελεί εισαγωγή στην προσέγγιση αυτομάτων για το πρόβλημα της αποτίμησης της MSO σε δέντρα.

**Παράδειγμα 2.2.2.** Ας υποθέσουμε ότι το  $\mathcal{T}$  είναι ένα  $\{a, b\}$ -δέντρο με τρεις κόμβους  $n_1, n_2$  και  $n_3$ , χρώματος  $a, a$  και  $b$  αντίστοιχα, που αποτελούν τη ρίζα, το αριστερό και το δεξί παιδί της ρίζας αντίστοιχα. Ας υποθέσουμε επίσης ότι μελετάμε έναν MSO τύπο  $\phi(X)$ . Υπάρχουν  $2^3 = 8$  υποσύνολα του συνόλου κόμβων  $\{n_1, n_2, n_3\}$ , τα οποία αντιστοιχούν πλήρως στις πιθανές αναθέσεις του τύπου  $\phi(X)$ . Αναλυτικότερα, αυτές είναι οι  $B_1 = \emptyset, B_2 = \{n_3\}, B_3 = \{n_2\}, B_4 = \{n_2, n_3\}, B_5 = \{n_1\}, B_6 = \{n_1, n_3\}, B_7 = \{n_1, n_2\}, B_8 = \{n_1, n_2, n_3\}$ . Θεωρώντας τη διάταξη  $n_1 > n_2 > n_3$  των κόμβων, τα παραπάνω σύνολα  $B_i, i = 1, \dots, 8$ , αναπαρίστανται ως διανύσματα αναθέσεων  $e_i$  με στοιχεία 0, 1 με πολύ φυσικό τρόπο:  $e_1 = (0, 0, 0), e_2 = (0, 0, 1), e_3 = (0, 1, 0), e_4 = (0, 1, 1), e_5 = (1, 0, 0), e_6 = (1, 0, 1), e_7 = (1, 1, 0), e_8 = (1, 1, 1)$ . Θυμίζουμε ότι η εμφάνιση του 1 στην  $i$ -οστή,  $i = 1, 2, 3$ , συνιστώσα του διανύσματος ανάθεσης  $e_j, j = 1, \dots, 8$ , δηλώνει την εμφάνιση του κόμβου  $n_i$  στην αντίστοιχη ανάθεση  $B_j$ .

Ας θεωρήσουμε τώρα τη διανυσματική αναπαράσταση  $c = (a, a, b)$  της συνάρτησης χρωματισμού του δεδομένου δέντρου, ως προς την αρχική διάταξη των κόμβων. Η κατά συνιστώσες συνένωση του διανύσματος  $c$  με κάθε ένα από τα  $e_i$ ,  $i = 1, \dots, 8$ , παράγει τα νέα διανύσματα  $c_1 = (a0, a0, b0)$  –για λόγους απλοποίησης, χρησιμοποιούμε τη συντομογραφία  $(a0, a0, b0)$  για το  $c_1 = ((a, 0), (a, 0), (b, 0))$  και ανάλογα για τα διανύσματα που ακολουθούν–  $c_2 = (a0, a0, b1)$ ,  $c_3 = (a0, a1, b0)$ ,  $c_4 = (a0, a1, b1)$ ,  $c_5 = (a1, a0, b0)$ ,  $c_6 = (a1, a0, b1)$ ,  $c_7 = (a1, a1, b0)$ ,  $c_8 = (a1, a1, b1)$ . Η αντικατάσταση της  $c$  από την  $c_i$ ,  $i = 1, \dots, 8$ , στο αρχικό δέντρο  $\mathcal{T}$ , δημιουργεί 8 νέα χρωματισμένα δέντρα  $\mathcal{T}_i$  με αλφάβητο χρωμάτων το  $\{a, b\} \times \{0, 1\}$ . Θυμίζουμε ότι λόγω της αντιστοιχίας ανάμεσα στα  $B_i$ ,  $e_i$ , κάθε  $(\{a, b\} \times \{0, 1\})$ -χρωματισμένο δέντρο  $\mathcal{T}_i$  –που αποτελεί επέκταση του  $\mathcal{T}$ – κωδικοποιεί στη συνάρτηση χρωματισμού του  $c_i$  την ανάθεση  $B_i$ . Αποδεικνύεται ότι το σύνολο που αποτελείται από εκείνες τις επεκτάσεις  $\mathcal{T}_i$  που κωδικοποιούν τις αναθέσεις που ικανοποιούν τον τύπο  $\phi$  πάνω στο δέντρο  $\mathcal{T}$  είναι MSO ορίσιμο και κατά συνέπεια αναγνωρίσιμο.



Σχήμα 2.1: Οι επιτυχημένες εκτελέσεις  $\rho_3$  και  $\rho_3^*$  των  $A_{P_a}$  και  $\mathcal{A}_{P_a}$  αντίστοιχα.

Πιο συγκεκριμένα, ας υποθέσουμε τώρα ότι ο τύπος που μελετάμε είναι ο  $\phi(X) = P_a(X)$ . Το  $(\{a, b\} \times \{0, 1\})$ -αυτόματο  $A_{P_a}$  –είναι το αυτόματο που κατασκευάζει η απόδειξη του θεωρήματος των Doner, Thatcher & Wright– που αναγνωρίζει τις επεκτάσεις του  $\mathcal{T}$  που κωδικοποιούν τις αναθέσεις που ικανοποιούν τον  $P_a(X)$  δίνεται στον Πίνακα 2.2 (γ') της σελίδας 30. Οι εκτελέσεις  $\rho_i$ ,  $i = 1, \dots, 8$  αυτού του αυτομάτου πάνω στα δέντρα  $\mathcal{T}_i$  δίνονται στον Πίνακα 2.1. Από αυτές μόνοι οι  $\rho_3$  και  $\rho_5$  αποτελούν επιτυχημένες εκτελέσεις, και τα δέντρα  $\mathcal{T}_3$  και  $\mathcal{T}_5$  είναι οι μοναδικές επεκτάσεις που γίνονται δεκτές από το αυτόματο  $A_{P_a}$ . Κατά συνέπεια, οι αναθέσεις  $B_3 = \{n_2\}$ ,  $B_5 = \{n_1\}$  είναι οι μοναδικές αναθέσεις που ικανοποιούν τον  $P_a(X)$  στο  $\mathcal{T}$ . Πράγματι, οι κόμβοι του  $\mathcal{T}$  που έχουν χρώμα  $a$  είναι οι  $n_1, n_2$ .

Θα ήταν όμως πολύ πιο βολικό αν μπορούσαμε να ορίσουμε ένα νέο αυτόματο  $k$ -αναθέσεων  $\mathcal{A}_{P_a}$  το οποίο καθώς θα εκτελείται πάνω στο δέντρο εισόδου  $\mathcal{T}$  –και όχι πάνω στις επεκτάσεις του όπως ισχύει για το  $A_{P_a}$ – θα υπολογίζει μαζί με τις επιτυχημένες εκτελέσεις του και τις αναθέσεις που ικανοποιούν τον  $\phi(X) = P_a(X)$  στο  $\mathcal{T}$ . Οι εκτελέσεις αυτού του νέου αυτομάτου  $\mathcal{A}_{P_a}$ , που συμβολίζονται με  $\rho_i^*$ , δίνονται στον Πίνακα 2.1. Και σε αυτή την περίπτωση, οι εκτελέσεις  $\rho_3^*$  και  $\rho_5^*$  που υπολογίζουν τις αναθέσεις  $B_3$  και  $B_4$  που ικανοποιούν τον  $P_a(X)$  στο  $\mathcal{T}$  είναι οι μοναδικές επιτυχημένες εκτελέσεις του  $\mathcal{A}_{P_a}$ . Στο Σχήμα 2.1 παρουσιάζονται συγκριτικά οι επιτυχημένες εκτελέσεις  $\rho_3$  και  $\rho_3^*$  των  $A_{P_a}$  και  $\mathcal{A}_{P_a}$  πάνω στα αντίστοιχα δέντρα  $\mathcal{T}_3$  και  $\mathcal{T}$ . Όπως μπορεί κανείς εύκολα να παρατηρήσει, η ιδέα της κατασκευής αυτού του νέου αυτομάτου είναι η μετακίνηση των διανυσμάτων αναθέσεων από τη συνάρτηση χρωματισμού των επεκτάσεων, στις καταστάσεις του νέου αυτομάτου.

$i$	$B_i \equiv e_i$	$c_i = c; e_i$	$\rho_i$	$c$	$\rho_i^* = \rho_i; e_i$
1	$\emptyset \equiv (0, 0, 0)$	$(a0, a0, b0)$	$q_0, q_0, q_0$	$(a, a, b)$	$q_00, q_00, q_00$
2	$\{n_3\} \equiv (0, 0, 1)$	$(a0, a0, b1)$	$q_f, q_0, q_f$	$(a, a, b)$	$q_f0, q_00, q_f1$
3	$\{n_2\} \equiv (0, 1, 0)$	$(a0, a1, b0)$	$q_a, q_a, q_0$	$(a, a, b)$	$q_a0, q_a1, q_00$
4	$\{n_2, n_3\} \equiv (0, 1, 1)$	$(a0, a1, b1)$	$q_f, q_a, q_f$	$(a, a, b)$	$q_f0, q_a1, q_f1$
5	$\{n_1\} \equiv (1, 0, 0)$	$(a1, a0, b0)$	$q_a, q_0, q_0$	$(a, a, b)$	$q_a1, q_00, q_00$
6	$\{n_1, n_3\} \equiv (1, 0, 1)$	$(a1, a0, b1)$	$q_f, q_0, q_f$	$(a, a, b)$	$q_f1, q_00, q_f1$
7	$\{n_1, n_2\} \equiv (1, 1, 0)$	$(a1, a1, b0)$	$q_0, q_a, q_f$	$(a, a, b)$	$q_01, q_a1, q_f0$
8	$\{n_1, n_2, n_3\} \equiv (1, 1, 1)$	$(a1, a1, b1)$	$q_f, q_a, q_f$	$(a, a, b)$	$q_f1, q_a1, q_f1$

Πίνακας 2.1: Οι εκτελέσεις των αυτομάτων  $A_{P_a}$  και  $A_{P_b}$  στα δέντρα  $T_i$  και  $T$ .

### 2.2.2 Η σχέση της MSO με τα Αυτόματα Αναθέσεων

Στην ενότητα αυτή, θα δείξουμε πώς το πρόβλημα αποτίμησης της MSO ανάγεται σε ένα πρόβλημα αυτομάτων (Θεώρημα 2.2.1). Η αναγωγή εκφράζεται μέσω των αυτομάτων αναθέσεων και βασίζεται στο γνωστό αποτέλεσμα που αφορά στην ορισμότητα της κλάσης των χρωματισμένων δέντρων που κωδικοποιούν τις αναθέσεις που ικανοποιούν έναν MSO τύπο  $\phi(X_1, \dots, X_k)$ . Πιο συγκεκριμένα, το θεώρημα των Doner, Thatcher & Wright αποδεικνύει την ύπαρξη ενός αυτομάτου  $A_\phi$  το οποίο εκτελείται σε χρωματισμένα δέντρα που κωδικοποιούν όλες τις πιθανές αναθέσεις του  $\phi$ , και αποδέχεται μόνο τα δέντρα εκείνα που κωδικοποιούν αναθέσεις που ικανοποιούν τον  $\phi$ .

Η απόδειξη γίνεται με επαγωγή στη δομή του  $\phi$ : η βάση της επαγωγής αντιστοιχεί στην κατασκευή αυτομάτων  $A_\phi$  για τους ατομικούς τύπους. Τα αυτόματα αυτά δίνονται στον Πίνακα 2.2. Στα αυτόματα αυτά, οι καταστάσεις της μορφής  $q_0$  δηλώνουν γενικά ότι κανένας κόμβος δεν έχει επιλεγεί ακόμα, η κατάσταση  $q_f$  δηλώνει αποτυχία αποδοχής ενώ η  $q_a$  είναι πάντα η τελική κατάσταση και δηλώνει ότι μια ανάθεση που ικανοποιεί τον τύπο έχει ήδη επιλεγεί. Πιο συγκεκριμένα, το αυτόματο για τον  $\text{In}(x, Y)$  εκτελείται σε ένα δέντρο ως εξής: i) αναθέτει σε ένα φύλλο  $n$  την κατάσταση  $q_0$  όταν αυτό δεν έχει επιλεγεί ως ανάθεση για την πρώτη θέση « $x$ » του  $\text{In}(x, Y)$ , ii) αναθέτει την κατάσταση  $q_a$  στο φύλλο  $n$  όταν ο εν λόγω κόμβος έχει επιλεγεί τόσο στην πρώτη θέση όσο και στη δεύτερη θέση, και κατά συνέπεια έχει υπολογιστεί η ανάθεση  $(n, \{n\})$  που ικανοποιεί τον  $\text{In}(x, Y)$  και iii) αναθέτει στο φύλλο  $n$  την κατάσταση  $q_f$  όταν το  $n$  ενώ έχει επιλεγεί στην πρώτη θέση, δεν έχει επιλεγεί στη δεύτερη και κατά συνέπεια η ανάθεση αυτή δε μπορεί με κανένα τρόπο να επεκταθεί σε ανάθεση που ικανοποιεί τον  $\text{In}(x, Y)$ . Καθώς το δέντρο διασχίζεται από τα φύλλα προς τη ρίζα, οι μεταβάσεις του αυτομάτου ορίζονται έτσι ώστε να γίνονται δεκτές μόνο οι επεκτάσεις εκείνες που κωδικοποιούν αναθέσεις της μορφής  $(n, \{n\} \cup A)$  για  $A \subseteq T$ . Έτσι, μια κατάσταση  $q_f$  που έχει ανατεθεί σε κάποιον κόμβο θα πρέπει να προωθείται πάντα ως τη ρίζα, ενώ μία κατάσταση  $q_a$  θα πρέπει να προωθείται ως τη ρίζα μόνο όταν εξακολουθεί να ικανοποιεί τη συνθήκη που αντιπροσωπεύει. Πιο συγκεκριμένα, καθώς η ανάθεση κατάστασης  $q_a$  σε



κάποιον κόμβο  $n$  δηλώνει τον υπολογισμό μιας ανάθεσης της μορφής  $(n, \{n\} \cup A)$ , θα πρέπει σε όλους του υπόλοιπους κόμβους η επιλογή να αφορά μόνο στη δεύτερη θέση του  $\text{In}(x, Y)$ : μία επιπλέον επιλογή κόμβου για την πρώτη θέση συνεπάγεται άμεσα αποτυχία αποδοχής που δηλώνεται με την κατάσταση  $q_f$ . Ανάλογες συνθήκες ισχύουν και για την περίπτωση του αυτομάτου για τον  $S_1(x, y)$ . Αρκεί να παρατηρήσουμε ότι οι επιτυχημένες εκτελέσεις του εν λόγω αυτομάτου δέχονται επεκτάσεις οι οποίες επιλέγουν ακριβώς δυο κόμβους οι οποίοι μάλιστα θα πρέπει να ικανοποιούν τα εξής: κάθε κόμβος που επιλέγεται στην δεύτερη θέση του  $S_1(x, y)$  θα πρέπει να είναι το αριστερό παιδί ενός κόμβου, ενώ κάθε κόμβος που επιλέγεται στην πρώτη θέση θα πρέπει να είναι οπωσδήποτε εσωτερικός κόμβος που το αριστερό του παιδί έχει ήδη επιλεγεί από την εν λόγω εκτέλεση. Έτσι, η επιλογή ενός φύλλου στην πρώτη θέση οδηγεί σε μία μη επιτυχημένη εκτέλεση το οποίο και δηλώνεται με την απόδοση της κατάστασης  $s_f$ , ενώ η απόδοση της κατάστασης  $s_1$  (αντίστοιχα  $s_0$ ) σε ένα φύλλο δηλώνει ότι το φύλλο αυτό έχει επιλεγεί (αντίστοιχα δεν έχει επιλεγεί) στη δεύτερη θέση. Ακριβώς ανάλογες αποδόσεις καταστάσεων πραγματοποιούνται και σε έναν εσωτερικό κόμβο όταν κανένας κόμβος δεν έχει επιλεγεί μέχρι τώρα, δηλαδή όταν και στα δύο του παιδιά έχει αποδοθεί κατάσταση  $s_0$ . Η τελική κατάσταση  $s_a$  αποδίδεται σε έναν εσωτερικό κόμβο ο οποίος επιλέγεται ως ανάθεση για την πρώτη θέση του  $S_1(x, y)$ , όταν στο αριστερό παιδί του έχει αποδοθεί κατάσταση  $s_1$  –δηλαδή μόνο το αριστερό παιδί έχει επιλεγεί σε όλο το αριστερό υποδέντρο– και κανένας κόμβος δεν έχει επιλεγεί στο δεξί υποδέντρο (η κατάσταση στο δεξί παιδί είναι  $s_0$ ). Μόλις αποδοθεί η κατάσταση  $s_a$  σε έναν κόμβο, αυτή προωθείται μέχρι τη ρίζα μόνο όταν κανένας άλλος κόμβος δεν επιλεγεί μέχρι τη ρίζα από την τρέχουσα εκτέλεση. Στην αντίθετη περίπτωση, έχουμε απόδοση της κατάστασης  $s_f$  η οποία πάντα προωθείται μέχρι τη ρίζα για να δηλώσει την αποτυχία αποδοχής της συγκεκριμένης επέκτασης. Ανάλογα ερμηνεύονται οι μεταβάσεις του αυτομάτου για τον  $P_a(x)$ . Αρκεί να παρατηρήσουμε ότι κάθε επιτυχημένη εκτέλεση επιλέγει ακριβώς έναν κόμβο που επιπρόσθετα έχει χρώμα  $a$  και ότι αυτές είναι οι μοναδικές επιτυχημένες εκτελέσεις. Τέλος, τα βήματα της επαγωγής κατασκευής των αυτομάτων για τους μη ατομικούς τύπους παρουσιάζονται στο Παράρτημα Α'.

Το αυτόματο  $A_\phi$  παρέχει τις αναθέσεις που ικανοποιούν τον  $\phi$  με έναν «έμμεσο» τρόπο καθώς λειτουργεί με τον συνήθη τρόπο, δηλαδή ως ένα αυτόματο που αναγνωρίζει. Στην ενότητα αυτή, εισάγουμε την έννοια του αυτομάτου αναθέσεων για τον  $\phi$  το οποίο είναι ένα αυτόματο  $k$ -αναθέσεων που όταν εκτελείται σε ένα δεδομένο δέντρο  $\mathcal{T}$ , παράγει τις αναθέσεις που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$  με έναν πιο άμεσο τρόπο, δηλαδή κωδικοποιημένες στις επιτυχημένες εκτελέσεις του. Πράγματι, το Θεώρημα 2.2.1 αποδεικνύει ότι για κάθε MSO τύπο  $\phi$ , υπάρχει ένα τέτοιο αυτόματο το οποίο συμβολίζεται  $\text{Assign}_\phi$ . Για την απόδειξη του θεωρήματος αυτού, θεωρούμε αρχικά ένα ντετερμινιστικό αυτόματο  $A_\phi$  που ικανοποιεί τις απαιτήσεις του θεωρήματος των Doner, Thatcher & Wright και κατασκευάζουμε το  $\text{Assign}_\phi$  αλλάζοντας τις μεταβάσεις του πρώτου ως εξής: μετακινούμε τα διανύσματα αναθέσεων από το χρώμα –αποτελούν τμήμα του χρώματος στις μεταβάσεις του  $A_\phi$ – στις καταστάσεις του νέου αυτομάτου. Πράγματι, σε κάθε εκτέλεση του  $\text{Assign}_\phi$  στο  $\mathcal{T} = (t, c)$ , κατάλληλες καταστάσεις επιλέγονται με μη ντετερμινιστικό τρόπο έτσι ώστε η  $\varsigma; \varepsilon$  να είναι μία επιτυχημένη εκτέλεση του  $\text{Assign}_\phi$  αν το αυτόματο  $A_\phi$  αποδέχεται το δέντρο  $(t, c; \varepsilon)$  μέσω της επιτυχημένης του

εκτέλεσης  $\varsigma$ . Συνεπώς, το  $\text{Assign}_\phi$  παράγει τις αναθέσεις που είναι κωδικοποιημένες στα δέντρα που αποδέχεται το  $A_\phi$ , και οι οποίες είναι ακριβώς οι αναθέσεις που ικανοποιούν τον  $\phi$ .

**Θεώρημα 2.2.1.** Για κάθε MSO τύπο  $\phi(X_1, \dots, X_k)$ , υπάρχει ένα αυτόματο  $k$ -αναθέσεων  $\text{Assign}_\phi$  τέτοιο ώστε για οποιοδήποτε δέντρο  $\mathcal{T}$  να υπολογίζει τις αναθέσεις που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$ , δηλαδή έτσι ώστε  $\text{Assign}_\phi(\mathcal{T}) = \phi(\mathcal{T})$ .

**Απόδειξη.** Ας υποθέσουμε ότι το  $\mathcal{T} = (t, c)$  είναι ένα  $\Gamma$ -δέντρο, ότι το  $\overline{B}$  είναι μία  $k$ -ανάθεση του  $\mathcal{T}$  και ότι  $\Gamma_k = \Gamma \times \{0, 1\}^k$ . Ορίζουμε το  $\Gamma_k$ -δέντρο  $(t, c')$ , και το συμβολίζουμε  $(\mathcal{T}, \overline{B})$ , επεκτείνοντας τη συνάρτηση χρωματισμού  $c$  του  $\mathcal{T}$  με το  $\varepsilon_{\overline{B}}$ , δηλαδή έτσι ώστε  $c' = c; \varepsilon_{\overline{B}}$ . Από το θεώρημα των Doner [Don70], Thatcher & Wright [TW68], έχουμε ότι για κάθε MSO τύπο  $\phi(X_1, \dots, X_k)$  υπάρχει ένα ντετερμινιστικό  $\Gamma_k$ -αυτόματο  $A_\phi$  που αναγνωρίζει την κλάση όλων των  $\Gamma_k$ -δέντρων  $(\mathcal{T}, \overline{B})$ , όπου το  $\mathcal{T}$  είναι ένα οποιοδήποτε  $\Gamma$ -δέντρο και η  $\overline{B}$  είναι μία ανάθεση που ικανοποιεί τον  $\phi$  στο  $\mathcal{T}$ . Δηλαδή ισχύει το εξής: μία εκτέλεση του  $A_\phi$  σε ένα δέντρο  $(\mathcal{T}, \overline{B})$  είναι επιτυχημένη αν  $\mathcal{T} \models \phi(\overline{B})$ .

Ξεκινώντας από το  $A_\phi$  ορίζουμε ένα  $\Gamma$ -αυτόματο  $k$ -αναθέσεων  $\text{Assign}_\phi$  το οποίο υπολογίζει όλα τα διανύσματα  $\varepsilon_{\overline{B}}$  έτσι ώστε  $\mathcal{T} \models \phi(\overline{B})$ . Η ιδέα της κατασκευής είναι η μετακίνηση των κωδικοποιημένων αναθέσεων  $\varepsilon_{\overline{B}}$  από τη συνάρτηση χρωματισμού  $c'$  μέσα στις καταστάσεις του νέου αυτομάτου με τέτοιο τρόπο ώστε η  $\varsigma$  να είναι μία εκτέλεση του  $A_\phi$  στο  $\Gamma_k$ -δέντρο  $(\mathcal{T}, \overline{B}) = (t, c; \varepsilon_{\overline{B}})$  αν η  $\varsigma; \varepsilon_{\overline{B}}$  είναι μία εκτέλεση του  $\text{Assign}_\phi$  στο  $\Gamma$ -δέντρο  $\mathcal{T} = (t, c)$ .

Πιο συγκεκριμένα, έστω ότι  $A_\phi = (\Gamma \times \{0, 1\}^k, Q, \delta_0, \delta, F)$ . Ορίζουμε το

$$\text{Assign}_\phi = (\Gamma, Q \times \{0, 1\}^k, \Delta_0, \Delta, F \times \{0, 1\}^k)$$

έτσι ώστε

- (i)  $(\gamma, (q, \bar{\varepsilon})) \in \Delta_0$  αν  $\delta_0((\gamma, \bar{\varepsilon})) = q$  και
- (ii) για κάθε  $\bar{\varepsilon}_1, \bar{\varepsilon}_2 \in \{0, 1\}^k$ ,  $((q_1, \bar{\varepsilon}_1), (q_2, \bar{\varepsilon}_2), \gamma, (q, \bar{\varepsilon})) \in \Delta$  αν  $\delta(q_1, q_2, (\gamma, \bar{\varepsilon})) = q$ .

Είναι προφανές ότι το  $\text{Assign}_\phi$  είναι ένα αυτόματο αναθέσεων για τον  $\phi$ , δηλαδή ένα αυτόματο που όταν εκτελείται στο  $\mathcal{T}$  υπολογίζει το  $\phi(\mathcal{T})$ .  $\square$

Σημειώνουμε ότι η απευθείας κατασκευή του  $\text{Assign}_\phi$  με επαγωγή στη δομή του  $\phi$ , ακριβώς ανάλογα με τον ορισμό του αυτομάτου  $A_\phi$  στην απόδειξη του θεωρήματος των Doner, Thatcher & Wright, αποτελεί μία εναλλακτική «κατασκευαστική» απόδειξη του παραπάνω θεωρήματος. Όπως θα δούμε παρακάτω, χρησιμοποιούμε αυτή την κατασκευαστική εκδοχή της απόδειξης στο περιβάλλον της datalog όταν ορίζουμε το πρόγραμμα  $\Pi_\phi$  που παράγει το  $\text{Assign}_\phi$ .

Από εδώ και στο εξής, κάθε φορά που αναφερόμαστε στο  $A_\phi$  (αντίστοιχα στο  $\text{Assign}_\phi$ ) θα εννοούμε οποιοδήποτε ντετερμινιστικό αυτόματο για τον  $\phi$ , δηλαδή όπως αυτό ορίζεται μέσω του θεωρήματος των Doner, Thatcher & Wright (αντίστοιχα οποιοδήποτε αυτόματο αναθέσεων για τον  $\phi$ , δηλαδή που ικανοποιεί τις απαιτήσεις του Θεωρήματος 2.2.1). Ωστόσο, οποτεδήποτε τα  $A_\phi$  και  $\text{Assign}_\phi$  αναφέρονται μαζί, θα θεωρούμε ότι το  $\text{Assign}_\phi$  είναι το αυτόματο αναθέσεων για τον  $\phi$ , μοναδικά ορισμένο από το συγκεκριμένο  $A_\phi$ . Ο λόγος για

τον οποίο επιλέγουμε το  $A_\phi$  να είναι πάντα ντετερμινιστικό, είναι επειδή στην περίπτωση αυτή οι μεταβάσεις του έχουν καλές ιδιότητες που έχουν ως αποτέλεσμα καλύτερα αποτελέσματα πολυπλοκότητας (βλέπε για παράδειγμα τον Ορισμό 2.2.3 και το Λήμμα 2.2.2).

**Παράδειγμα 2.2.3.** (συνέχεια) Ας υποθέσουμε ότι το  $\mathcal{T}$  είναι ένα χρωματισμένο δέντρο με τρεις κόμβους  $n_1$  (ρίζα χρώματος  $b$ ),  $n_2$  (αριστερό παιδί χρώματος  $a$ ),  $n_3$  (δεξί παιδί χρώματος  $b$ ). Ας υποθέσουμε επίσης ότι οι αναθέσεις  $\overline{B}_1, \overline{B}_2$  του Παραδείγματος 2.2.1 είναι οι αναθέσεις που ικανοποιούν κάποιον MSO τύπο  $\phi(X_1, X_2)$  στο  $\mathcal{T}$ . Ένας τέτοιος τύπος είναι ο

$$\phi(x, Y) \equiv P_a(x) \wedge \exists z \exists y (S_1(z, x) \wedge S_2(z, y) \wedge \text{In}(y, Y) \wedge \neg \text{In}(x, Y))$$

που εκφράζει ότι «ο κόμβος  $x$  έχει χρώμα  $a$  και το  $Y$  είναι ένα σύνολο κόμβων που περιέχει τον αδερφό του κόμβου  $x$  αλλά όχι τον  $x$ ». Στην περίπτωση αυτή, το  $\text{Assign}_\phi$  έχει δυο επιτυχημένες εκτελέσεις στο  $\mathcal{T}$ :

$$\rho_1 = ((q_1, 0, 1), (q_2, 1, 0), (q_3, 0, 1)) \quad \text{και} \quad \rho_2 = ((s_1, 0, 0), (s_2, 1, 0), (s_3, 0, 1))$$

διατεταγμένες σύμφωνα με την προδιατεταγμένη διάσχιση  $o = (n_1, n_2, n_3)$  των κορυφών του  $\mathcal{T}$ , δηλαδή με διανύσματα αναθέσεων  $\varepsilon_1$  και  $\varepsilon_2$  που αντιστοιχούν στις  $\overline{B}_1$  και  $\overline{B}_2$  αντίστοιχα.

**Παρατήρηση 2.2.2.** Όταν ο  $\phi$  είναι μία πρόταση, το ντετερμινιστικό bottom-up  $\Gamma$ -αυτόματο  $A_\phi$  εκτελείται στο δέντρο εισόδου  $\mathcal{T}$  και το αποδέχεται αν  $\mathcal{T} \models \phi$ . Στην περίπτωση αυτή, το  $\text{Assign}_\phi$  απλά συμβολίζει μία εναλλακτική αναπαράσταση του

$$A_\phi = (\Gamma, Q, \delta_0, \delta, F)$$

όπου οι συναρτήσεις  $\delta_0$  και  $\delta$  του  $A_\phi$  αντικαθίστανται στο  $\text{Assign}_\phi$  από τις αναπαραστάσεις τους ως σχέσεις  $\Delta_0$  και  $\Delta$  αντίστοιχα. Αξίζει να σημειωθεί ότι τότε το  $\text{Assign}_\phi$  έχει μία μοναδική εκτέλεση, όπως προφανώς ισχύει για το  $A_\phi$ .

**Το  $A_\phi$  ως εναλλακτική αναπαράσταση του  $\text{Assign}_\phi$ : οι  $\phi$ -μεταβάσεις.** Το λήμμα που ακολουθεί αποτελεί άμεση συνέπεια του ορισμού του  $\text{Assign}_\phi$  από το  $A_\phi$  και ουσιαστικά δείχνει ότι θα μπορούσαμε ισοδύναμα να έχουμε ορίσει το  $\text{Assign}_\phi$  ως το ίδιο το  $A_\phi$ , απλά τροποποιώντας τον συνήθη ορισμό της εκτέλεσης.

**Λήμμα 2.2.1.** Οι εκτελέσεις του  $\text{Assign}_\phi$  μπορούν να υπολογιστούν από τις μεταβάσεις του  $A_\phi$ .

**Απόδειξη.** Ας υποθέσουμε ότι η  $\rho$  είναι μία εκτέλεση του  $\text{Assign}_\phi$  σε ένα δέντρο  $\mathcal{T}$ . Από τον ορισμό της εκτέλεσης και τη σχέση που συνδέει τα  $\text{Assign}_\phi$  και  $A_\phi$  έχουμε ότι όταν το  $n$  είναι ένα φύλλο χρώματος  $a$ , τότε  $\rho(n) = q\bar{e}$  αν υπάρχει μία μετάβαση  $(a, q\bar{e}) \in \Delta_0$  ή ισοδύναμα αν  $\delta_0((a, \bar{e})) = q$ . Ανάλογα, όταν ο  $n$  είναι εσωτερικός κόμβος χρώματος  $a$  με παιδιά  $n_1, n_2$  για τα οποία έχουμε  $\rho(n_1) = q_1\bar{e}_1, \rho(n_2) = q_2\bar{e}_2$ , τότε  $\rho(n) = q\bar{e}$  αν  $(q_1\bar{e}_1, q_2\bar{e}_2, a, q\bar{e}) \in \Delta$  ή ισοδύναμα αν  $\delta(q_1, q_2, (a, \bar{e})) = q$ .  $\square$

(α')  $\phi(x, Y) \equiv Y(x)$ :  $Q = \{q_0, q_a, q_f\}$ ,  $F = \{q_a\}$ .

$\delta_0$	
$q$	$\bar{e}$
$q_0$	00
$q_0$	01
$q_f$	10
$q_a$	11

$\delta$			
$q_1$	$q_2$	$q$	$\bar{e}$
$q_0$	$q_0$	$q_0$	00
$q_0$	$q_0$	$q_0$	01
$q_0$	$q_0$	$q_f$	10
$q_0$	$q_0$	$q_a$	11
$q_a$	$q_0$	$q_a$	00
$q_a$	$q_0$	$q_a$	01
$q_a$	$q_0$	$q_f$	10
$q_a$	$q_0$	$q_f$	11

$q_1$	$q_2$	$q$	$\bar{e}$
$q_0$	$q_a$	$q_a$	00/01
$q_0$	$q_a$	$q_f$	10/11
$q_a$	$q_a$	$q_f$	00/01/10/11
$q_f$	$q_0$	$q_f$	00/01/10/11
$q_0$	$q_f$	$q_f$	00/01/10/11
$q_f$	$q_a$	$q_f$	00/01/10/11
$q_a$	$q_f$	$q_f$	00/01/10/11
$q_f$	$q_f$	$q_f$	00/01/10/11

(β')  $\phi(x, y) \equiv S_1(x, y)$ :  $Q = \{s_0, s_1, s_a, s_f\}$ ,  $F = \{s_a\}$ .

$\delta_0$	
$q$	$\bar{e}$
$s_0$	00
$s_1$	01
$s_f$	10/11

$\delta$			
$q_1$	$q_2$	$q$	$\bar{e}$
$s_0$	$s_0$	$s_0$	00
$s_0$	$s_0$	$s_1$	01
$s_0$	$s_0$	$s_f$	10/11
$s_0$	$s_1$	$s_f$	00/01/10/11
$s_1$	$s_0$	$s_f$	00/01
$s_1$	$s_0$	$s_a$	10
$s_1$	$s_0$	$s_f$	11
$s_1$	$s_1$	$s_f$	00/01/10/11
$s_a$	$s_0$	$s_a$	00
$s_a$	$s_0$	$s_f$	01
$s_a$	$s_0$	$s_f$	10/11

$q_1$	$q_2$	$q$	$\bar{e}$
$s_0$	$s_a$	$s_a$	00
$s_0$	$s_a$	$s_f$	01
$s_0$	$s_a$	$s_f$	10/11
$s_a$	$s_a$	$s_f$	00/01/10/11
$s_f$	$s_0$	$s_f$	00/01/10/11
$s_0$	$s_f$	$s_f$	00/01/10/11
$s_f$	$s_1$	$s_f$	00/01/10/11
$s_1$	$s_f$	$s_f$	00/01/10/11
$s_f$	$s_a$	$s_f$	00/01/10/11
$s_a$	$s_f$	$s_f$	00/01/10/11
$s_f$	$s_f$	$s_f$	00/01/10/11

(γ')  $\phi(x) \equiv P_a(x)$ :  $Q = \{q_0, q_a, q_f\}$ ,  $F = \{q_a\}$ .

$\delta_0$	
$\gamma$	$q, e$
$a$	$q_0, 0$
$a$	$q_a, 1$
$b$	$q_0, 0$
$b$	$q_f, 1$

$\delta$			
$q_1$	$q_2$	$\gamma$	$q, e$
$q_0$	$q_0$	$a$	$q_0, 0$
$q_0$	$q_0$	$a$	$q_a, 1$
$q_0$	$q_0$	$b$	$q_0, 0$
$q_0$	$q_0$	$b$	$q_f, 1$
$q_a$	$q_a$	$a/b$	$q_f, 0$
$q_a$	$q_a$	$a/b$	$q_f, 1$

$q_1$	$q_2$	$\gamma$	$q, e$
$q_a$	$q_0$	$a/b$	$q_a, 0$
$q_a$	$q_0$	$a/b$	$q_f, 1$
$q_0$	$q_a$	$a/b$	$q_a, 0$
$q_0$	$q_a$	$a/b$	$q_f, 1$
$q_f$	$q_0$	$a/b$	$q_f, 0$
$q_f$	$q_0$	$a/b$	$q_f, 1$
$q_f$	$q_a$	$a/b$	$q_f, 0$
$q_f$	$q_a$	$a/b$	$q_f, 1$

$q_1$	$q_2$	$\gamma$	$q, e$
$q_f$	$q_f$	$a/b$	$q_f, 0$
$q_f$	$q_f$	$a/b$	$q_f, 1$
$q_0$	$q_f$	$a/b$	$q_f, 0$
$q_0$	$q_f$	$a/b$	$q_f, 1$
$q_a$	$q_f$	$a/b$	$q_f, 0$
$q_a$	$q_f$	$a/b$	$q_f, 1$

Πίνακας 2.2: Τα αυτόματα  $A_\phi$  για τους ατομικούς τύπους  $\phi(x, Y) \equiv Y(x)$ ,  $\phi(x, y) \equiv S_1(x, y)$  και  $\phi(x) \equiv P_a(x)$  σε πεπερασμένα δυαδικά δέντρα.

Οι μεταβάσεις του  $A_\phi$  είναι πιο απλές, και για αυτό το λόγο συνεπάγονται μικρότερου μεγέθους δεδομένα εισόδου –που εξακολουθούν βέβαια να είναι ιδιαίτερα μεγάλα ούτως ή αλλιώς– στη περίπτωση της σχεσιακής άλγεβρας αλλά και μικρότερου μεγέθους προγράμματα στην datalog. Άμεση συνέπεια αυτού του γεγονότος είναι βελτιστοποιήσεις που αφορούν στην πολυπλοκότητα των αντίστοιχων λύσεων. Για το λόγο αυτό, επιλέγουμε να χρησιμοποιούμε –κατά τον υπολογισμό των εκτελέσεων του  $\text{Assign}_\phi$ – τις μεταβάσεις του  $A_\phi$  με τη μορφή που δίνονται στον παρακάτω Ορισμό 2.2.3. Μάλιστα από εδώ και στο εξής τις αναφέρουμε και αυτές ως μεταβάσεις του  $\text{Assign}_\phi$ .

**Ορισμός 2.2.3.** Θεωρούμε το αυτόματο  $A_\phi = (\Gamma \times \{0, 1\}^k, Q, \delta_0, \delta, F)$  και το αυτόματο  $\text{Assign}_\phi = (\Gamma, Q \times \{0, 1\}^k, \Delta_0, \Delta, F \times \{0, 1\}^k)$ . Ονομάζουμε  $\phi$ -καταστάσεις τα στοιχεία του συνόλου  $Q$  και  $\phi$ -μεταβάσεις τα στοιχεία των συνόλων  $D_0, D'_0, D, D'$  που ορίζονται παρακάτω (όπου θεωρούμε ότι  $\bar{e} = (e_1, \dots, e_k)$  και  $e = e_1 \cdots e_k$ ):

- $D_0 = \{\gamma q e \mid (\gamma, (q, \bar{e})) \in \Delta_0\} = \{\gamma q e \mid \delta_0((\gamma, \bar{e})) = q\}$  με πληθάρημο  $|\Gamma| \cdot 2^k$ ,
- $D'_0 = \{\gamma q \mid \text{υπάρχει } e \text{ τ.ω. } \gamma q e \in D_0\}$  με πληθάρημο το πολύ  $|\Gamma| \cdot 2^k$ ,
- $D = \{\gamma q q_1 q_2 e \mid \text{υπάρχουν } \bar{e}_1, \bar{e}_2 \text{ τ.ω. } ((q_1, \bar{e}_1), (q_2, \bar{e}_2), \gamma, (q, \bar{e})) \in \Delta\}$   
 $= \{\gamma q q_1 q_2 e \mid \delta(q_1, q_2, (\gamma, \bar{e})) = q\}$  με πληθάρημο  $|\Gamma| \cdot |Q|^2 \cdot 2^k$ ,
- $D' = \{\gamma q q_1 q_2 \mid \text{υπάρχει } e \text{ τ.ω. } \gamma q q_1 q_2 e \in D\}$  με πληθάρημο το πολύ  $|\Gamma| \cdot |Q|^2 \cdot 2^k$ .

Για κάθε σύνολο  $D$  που ορίστηκε παραπάνω, ορίζουμε τα  $D|_\gamma = \{u \mid \gamma u \in D\}$ .

Οι  $\phi$ -καταστάσεις και οι  $\phi$ -μεταβάσεις είναι στην πραγματικότητα  $A_\phi$ -/ $\text{Assign}_\phi$ - καταστάσεις και μεταβάσεις, δηλαδή τα  $Q, D_0, D'_0, D, D'$  ορίζονται ως προς ένα συγκεκριμένο αυτόματο  $A_\phi$ . Οποτεδήποτε τα χρησιμοποιούμε θα είναι εμφανές από τα συμφραζόμενα σε ποιο αυτόματο  $A_\phi$ / $\text{Assign}_\phi$  αντιστοιχούν. Και το ίδιο ισχύει και για τον Ορισμό 2.2.4 παρακάτω.

Σημειώνουμε ότι ο λόγος που στους παραπάνω ορισμούς χρησιμοποιούμε τη λέξη  $e$  ως αναπαράσταση του διανύσματος  $\bar{e}$  είναι επειδή τόσο στη σχεσιακή άλγεβρα όσο και στην datalog η επιλογή αυτή μας οδηγεί σε σχέσεις/προγράμματα με λιγότερες μεταβλητές.

**Παρατήρηση 2.2.3.** Όταν η  $\phi$  είναι πρόταση, το αντίστοιχο αυτόματο είναι της μορφής  $\text{Assign}_\phi = (\Gamma, Q, \Delta_0, \Delta, F)$  και δεν υπάρχει  $e$  στις σχέσεις που περιγράφουν τις μεταβάσεις. Έτσι, από τις  $\phi$ -μεταβάσεις του Ορισμού 2.2.3 οι μόνες που έχουν νόημα είναι οι  $D'_0$  και  $D'$ :  $\gamma q \in D'_0$  ανν  $(\gamma, q) \in \Delta_0$   $\gamma q q_1 q_2 \in D'$  ανν  $(q_1, q_2, \gamma, q) \in \Delta$ .

Ο ορισμός που ακολουθεί τυποποιεί –υπό την ονομασία πιθανές (αντίστοιχα επιτυχημένες) μεταβάσεις/καταστάσεις– την έννοια των μεταβάσεων/καταστάσεων που εμφανίζονται σε μια εκτέλεση (αντίστοιχα επιτυχημένη εκτέλεση). Σε όλη τη διατριβή, υπολογίζουμε επιτυχημένες εκτελέσεις συνθέτοντας με κατάλληλο τρόπο είτε επιτυχημένες μεταβάσεις/καταστάσεις ή κατάλληλα υποσύνολα αυτών.

**Ορισμός 2.2.4.** Ας θεωρήσουμε ότι το  $\text{Assign}_\phi$  εκτελείται σε ένα δέντρο  $\mathcal{T}$ . Κάθε στοιχείο  $qe$  του  $D_0|_\gamma$  ονομάζεται *έγκυρη  $\phi$ -μετάβαση* για κάθε φύλλο  $n$  χρώματος  $\gamma$ . Επιπρόσθετα, αυτή ονομάζεται *πιθανή (αντίστοιχα επιτυχημένη)  $\phi$ -μετάβαση* στο  $n$  αν υπάρχει μία εκτέλεση (αντίστοιχα επιτυχημένη εκτέλεση)  $\rho$  του  $\text{Assign}_\phi$  στο  $\mathcal{T}$  τέτοια ώστε  $\rho(n) = q\bar{e}$ . Κάθε στοιχείο  $qq_1q_2e$  του  $D|_\gamma$  ονομάζεται *έγκυρη  $\phi$ -μετάβαση* στον εσωτερικό κόμβο  $n$  χρώματος  $\gamma$ . Επιπρόσθετα, αυτή ονομάζεται *πιθανή (αντίστοιχα επιτυχημένη)  $\phi$ -μετάβαση* στον  $n$  εάν υπάρχει μία εκτέλεση (αντίστοιχα επιτυχημένη εκτέλεση)  $\rho$  του  $\text{Assign}_\phi$  στο  $\mathcal{T}$  τέτοια ώστε  $\rho(n) = q\bar{e}$  και  $\rho(n_i) = q_i\bar{e}_i$ , για κάποια  $e_i$ ,  $i = 1, 2$ . Η κατάσταση  $q$  (ή  $qe$ ) που εμφανίζεται σε μία έγκυρη (αντίστοιχα πιθανή, αντίστοιχα επιτυχημένη)  $\phi$ -μετάβαση  $qe$  ή  $qq_1q_2e$  στον  $n$ , ονομάζεται *έγκυρη (αντίστοιχα πιθανή, αντίστοιχα επιτυχημένη) κατάσταση* στον  $n$ .

Κλείνουμε αυτή την ενότητα δίνοντας την ακριβή σύνδεση ανάμεσα στις υποεκτελέσεις, στις εκτελέσεις και στα διανύσματα ανάθεσης. Τα αποτελέσματα αυτά συμβάλουν στον ακριβή προσδιορισμό των αντίστοιχων πληθάρθμων και θα χρησιμοποιηθούν στη συνέχεια στις αναλύσεις πολυπλοκότητας που κάνουμε.

**Λήμμα 2.2.2.** Ας υποθέσουμε ότι ο  $\phi(X_1, \dots, X_k)$  είναι ένας MSO τύπος, το  $\mathcal{T}$  είναι ένα δέντρο μεγέθους  $m$  και το  $\mathcal{T}_n$  είναι ένα πλήρες υποδέντρο του  $\mathcal{T}$  μεγέθους  $m_n$ .

1. Υπάρχει μία αμφιμονοσήμαντη συνάρτηση από τις εκτελέσεις του  $\text{Assign}_\phi$ , πάνω στο  $\mathcal{T}$ , στα διανύσματα  $(k, m)$ -αναθέσεων.
2. Υπάρχει μία αμφιμονοσήμαντη συνάρτηση από τις  $\mathcal{T}_n$ -εκτελέσεις του  $\text{Assign}_\phi$ , πάνω στο  $\mathcal{T}$ , στα διανύσματα  $(k, m_n)$ -αναθέσεων.
3. Υπάρχει μία αμφιμονοσήμαντη συνάρτηση από τις επιτυχημένες εκτελέσεις του  $\text{Assign}_\phi$  στις κωδικοποιημένες αναθέσεις που ικανοποιούν τον  $\phi$ .
4. Υπάρχουν  $2^{km}$  εκτελέσεις του  $\text{Assign}_\phi$  στο  $\mathcal{T}$  και υπάρχουν  $2^{km_n}$  εκτελέσεις του  $\text{Assign}_\phi$  στο  $\mathcal{T}_n$ .

**Απόδειξη.** Προφανώς, οι απεικονίσεις που ορίζονται στα (1), (2), (3) αντιστοιχούν σε κάθε εκτέλεση το αντίστοιχο διάνυσμα ανάθεσης. Οι ιδιότητες αυτών των απεικονίσεων είναι συνέπεια της κατασκευής του  $\text{Assign}_\phi$  από το αυτόματο  $A_\phi = (\Gamma \times \{0, 1\}^k, Q, \delta_0, \delta, F)$ , που παρουσιάστηκε στην απόδειξη του Θεωρήματος 2.2.1. Πιο συγκεκριμένα, το γεγονός ότι οι  $\delta_0, \delta$  είναι ολικές συναρτήσεις αποδεικνύει ότι και αυτές οι απεικονίσεις αποτελούν αμφιμονοσήμαντες συναρτήσεις, είναι δηλαδή 1-1 και επί. Επιπρόσθετα, η απόδειξη του (2) ολοκληρώνεται με την παρατήρηση για την ύπαρξη μιας 1-1 και επί απεικόνισης από τις  $\mathcal{T}_n$ -υποεκτελέσεις του  $\text{Assign}_\phi$  πάνω στο  $\mathcal{T}$  στις εκτελέσεις του  $\text{Assign}_\phi$  πάνω στο  $\mathcal{T}_n$ . Το (4) αποτελεί άμεση συνέπεια των (1) και (2).  $\square$

## 2.3 Επίλυση μέσω της Σχεσιακής Άλγεβρας

Στην ενότητα αυτή αποδεικνύουμε ότι η κωδικοποιημένη εκδοχή του προβλήματος αποτίμησης της MSO ανάγεται σε ένα πρόβλημα αποτίμησης ενός ακυκλικού συζευκτικού ερωτήματος. Η

απόδειξη βασίζεται στο γεγονός ότι ο υπολογισμός των εκτελέσεων ενός αυτομάτου μπορεί να εκφραστεί φυσικά στη σχεσιακή άλγεβρα μέσω της συνένωσης (join) των σχέσεων μετάβασης: συνενώνουμε τόσες σχέσεις μετάβασης όσοι οι κόμβοι του δέντρου εισόδου. Η αποτίμηση του ακυκλικού συζευκτικού ερωτήματος που προκύπτει, μέσω του βελτιστοποιημένου αλγόριθμου του Yannakakis, δίνει μία αποδοτική λύση, που βασίζεται στη σχεσιακή άλγεβρα, για την κωδικοποιημένη εκδοχή του προβλήματος αποτίμησης της MSO σε δέντρα. Ανάμεσα στα ενδιαφέροντα αποτελέσματα της προτεινόμενης προσέγγισης είναι μία εναλλακτική απόδειξη, που βασίζεται στη σχεσιακή άλγεβρα, για την πολυπλοκότητα του προβλήματος αποτίμησης της MSO σε δέντρα.

### 2.3.1 Προκαταρκτικά της σχεσιακής άλγεβρας

Παρουσιάζουμε τις έννοιες της σχεσιακής άλγεβρας που είναι απαραίτητες για την κατανόηση της παρούσας δουλειάς. Ο αναγνώστης παραπέμπεται στα [AHV95, Ull89] για μία πιο ολοκληρωμένη παρουσίαση του χώρου.

**Βασικές έννοιες.** Στο σχεσιακό μοντέλο δεδομένων, τα δεδομένα οργανώνονται ως σχέσεις. Κάθε σχέση έχει ένα όνομα (name) και αναπαρίσταται ως ένας πίνακας, όπου κάθε γραμμή αντιστοιχεί σε ένα στοιχείο της σχέσης και κάθε στήλη ονομάζεται με ένα γνώρισμα (attribute). Σε κάθε όνομα σχέσης  $R$  αντιστοιχεί ένα μη κενό, πεπερασμένο, πλήρως διατεταγμένο σύνολο γνωρισμάτων που ονομάζεται *sort* του  $R$  και συμβολίζεται με  $\text{sort}(R)$ , ενώ συχνά αναπαρίσταται και ως λέξη που διατηρεί τη διάταξη των γνωρισμάτων του. Ο *βαθμός* (arity) του ονόματος σχέσης  $R$  είναι ο πληθάρηθος του *sort* του. Σε κάθε γνώρισμα  $A$  αντιστοιχεί ένα σύνολο τιμών που ονομάζεται *πεδίο ορισμού* (domain) του  $A$  και συμβολίζεται με  $\text{dom}(A)$ . Ας υποθέσουμε ότι το  $U$  είναι ένα μη κενό πεπερασμένο σύνολο γνωρισμάτων: μία  $U$ -πλειάδα είναι μία ολική απεικόνιση

$$t : U \rightarrow \text{dom}(U) = \bigcup_{A \in U} \text{dom}(A).$$

Αν  $X \subseteq U$ , τότε το  $t[X]$  συμβολίζει την  $X$ -πλειάδα που αποτελεί τον περιορισμό της απεικόνισης  $t$  στο  $X$ . Μία  $U$ -πλειάδα  $t$  συχνά αναπαρίσταται ως η διατεταγμένη λίστα

$$\langle t(A_1), \dots, t(A_n) \rangle$$

όπου  $U = A_1 \cdots A_n$ . Ένα σχήμα σχέσης (relation schema) η  $R[U]$  είναι ένα ζευγάρι

$$(R, U)$$

που αποτελείται από ένα όνομα σχέσης  $R$  και ένα σύνολο γνωρισμάτων  $U = \text{sort}(R)$ . Ένα στιγμιότυπο σχέσης (relation instance) ή σχέση  $r$  με σχήμα σχέσης  $R[U]$  είναι ένα πεπερασμένο σύνολο από  $U$ -πλειάδες, στο οποίο αναφερόμαστε συχνά και με τον όρο  $U$ -σχέση. Ένα σχήμα βάσης δεδομένων (database schema)  $D$  είναι ένα μη κενό σύνολο

$$\{R_1[U_1], \dots, R_n[U_n]\}$$

από σχήματα σχέσεων, και ισχύει ότι  $\text{sort}(D) = \{U_1, \dots, U_n\}$ . Ένα στιγμιότυπο βάσης δεδομένων (database instance) ή βάση δεδομένων  $d$  πάνω στο  $D$  είναι ένα πεπερασμένο σύνολο

$$\{r_1, \dots, r_n\}$$

από σχέσεις τ.ω. για  $i = 1, \dots, n$ , η  $r_i$  να έχει σχήμα σχέσης  $R_i[U_i]$ .

Θα λέμε ότι μια  $Y$ -σχέση  $r$  ικανοποιεί την *συναρτησιακή εξάρτηση* (functional dependency)  $X \rightarrow Z$  για  $X, Z \subseteq Y$  και θα το συμβολίζουμε με

$$r \models X \rightarrow Z$$

αν για κάθε ζευγάρι  $(t, s)$  από πλειάδες της  $r$ , η ισότητα  $t[X] = s[X]$  συνεπάγεται την ισότητα  $t[Z] = s[Z]$ . Όταν  $Z = Y$ , η εξάρτηση ονομάζεται *δομική εξάρτηση* (key dependency).

Στην ενότητα αυτή, από το σύνολο των τελεστών της σχεσιακής άλγεβρας χρησιμοποιούμε μόνο την προβολή (projection) και τη φυσική συνένωση (natural join) οι οποίες ορίζονται ως εξής:

- δεδομένης μιας  $Z$ -σχέσης  $r$ , η *προβολή*  $\pi_X(r)$  της  $r$  στο  $X \subseteq Z$  είναι η  $X$ -σχέση

$$\pi_X(r) = \{t[X] \mid t \in r\}$$

- δεδομένων μιας  $X$ -σχέσης  $r_1$  και μιας  $Y$ -σχέσης  $r_2$ , η *φυσική συνένωση* τους  $r_1 \bowtie r_2$  είναι η  $X \cup Y$ -σχέση

$$r_1 \bowtie r_2 = \{t \mid \text{η } t \text{ είναι μία } X \cup Y\text{-πλειάδα τ.ω. } t[X] \in r_1 \text{ και } t[Y] \in r_2\}.$$

Συμβολίζουμε με  $\|r\|$  το μέγεθος μιας  $X$ -σχέσης  $r$  που ορίζεται ως εξής:

$$\|r\| = |X| \cdot |r|$$

όπου το  $|r|$  είναι το πλήθος των πλειάδων που περιέχει η  $r$ . Το κόστος της προβολής πάνω σε μία σχέση μεγέθους  $s$  είναι  $O(s)$  και το κόστος της συνένωσης δύο σχέσεων  $s, t$  που παράγουν μία σχέση μεγέθους  $j$  είναι  $O(s + t + j)$ .

Ας υποθέσουμε ότι το  $D = \{R_1, \dots, R_n\}$  είναι ένα σχήμα βάσης δεδομένων. Τα ερωτήματα της σχεσιακής άλγεβρας με προβολή και φυσική συνένωση πάνω στο  $D$  ορίζονται επαγωγικά:

- κάθε  $R_i$  στο  $D$  αποτελεί ερώτημα της σχεσιακής άλγεβρας με  $\text{sort } \text{sort}(R_i)$ ,
- αν τα  $Q_1, Q_2$  είναι ερωτήματα της σχεσιακής άλγεβρας με  $\text{sort } S_1$  και  $S_2$  αντίστοιχα, τότε
  - το  $\pi_X(Q_1)$ ,  $X \subseteq S_1$  είναι ένα ερώτημα της σχεσιακής άλγεβρας με  $\text{sort } X$  και
  - το  $Q_1 \bowtie Q_2$  ένα ερώτημα της σχεσιακής άλγεβρας με  $\text{sort } S_1 \cup S_2$ .

Κάθε ερώτημα της σχεσιακής άλγεβρας  $Q$  πάνω στο  $D$ , με  $\text{sort } S$ , ορίζει μία απεικόνιση από τις βάσεις δεδομένων πάνω στο  $D$  στις  $S$ -σχέσεις. Πιο συγκεκριμένα, ας υποθέσουμε ότι η  $d = \{r_1, \dots, r_n\}$  είναι μία βάση δεδομένων πάνω στο  $D$ . Η έξοδος (output) του  $Q$  στην  $d$ , δηλαδή το αποτέλεσμα της αποτίμησης του  $Q$  στην  $d$ , συμβολίζεται με  $Q(d)$  και είναι μία  $S$ -σχέση που υπολογίζεται επαγωγικά ως εξής:



- αν  $Q = R_i$  τότε  $Q(d) = r_i$
- αν  $Q = \pi_X(Q_1)$  τότε  $Q(d) = \pi_X(Q_1(d))$
- αν  $Q = Q_1 \bowtie Q_2$ , τότε  $Q(d) = Q_1(d) \bowtie Q_2(d)$ .

Το πρόβλημα της αποτίμησης ενός ερωτήματος  $Q$ , για μία δεδομένη βάση δεδομένων  $d$ , ορίζεται ως το πρόβλημα του υπολογισμού του  $Q(d)$ . Το πρόβλημα ελέγχου μοντέλου για το ερώτημα  $Q$ , για δεδομένη βάση δεδομένων  $d$ , είναι το πρόβλημα απόφασης για το αν ισχύει ότι  $Q(d) \neq \emptyset$ .

**Ακυκλικά συζευκτικά ερωτήματα – Ο αλγόριθμος του Yannakakis.** Προτού παρουσιάσουμε αναλυτικά τα βήματα του αλγορίθμου, παραθέτουμε κάποιους χρήσιμους ορισμούς.

Ας υποθέσουμε ότι το  $I$  είναι ένα σύνολο δεικτών (indices) και ότι τα  $Y_i$  είναι σύνολα γνωρισμάτων με δείκτες από το σύνολο  $I$ . Η ένωση των συνόλων  $Y_i$  για όλα τα  $i \in I$  συμβολίζεται με  $Y_I$ , δηλαδή  $Y_I = \bigcup_{i \in I} Y_i$ .

Έστω  $d = \{r_1, \dots, r_m\}$  και  $d' = \{r'_1, \dots, r'_m\}$  δύο βάσεις δεδομένων σχήματος  $D$ . Λέμε ότι η  $d'$  είναι υποστιγμιότυπο της  $d$ , και το συμβολίζουμε με  $d' \subseteq d$ , αν για κάθε  $i = 1, \dots, m$ , ισχύει  $r'_i \subseteq r_i$ . Οι  $d$  και  $d'$  ονομάζονται *ισοδύναμες* αν

$$r_1 \bowtie \dots \bowtie r_m = r'_1 \bowtie \dots \bowtie r'_m.$$

Τέλος, λέμε ότι η  $d$  είναι *καθολικά συνεπής* (globally consistent) αν ισχύει

$$\pi_{Y_i}(r_1 \bowtie \dots \bowtie r_m) = r_i$$

για κάθε  $i = 1, \dots, m$ .

Θεωρούμε ότι το  $D = \{R_1, \dots, R_m\}$  είναι ένα σχήμα βάσης δεδομένων με  $\text{sort } Y = \{Y_1, \dots, Y_m\}$ , όπου  $Y_i \neq Y_j$  για κάθε  $i, j$ . Το  $D$  ονομάζεται *ακυκλικό* αν κάθε ένα από τα στοιχεία του  $Y$  μπορεί να αντιστοιχιστεί στους κόμβους  $1, \dots, m$  ενός δέντρου με τέτοιο τρόπο ώστε, για κάθε γνώρισμα  $A$  που εμφανίζεται στο  $Y$ , οι κόμβοι του δέντρου που περιέχουν το  $A$  να σχηματίζουν ένα υποδέντρο. Το δέντρο που προκύπτει ονομάζεται δέντρο συντακτικής ανάλυσης ή συντακτικό δέντρο (parse tree).

Τυπικά, ονομάζουμε *συντακτικό δέντρο του σχήματος  $D$*  κάθε  $Y$ -χρωματισμένο δέντρο

$$\mathcal{P} = (t, \alpha)$$

που έχει

- υποκείμενο δέντρο  $t = (T, s)$ , όπου  $T = \{1, \dots, m\}$  και  $s : T \rightarrow \mathcal{P}(T)$ , και
- συνάρτηση χρωματισμού  $\alpha : T \rightarrow Y$  που αντιστοιχεί το  $i$  στο  $Y_i$  έτσι ώστε να ικανοποιείται η παρακάτω ιδιότητα: (\*) για κάθε  $i, j \in T$  τέτοια ώστε το  $j$  να είναι παιδί του  $i$  στο  $t$ , τα γνωρίσματα του  $Y_{T_j} \setminus Y_i$  εμφανίζονται μόνο στο  $Y_{T_j}$  και σε κανένα άλλο στοιχείο του  $Y$ , δηλαδή ισχύει ότι  $Y_{T_j} \cap Y_i \subseteq Y_{T_j}$ .<sup>1</sup>

<sup>1</sup>Θυμίζουμε ότι το  $T_j$  συμβολίζει το σύνολο κόμβων του πλήρους υποδέντρου του  $(t, \alpha)$  που έχει ρίζα τον κόμβο  $j$ .

Αξίζει να σημειωθεί ότι το  $\mathcal{P}$  δεν είναι παρά μία συγκεκριμένη δεντροδιάσπαση του υπεργράφου (hypergraph)  $\mathcal{H}_D = (Y_T, Y)$ .

Ένα συζευκτικό ερώτημα πάνω στο  $D$  είναι ένα ερώτημα της μορφής

$$Q = \pi_X(R_1 \bowtie \cdots \bowtie R_m).$$

Το  $Q$  ονομάζεται ακυκλικό αν το  $D$  είναι ακυκλικό. Το συντακτικό δέντρο του  $Q$  είναι το συντακτικό δέντρο του  $D$ .

Ο αλγόριθμος του Γιαννακάκης είναι ένας αποδοτικός αλγόριθμος που προσφέρει μία βέλτιστη λύση στο πρόβλημα αποτίμησης ενός ακυκλικού συζευκτικού ερωτήματος  $Q$  πάνω στο  $D$ . Πιο συγκεκριμένα, με είσοδο μία βάση δεδομένων  $d = \{r_1, \dots, r_m\}$  σχήματος  $D$ , ο αλγόριθμος του Γιαννακάκης υπολογίζει το τελικό αποτέλεσμα

$$Q(d) = \pi_X(r_1 \bowtie \cdots \bowtie r_m)$$

μετασχηματίζοντας τη  $d$  σε μία ισοδύναμη βάση δεδομένων  $d' = \{r'_1, \dots, r'_m\}$  η οποία είναι καθολικά συνεπής. Αν η  $d'$  είναι καθολικά συνεπής, τότε κάθε σχέση  $r'_i$  της  $d'$  αποτελείται ακριβώς από εκείνες τις πλειάδες που συμμετέχουν στον υπολογισμό της συνένωσης  $r'_1 \bowtie \cdots \bowtie r'_m = r_1 \bowtie \cdots \bowtie r_m$ . Επομένως, η συνένωση των σχέσεων  $r_i$  της  $d$  –και κατά συνέπεια το  $Q(d)$ – υπολογίζονται με το ελάχιστο κόστος. Αναλυτικότερα, ο αλγόριθμος του Γιαννακάκης υπολογίζει το  $Q(d)$  σε δύο βήματα, I και II, τα οποία περιγράφονται παρακάτω για την περίπτωση που το συντακτικό δέντρο  $\mathcal{P} = (t, \alpha)$  του  $Q$  είναι δυαδικό.

**Βήμα I.** Υπολογισμός μιας καθολικά συνεπούς βάσης δεδομένων  $s$  σχήματος  $D$  που είναι ισοδύναμη με τη βάση δεδομένων  $d$  που αποτελεί την είσοδο του αλγορίθμου. Ο υπολογισμός πραγματοποιείται σε δύο φάσεις οι οποίες αντιστοιχούν σε δυο διαδοχικές διασχίσεις του συντακτικού δέντρου  $\mathcal{P}$ .

α. Διάσχιση «από κάτω προς τα πάνω» (bottom-up traversal). Ξεκινώντας από τη βάση δεδομένων εισόδου  $d$ , υπολογίζουμε μία νέα βάση δεδομένων  $p = \{p_1, \dots, p_m\} \subseteq d$  που είναι ισοδύναμη με την  $d$ . Ο σκοπός αυτής της φάσης είναι να διαγραφούν από τις σχέσεις  $r_i$  εκείνες οι πλειάδες που δεν μπορούν να συνενωθούν με τις σχέσεις που αντιστοιχούν σε απογόνους τους ως προς το  $\mathcal{P}$ . Πιο συγκεκριμένα, ξεκινώντας από τα φύλλα και προχωρώντας προς τη ρίζα, για κάθε φύλλο  $i$  του  $t$  έχουμε  $p_i = r_i$ , και για κάθε εσωτερικό κόμβο  $i$  του  $t$  με παιδιά  $i_1, i_2$  υπολογίζουμε την  $Y_i$ -σχέση  $p_i$  ως εξής

$$p_i = (r_i \bowtie \pi_{Y_i \cap Y_{i_1}} p_{i_1}) \bowtie \pi_{Y_i \cap Y_{i_2}} p_{i_2}.$$

Για τις σχέσεις  $p_i$ ,  $i = 1, \dots, m$ , που υπολογίζονται σε αυτή τη φάση ισχύει ότι

$$r_i \supseteq p_i = \pi_{Y_i} \bowtie_{j \in T_i} r_j.$$

Επίσης, αν το  $r$  είναι η ρίζα, τότε

$$p_r = \emptyset \quad \text{ανν} \quad r_1 \bowtie \cdots \bowtie r_m = \emptyset \quad \text{ανν} \quad Q(d) = \emptyset.$$

Επομένως, αν  $p_r = \emptyset$ , τότε ο αλγόριθμος τερματίζει με έξοδο  $\emptyset$ . Το γεγονός αυτό σημαίνει ότι το βήμα αυτό επιλύει άμεσα το πρόβλημα ελέγχου μοντέλου για το  $Q$  και τη συνένωση  $R_1 \bowtie \cdots \bowtie R_m$ .

**β. Διάσχιση «από πάνω προς τα κάτω» (top-down traversal.)** Με είσοδο τη βάση δεδομένων  $p$  που υπολογίστηκε στην προηγούμενη φάση, υπολογίζουμε μία ισοδύναμη βάση δεδομένων  $s = \{s_1, \dots, s_m\} \subseteq p$ . Ο σκοπός αυτής της φάσης είναι να διαγραφούν από τις σχέσεις  $r_i$  εκείνες οι πλειάδες που δεν μπορούν να συνενωθούν με τις σχέσεις που αντιστοιχούν σε προγόνους τους ως προς το  $\mathcal{P}$ . Αρχικά, στη ρίζα  $r$  έχουμε  $s_r = p_r$ , και στη συνέχεια για κάθε κόμβο  $j \in T$  με πατέρα  $i$  –και ενώ η  $s_i$  έχει υπολογιστεί– υπολογίζουμε την  $Y_j$ -σχέση  $s_j$  ως εξής

$$s_j = p_j \bowtie \pi_{Y_j \cap Y_i}(s_i).$$

Το στιγμιότυπο  $s$  είναι ένα υποστιγμιότυπο του  $d$ , ισοδύναμο του  $d$ , το οποίο επιπρόσθετα είναι καθολικά συνεπές, δηλαδή

$$s_i = \pi_{Y_i} \bowtie_{j \in T} r_j.$$

*Πολυπλοκότητα του Βήματος I.* Προφανώς, η πολυπλοκότητα του βήματος I είναι

$$m \cdot O\left(\max_{1 \leq i \leq m} \{\|r_i\|\}\right).$$

**Βήμα II.** Υπολογισμός του  $Q(d)$  μέσω της συνένωσης, με κατεύθυνση από τα φύλλα προς τη ρίζα, του καθολικά συνεπούς στιγμιότυπου βάσης δεδομένων  $s$ . Ας υποθέσουμε ότι  $X \supseteq X_j = X \cap Y_{T_j}$ , και ότι  $Z_j = Y_j \cap Y_i$  όπου το  $i \in T$  είναι ο πατέρας του  $j$  στο  $t$ . Αρχικά, για κάθε φύλλο  $i$ , υπολογίζουμε τη σχέση  $a_i = \pi_{Z_i \cup X_i} s_i$ . Στη συνέχεια, για κάθε εσωτερικό κόμβο  $i$ , που έχει παιδιά  $i_1, i_2$ , υπολογίζουμε

$$a_i = \pi_{Z_i \cup X_i} ((s_i \bowtie a_{i_1}) \bowtie a_{i_2}).$$

Για κάθε  $Z_i \cup X_i$ -σχέση  $a_i$  που υπολογίζεται σε αυτό το βήμα, έχουμε

$$a_i = \pi_{Z_i \cup X_i} \left( \bowtie_{j \in T_i} s_j \right) = \pi_{Z_i \cup X_i} (s_1 \bowtie \cdots \bowtie s_m) = \pi_{Z_i \cup X_i} (r_1 \bowtie \cdots \bowtie r_m).$$

Έτσι, στη ρίζα  $r$  όπου έχουμε  $X_r = X$  και  $Z_r = \emptyset$ , ισχύει τελικά

$$a_r = \pi_X (r_1 \bowtie \cdots \bowtie r_m) = Q(d).$$

*Πολυπλοκότητα του Βήματος II.* Η πολυπλοκότητα του βήματος II αποδεικνύεται ίση με

$$m \cdot O\left(\max_{1 \leq i \leq m} \{\|r_i\|\} \cdot \|Q(d)\|\right).$$

Πράγματι, σε κάθε κόμβο  $i$  το κόστος υπολογισμού της  $a_i$  φράσσεται από το μέγεθος της  $U_i$ -σχέσης

$$b_i = (s_i \bowtie a_{i_1}) \bowtie a_{i_2} = \pi_{U_i} (r_1 \bowtie \cdots \bowtie r_m)$$

όπου το  $U_i$  είναι ένα κατάλληλο σχήμα που περιέχει το  $Z_i \cup X_i$ , και  $\|b_i\| \leq 2 \cdot \|r_i\| \cdot \|Q(d)\|$ .

Όπως θα δούμε στη συνέχεια, για τα στιγμιότυπα που μελετάμε στην παρούσα διατριβή, καταφέρνουμε να έχουμε καλύτερα αποτελέσματα πολυπλοκότητας για το βήμα αυτό. Πιο συγκεκριμένα, αποδεικνύουμε ότι η πολυπλοκότητα ισούται με

$$m \cdot O(\max_{1 \leq i \leq m} \{\|r_i\|\} + \|Q(d)\|)$$

γεγονός που οφείλεται στην ύπαρξη συναρτησιακών εξαρτήσεων που ικανοποιούνται από τις σχέσεις που συναποτελούν τη βάση δεδομένων που είναι η είσοδος μας.

### 2.3.2 Εκφράζοντας το πρόβλημα μέσω ενός ακυκλικού συζευκτικού ερωτήματος $Q_t$

Είμαστε πλέον έτοιμοι να δούμε πώς ο υπολογισμός των εκτελέσεων του  $\text{Assign}_\phi$  μπορεί να εκφραστεί ως ένα ερώτημα της σχεσιακής άλγεβρας. Ανακαλώντας τον Ορισμό 2.2.2, παρατηρούμε ότι ο υπολογισμός μιας εκτέλεσης δεν είναι τίποτα παραπάνω από την κατάλληλη σύνθεση μεταβάσεων. Πιο συγκεκριμένα, η σύνθεση αυτή πραγματοποιείται λαμβάνοντας υπόψη δυο παράγοντες:

- (i) το χρώμα κάθε κόμβου  $n$ , έτσι ώστε ανάμεσα σε όλες τις πιθανές μεταβάσεις να χρησιμοποιούμε μόνο τις έγκυρες μεταβάσεις για τον  $n$  (Ορισμός 2.2.4), και
- (ii) τις καταστάσεις που ανατίθενται στα παιδιά κάθε κόμβου  $n$ , έτσι ώστε κάθε μετάβαση που αναθέτει μία κατάσταση στον  $n$  να συμφωνεί με αυτές.

Η παραπάνω διαδικασία μπορεί να υλοποιηθεί άμεσα αν αντιστοιχήσουμε σε κάθε κόμβο  $n$  μία σχέση που να αποτελείται από τις έγκυρες μεταβάσεις για τον  $n$  (απαίτηση (i) παραπάνω), και υπολογίσουμε τη φυσική συνένωση αυτών των σχέσεων χρησιμοποιώντας κατάλληλα σχήματα σχέσεων έτσι ώστε η παραπάνω απαίτηση (ii) να ικανοποιείται. Έτσι, οδηγούμαστε φυσικά στην έκφραση των εκτελέσεων του  $\text{Assign}_\phi$  πάνω σε ένα δέντρο  $\mathcal{T}$  στη σχεσιακή άλγεβρα: αν το μέγεθος του  $\mathcal{T}$  είναι  $m$ , τότε η έκφραση αυτή αντιστοιχεί σε μία φυσική συνένωση  $m$  σχέσεων (Πρόταση 2.3.1).

Θα ξεκινήσουμε την τυπική παρουσίαση της προσέγγισής μας με τον ορισμό των σχέσεων  $\text{itrans}_n$  που περιέχουν τις έγκυρες μεταβάσεις στον  $n$  (Ορισμός 2.3.1). Ελάχιστα διαφορετικές από τις  $\text{itrans}_n$  είναι οι σχέσεις  $\text{trans}_n$ , που ορίζονται ώστε να μπορέσουμε να εκφράσουμε τις επιτυχημένες εκτελέσεις. Στη συνέχεια, στο Λήμμα 2.3.1 αποδεικνύουμε ότι οι σχέσεις αυτές αποτελούν ένα στιγμιότυπο βάσης δεδομένων πάνω σε ένα ακυκλικό σχήμα βάσης, που κωδικοποιεί το υποκείμενο δέντρο  $t$  του δέντρου εισόδου  $\mathcal{T}$ .

**Ορισμός 2.3.1.** Θεωρούμε ένα  $\Gamma$ -δέντρο  $\mathcal{T} = (t, c)$  και ένα αυτόματο  $\text{Assign}_\phi = (\Gamma, Q \times \{0, 1\}^k, \Delta_0, \Delta, F \times \{0, 1\}^k)$ . Έστω  $\mathcal{G}_t = (\text{Trans}_n)_{n \in \mathcal{T}}$  ένα σχήμα βάσης δεδομένων με  $\text{sort}(Y_n)_{n \in \mathcal{T}}$ , όπου  $Y_n = Q_n E_n$  αν ο  $n$  είναι ένα φύλλο του  $\mathcal{T}$  και  $Y_n = Q_n Q_{n_1} Q_{n_2} E_n$  αν ο  $n$  είναι ένας εσωτερικός κόμβος του  $\mathcal{T}$  με αριστερό παιδί  $n_1$  και δεξί παιδί  $n_2$ . Ορίζουμε τα

στιγμιότυπα βάσεων δεδομένων  $i_{\phi, \mathcal{T}} = (\text{itrans}_n)_{n \in \mathcal{T}}$  και  $g_{\phi, \mathcal{T}} = (\text{trans}_n)_{n \in \mathcal{T}}$  πάνω στο  $\mathcal{G}_t$  ως εξής

- $\text{itrans}_n = \{\langle q, e \rangle \mid qe \in D_0|_{\gamma}\}$  αν το  $n$  είναι φύλλο χρώματος  $\gamma$ ,
- $\text{itrans}_n = \{\langle q, q_1, q_2, e \rangle \mid qq_1q_2e \in D|_{\gamma}\}$  αν το  $n$  είναι εσωτερικός κόμβος χρώματος  $\gamma$ ,
- $\text{trans}_r = \{t \in \text{itrans}_r \mid t[Q_r] \in F\} \subseteq \text{itrans}_r$  για τη ρίζα  $r$  του  $\mathcal{T}$ ,
- $\text{trans}_n = \text{itrans}_n$  για κάθε  $n \neq r$ .

Όταν ο  $\phi$  είναι πρόταση (βλέπε Παρατήρηση 2.2.2), τότε αν ο  $n$  είναι φύλλο χρώματος  $\gamma$ , έχουμε  $Y_n = Q_n$  και  $\text{itrans}_n = \{\langle q \rangle \mid (\gamma, q) \in \Delta_0\}$ . Ενώ αν ο  $n$  είναι εσωτερικός κόμβος χρώματος  $\gamma$ , τότε  $Y_n = Q_n Q_{n_1} Q_{n_2}$  και  $\text{itrans}_n = \{\langle q, q_1, q_2 \rangle \mid (q_1, q_2, \gamma, q) \in \Delta\}$ . Ανάλογα ορίζονται και οι σχέσεις  $\text{trans}_n$ .

**Παράδειγμα 2.3.1.** Ας υποθέσουμε ότι το  $\mathcal{T} = (t, c)$  είναι ένα  $\{\alpha, \beta\}$ -χρωματισμένο δέντρο με τρεις κόμβους  $n_1$  (ρίζα χρώματος  $\alpha$ ),  $n_2$  (αριστερό παιδί χρώματος  $\alpha$ ),  $n_3$  (δεξί παιδί χρώματος  $\beta$ ). Σύμφωνα με τον παραπάνω ορισμό, έχουμε

$$\mathcal{G}_t = \{\text{Trans}_{n_1}[Q_{n_1} Q_{n_2} Q_{n_3} E_{n_1}], \text{Trans}_{n_2}[Q_{n_2} E_{n_2}], \text{Trans}_{n_3}[Q_{n_3}, E_{n_3}]\}.$$

Ας υποθέσουμε τώρα ότι  $\phi(X) = P_{\alpha}(X)$ . Τα σύνολα καταστάσεων  $Q$ ,  $F$  και τα σύνολα  $D_0$ ,  $D$  που αντιστοιχούν στις μεταβάσεις  $\Delta_0, \Delta$  του  $\text{Assign}_{\phi} = (\{\alpha, \beta\}, Q \times \{0, 1\}, \Delta_0, \Delta, F \times \{0, 1\})$  δίνονται στην απόδειξη της Πρότασης 2.4.8 με τη μορφή γεγονότων. Πιο συγκεκριμένα, έχουμε

- $Q = \{q_0, q_a, q_f\}$  και  $F = \{q_a\}$ ,
- $D_0 = \{\alpha q_0 0, \alpha q_a 1, \beta q_0 0, \beta q_f 1\}$ , σύμφωνα με τον Ορισμό 2.2.3 καθώς  
 $\Delta_0 = \{(\alpha, (q_0, 0)), (\alpha, (q_a, 1)), (\beta, (q_0, 0)), (\beta, (q_f, 1))\}$ ,
- $D = \{\alpha q_0 q_0 q_0 0, \alpha q_a q_0 q_0 1, \beta q_0 q_0 q_0 0, \beta q_f q_0 q_0 1, \dots\}$ .

Τα στιγμιότυπα βάσεων δεδομένων  $i_{\phi, \mathcal{T}}$  και  $g_{\phi, \mathcal{T}}$  ορίζονται ως εξής

- $\text{itrans}_{n_2} = \text{trans}_{n_2} = \{\langle q_0, 0 \rangle, \langle q_a, 1 \rangle\}$  ( $n_2$  φύλλο χρώματος  $\alpha$  και  $D_0|_{\alpha} = \{q_0 0, q_a 1\}$ )
- $\text{itrans}_{n_3} = \text{trans}_{n_3} = \{\langle q_0, 0 \rangle, \langle q_f, 1 \rangle\}$  ( $n_3$  φύλλο χρώματος  $\beta$  και  $D_0|_{\beta} = \{q_0 0, q_f 1\}$ )
- $\text{trans}_{n_1} = \{\langle q_a, q_0, q_0, 1 \rangle, \langle q_a, q_a, q_0, 0 \rangle, \langle q_a, q_0, q_a, 0 \rangle\}$
- $\text{itrans}_{n_1} = \text{trans}_{n_1} \cup \{\langle q_0, q_0, q_0, 0 \rangle, \langle q_f, q_0, q_f, 0 \rangle, \langle q_f, q_0, q_f, 1 \rangle, \langle q_f, q_f, q_0, 0 \rangle, \dots\}$ .

**Λήμμα 2.3.1.** Το σχήμα βάσης δεδομένων  $\mathcal{G}_t$  είναι ακυκλικό και το συντακτικό του δέντρο έχει υποκείμενο δέντρο το  $t$ .

**Απόδειξη.** Προφανώς, το δέντρο

$$\mathcal{P}_t = (t, c')$$

με  $c'(n) = Y_n$ , για κάθε  $n$  στο  $T$ , είναι ένα συντακτικό δέντρο του  $\mathcal{G}_t$ . Αρκεί κανείς να παρατηρήσει ότι όταν δύο κόμβοι  $i, j$  του  $(t, c')$  έχουν κοινά γνωρίσματα, δηλαδή είναι τέτοιοι ώστε  $Y_i \cap Y_j \neq \emptyset$ , τότε αυτοί σχετίζονται απαραίτητως με τη σχέση πατέρα–παιδιού. Συνεπώς το  $(t, c')$  ικανοποιεί τετριμμένα τη συνθήκη (\*) του ορισμού των συντακτικών δέντρων στη σελίδα 35.  $\square$

Συνεπάγεται λοιπόν ότι η βάση δεδομένων  $i_{\phi, \mathcal{T}}$  (αντίστοιχα η  $g_{\phi, \mathcal{T}}$ ) μπορεί να ειδωθεί ως ένα δέντρο  $(t, c')$ , όπου το  $t$  συμπίπτει με το υποκείμενο δέντρο του δέντρου εισόδου  $\mathcal{T}$ , και η συνάρτηση χρωματισμού του  $c'$  αντιστοιχεί κάθε κόμβο  $n$  στη σχέση  $\text{itrans}_n$  (αντίστοιχα  $\text{trans}_n$ ).

**Το ακυκλικό συζευκτικό ερώτημα  $\mathcal{Q}_t$ .** Είμαστε τώρα έτοιμοι να ορίσουμε τυπικά τη συνένωση πάνω στο σχήμα  $\mathcal{G}_t$  που υπολογίζει την  $Y_T$ -σχέση με στοιχεία τις εκτελέσεις του  $\text{Assign}_\phi$ . Η προβολή αυτής της συνένωσης πάνω στο  $E_T$  υπολογίζει τα αντίστοιχα διανύσματα αναθέσεων. Αποτιμώντας την ίδια προβολή πάνω στην  $g_{\phi, \mathcal{T}}$  λαμβάνουμε το  $\phi_{\text{enc}}(\mathcal{T})$ .

Θεωρούμε ότι  $Y_T = Q_{n_1} E_{n_1} \cdots Q_{n_m} E_{n_m}$  και  $E_T = E_{n_1} \cdots E_{n_m}$  όπου  $(n_1, \dots, n_m)$  είναι η διάταξη των στοιχείων  $T$  σύμφωνα με την προδιατεταγμένη διάσχιση. Ανάλογα ορίζονται τα  $Y_{T_n}, E_{T_n}$ .

**Πρόταση 2.3.1.** Για κάθε δέντρο  $t$ , υπάρχει ένα ακυκλικό συζευκτικό ερώτημα  $\mathcal{Q}_t$  πάνω στο σχήμα βάσης δεδομένων  $\mathcal{G}_t$  το οποίο, με είσοδο τη βάση δεδομένων  $g_{\phi, \mathcal{T}}$  που περιγράφει έναν MSO τύπο  $\phi(\bar{X})$  και ένα χρωματισμένο δέντρο  $\mathcal{T} = (t, c)$ , υπολογίζει όλες τις κωδικοποιημένες αναθέσεις που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$ .

**Απόδειξη.** Ας υποθέσουμε ότι τα  $\mathcal{R}_t, \mathcal{Q}_t$  είναι τα ακόλουθα ακυκλικά συζευκτικά ερωτήματα πάνω στο σχήμα βάσης δεδομένων  $\mathcal{G}_t$

$$\mathcal{R}_t = \bigotimes_{n \in T} \text{Trans}_n \quad \mathcal{Q}_t = \pi_{E_T} \bigotimes_{n \in T} \text{Trans}_n.$$

Από τον Ορισμό 2.3.1 έπεται ότι το  $\mathcal{R}_t$  αντιστοιχεί τις βάσεις δεδομένων  $i_{\phi, \mathcal{T}}$  και  $g_{\phi, \mathcal{T}}$  αντίστοιχα στις  $Y_T$ -σχέσεις

$$r_{\phi, \mathcal{T}} = \bigotimes_{n \in T} \text{itrans}_n \quad \text{και} \quad s_{\phi, \mathcal{T}} = \bigotimes_{n \in T} \text{trans}_n$$

με στοιχεία τις εκτελέσεις και τις επιτυχημένες εκτελέσεις αντίστοιχα του  $\text{Assign}_\phi$  στο δέντρο  $\mathcal{T}$ . Κατά συνέπεια, το ερώτημα  $\mathcal{Q}_t$  αντιστοιχεί τη βάση δεδομένων  $g_{\phi, \mathcal{T}}$  στην  $E_T$ -σχέση

$$\mathcal{Q}_t(g_{\phi, \mathcal{T}}) = \pi_{E_T} \bigotimes_{n \in T} \text{trans}_n$$

με στοιχεία όλες τις κωδικοποιημένες αναθέσεις που ικανοποιούν τον  $\phi$  στο δέντρο  $\mathcal{T}$ .  $\square$

Αποδείξαμε λοιπόν ότι η η κωδικοποιημένη εκδοχή του προβλήματος αποτίμησης της MSO σε πεπερασμένα δέντρα ανάγεται σε ένα πρόβλημα αποτίμησης ενός ακυκλικού συζευκτικού ερωτήματος. Πιο συγκεκριμένα, η παραπάνω πρόταση αποδεικνύει ότι σε κάθε ζευγάρι  $(\phi, \mathcal{T})$ , όπου το  $\mathcal{T}$  έχει υποκείμενο δέντρο το  $t$ , μπορούμε να αντιστοιχήσουμε ένα ερώτημα  $Q_t$  και μία βάση δεδομένων  $g_{\phi, \mathcal{T}}$  τ.ω.

$$\bar{B} \in \phi(\mathcal{T}) \quad \text{ανν} \quad \varepsilon_{\bar{B}} \in Q_t(g_{\phi, \mathcal{T}})$$

δηλαδή

$$\phi_{\text{enc}}(\mathcal{T}) = Q_t(g_{\phi, \mathcal{T}}).$$

**Παρατήρηση 2.3.1.** Το πρόβλημα ελέγχου μοντέλου για το  $\mathcal{R}_t$  και τη βάση δεδομένων  $g_{\phi, \mathcal{T}}$  επιλύει το πρόβλημα ελέγχου μοντέλου της MSO για την πρόταση  $\phi$  και το δέντρο  $\mathcal{T}$ , με την έννοια ότι το τελευταίο ανάγεται στο πρώτο. Πράγματι, από την Παρατήρηση 2.2.2 έχουμε ότι  $\mathcal{T} \models \phi$  ανν η μοναδική εκτέλεση του  $\text{Assign}_{\phi}$  στο δέντρο  $\mathcal{T}$  είναι επιτυχημένη. Συνεπώς,

$$\mathcal{T} \models \phi \quad \text{ανν} \quad \mathcal{R}_t(g_{\phi, \mathcal{T}}) = s_{\phi, \mathcal{T}} = \bigtimes_{n \in T} \text{trans}_n = \{\langle \rho \rangle\} \neq \emptyset.$$

**Συναρτησιακές εξαρτήσεις.** Μεταφέροντας την ιδιότητα που περιγράφει το Λήμμα 2.2.2 και αφορά στις εκτελέσεις του  $\text{Assign}_{\phi}$  στο περιβάλλον της σχεσιακής άλγεβρας προκύπτουν συναρτησιακές εξαρτήσεις που ικανοποιούνται από την  $s_{\phi, \mathcal{T}}$ .

**Λήμμα 2.3.2.** Η  $Y_T$ -σχέση  $s_{\phi, \mathcal{T}}$  ικανοποιεί τη δομική εξάρτηση

$$E_{T_n} \rightarrow Y_{T_n}, \quad \text{για κάθε } n \in T.$$

**Απόδειξη.** Από τον ορισμό των εκτελέσεων έχουμε ότι οι  $T_n$ -υποεκτελέσεις του  $\text{Assign}_{\phi}$  αποτελούν εκτελέσεις του  $\text{Assign}_{\phi}$  στο υποδέντρο  $T_n$ . Έτσι, καθώς τα στοιχεία της  $s_{\phi, \mathcal{T}}$  είναι (επιτυχημένες) εκτελέσεις του  $\text{Assign}_{\phi}$  στο  $\mathcal{T}$ , τα στοιχεία της προβολής  $\pi_{Y_{T_n}} s_{\phi, \mathcal{T}}$  είναι εκτελέσεις (όχι απαραίτητα επιτυχημένες) του  $\text{Assign}_{\phi}$  στο  $T_n$ , δηλαδή στοιχεία της  $r_{\phi, T_n}$ . Με άλλα λόγια, έχουμε

$$\pi_{Y_{T_n}} s_{\phi, \mathcal{T}} \subseteq r_{\phi, T_n}.$$

Επιπλέον, η  $\pi_{E_{T_n}} s_{\phi, \mathcal{T}}$  αποτελείται από τα διανύσματα αναθέσεων αυτών των εκτελέσεων και άρα προκύπτει άμεσα από το Λήμμα 2.2.2 ότι

$$\pi_{Y_{T_n}} s_{\phi, \mathcal{T}} \models E_{T_n} \rightarrow Y_{T_n}$$

και άρα για κάθε  $n \in T$ , έχουμε  $s_{\phi, \mathcal{T}} \models E_{T_n} \rightarrow Y_{T_n}$ .  $\square$

Αξίζει να σημειωθεί ότι το παραπάνω αποτέλεσμα ήταν αναμενόμενο, λόγω των συναρτησιακών εξαρτήσεων  $E_n \rightarrow Q_n$  και  $Q_{n_1} Q_{n_2} E_n \rightarrow Q_n$  που ικανοποιούνται από τις σχέσεις εισόδου  $\text{trans}_n$  και οι οποίες είναι άμεσο αποτέλεσμα της αντιστοιχίας τους με το ντετερμινιστικό αυτόματο  $A_{\phi}$ . Είναι αυτές οι εξαρτήσεις συνεπάγονται τις εξαρτήσεις  $E_{T_n} \rightarrow Q_{T_n}$  στη σχέση  $s_{\phi, \mathcal{T}} = \bigtimes_{n \in T} \text{trans}_n$ .

### 2.3.3 Αποτίμηση του $Q_t$ με τον αλγόριθμο του Yannakakis

Αποδείξαμε στην προηγούμενη ενότητα ότι υπάρχει ένα ακυκλικό συζευκτικό ερώτημα  $Q_t$  που, με είσοδο την βάση δεδομένων  $g_{\phi, \mathcal{T}}$ , έχει ως έξοδο την  $E_T$ -σχέση με στοιχεία τις κωδικοποιημένες αναθέσεις που ικανοποιούν τον  $\phi(X_1, \dots, X_k)$  στο  $\mathcal{T}$ . Οι προτάσεις που ακολουθούν περιγράφουν τα δυο βήματα του αλγορίθμου του Yannakakis για τον υπολογισμό του  $Q_t(g_{\phi, \mathcal{T}})$  και αναδεικνύουν βασικές ιδιότητες των σχέσεων που υπολογίζονται. Ταυτόχρονα δίνονται οι αντίστοιχες αναλύσεις πολυπλοκότητας, που χρησιμοποιούν τις συναρτησιακές εξαρτήσεις του Λήμματος 2.3.2 για την επίτευξη καλύτερων φραγμάτων.

Πιο συγκεκριμένα, η Πρόταση 2.3.2 δίνει τον μετασχηματισμό δύο φάσεων –σύμφωνα με το Βήμα I του αλγορίθμου του Yannakakis– του στιγμιότυπου  $(\text{trans}_n)_{n \in T}$  των έγκυρων μεταβάσεων του  $\text{Assign}_\phi$ . Μάλιστα, αποδεικνύεται ότι τα στιγμιότυπα που υπολογίζονται στη φάση (α) (αντίστοιχα φάση (β)) αντιστοιχούν στις πιθανές (αντίστοιχα επιτυχημένες) μεταβάσεις του  $\text{Assign}_\phi$  στο  $\mathcal{T}$ . Η Πρόταση 2.3.3 παρουσιάζει το αποτέλεσμα την εφαρμογή του Βήματος II του αλγορίθμου του Yannakakis στο καθολικά συνεπές στιγμιότυπο  $(s\text{-trans}_n)_{n \in T}$  που υπολογίζεται στην Πρόταση 2.3.2: η σχέση που υπολογίζεται τελικά έχεις ως στοιχεία τις κωδικοποιημένες αναθέσεις που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$ .

**Πρόταση 2.3.2.** Σύμφωνα με το Βήμα I του αλγορίθμου του Yannakakis, το στιγμιότυπο βάσης δεδομένων  $g_{\phi, \mathcal{T}} = (\text{trans}_n)_{n \in T}$  πάνω στο  $\mathcal{G}_t$  μετασχηματίζεται σε δύο ισοδύναμα υπο-στιγμιότυπα  $(p\text{-trans}_n)_{n \in T}$  και  $(s\text{-trans}_n)_{n \in T}$  που ικανοποιούν τα εξής

$$p\text{-trans}_n = \pi_{Y_n} \boxtimes_{i \in T} i\text{trans}_i = \pi_{Y_n} r_{\phi, \mathcal{T}} \quad \text{για κάθε } n \neq r \quad (\text{i})$$

$$s\text{-trans}_n = \pi_{Y_n} \boxtimes_{i \in T} \text{trans}_i = \pi_{Y_n} s_{\phi, \mathcal{T}} \quad \text{για κάθε } n \quad (\text{ii})$$

δηλαδή η σχέση  $p\text{-trans}_n$ , για κάθε  $n \neq r$  (αντίστοιχα η  $s\text{-trans}_n$ , για κάθε  $n$ ) αποτελείται από τις πιθανές (αντίστοιχα επιτυχημένες) μεταβάσεις στον  $n$ .

Η πολυπλοκότητα είναι  $O(|T| \cdot |\text{trans}_n|)$ ,  $n$  εσωτερικός κόμβος ή ισοδύναμα  $O(m \cdot s^2 \cdot 2^k)$  όπου το  $m$  είναι το μέγεθος του δέντρου  $\mathcal{T}$  και το  $s$  είναι το μέγεθος του συνόλου  $Q$  των  $\phi$ -καταστάσεων.

**Απόδειξη.** Φάση α. (*bottom-up διάσχιση του  $t$* ). Ο σκοπός αυτής της φάσης είναι να διαγραφούν από τις  $Y_n$ -σχέσεις  $\text{trans}_n$  εκείνες οι πλειάδες που δε μπορούν να συνενωθούν με τις σχέσεις που αντιστοιχούν σε απογόνους τους ως προς το συντακτικό δέντρο  $\mathcal{P}_t = (t, c')$  του  $\mathcal{G}_t$ . Αναλυτικότερα, η φάση (α) συνίσταται στον υπολογισμό του υποστιγμιότυπου  $(p\text{-trans}_n)_{n \in T}$  του  $(\text{trans}_n)_{n \in T}$  με τον εξής τρόπο: αν το  $n \in T$  είναι φύλλο, τότε  $p\text{-trans}_n = \text{trans}_n$  (θυμίζουμε ότι  $Y_n \cap Y_{n_i} = Q_{n_i}$ , για κάθε παιδί  $n_i$  του  $n$  στο  $t$ ). Αν το  $n \in T$  είναι εσωτερικός κόμβος με παιδιά  $n_1, n_2$ , τότε

$$p\text{-trans}_n = (\text{trans}_n \boxtimes \pi_{Q_{n_1}} p\text{-trans}_{n_1}) \boxtimes \pi_{Q_{n_2}} p\text{-trans}_{n_2}.$$

Δηλαδή, για κάθε εσωτερικό κόμβο  $n$ , τα στοιχεία της σχέσης  $p\text{-trans}_n$  είναι εκείνες οι πλειάδες της σχέσης  $\text{trans}_n$  που συμφωνούν –ως προς τα κοινά τους γνωρίσματα  $Q_{n_i}$ ,  $i = 1, 2$ – με



τις σχέσεις  $p\text{-trans}_{n_i}$  που υπολογίζονται στα παιδιά τους. Έχουμε λοιπόν ότι  $\pi_{Q_{n_i}} p\text{-trans}_n \subseteq \pi_{Q_{n_i}} p\text{-trans}_{n_i}$ .

Προφανώς, η (i) δηλώνει ότι, για κάθε  $n \neq r$ , η σχέση  $p\text{-trans}_n$  αποτελείται από τις πιθανές μεταβάσεις στον  $n$  ως προς τις εκτελέσεις του  $\text{Assign}_\phi$  στο  $\mathcal{T}$ . Αξίζει να σημειωθεί εδώ ότι λόγω της ιδιότητας που χαρακτηρίζει τις σχέσεις της φάσης (α) του Βήματος II του αλγορίθμου του Γιαννακάκης έχουμε άμεσα ότι

$$p\text{-trans}_n = \pi_{Y_n} \bowtie_{i \in T_n} \text{trans}_i \quad (*)$$

που αποδεικνύει το ζητούμενο μόνο ως προς τις  $\mathcal{T}_n$ -υποεκτελέσεις του  $\text{Assign}_\phi$  (και όχι τις εκτελέσεις). Συνεπώς η (i) εκφράζει μία ιδιότητα πιο ισχυρή από αυτή που εκφράζει η (\*).

Η απόδειξη της ισότητας (i), που δίνεται παρακάτω, είναι αρκετά τεχνική και βασίζεται στα εξής δυο σημεία:

- η εφαρμογή του Βήματος I του αλγορίθμου του Γιαννακάκης πάνω στο στιγμιότυπο  $(\text{itrans}_n)_{n \in T}$  αποδίδει στο τέλος της φάσης (α) ένα καθολικά συνεπές υποστιγμιότυπο του  $(\text{itrans}_n)_{n \in T}$  το οποίο ονομάζουμε  $(p\text{-itrans}_n)_{n \in T}$
- για τις σχέσεις  $p\text{-itrans}_n$  με  $n \neq r$  έχουμε ότι  $p\text{-itrans}_n = p\text{-trans}_n$ , καθώς  $\text{trans}_n = \text{itrans}_n$ , για κάθε  $n \neq r$  (από τον Ορισμό 2.3.1).

Πράγματι, η  $(p\text{-itrans}_n)_{n \in T}$  είναι ένα καθολικά συνεπές στιγμιότυπο καθώς, όπως δείχνουμε παρακάτω, ικανοποιεί την ισότητα

$$\pi_{Q_{n_i}} p\text{-itrans}_n = \pi_{Q_{n_i}} p\text{-itrans}_{n_i}, \quad i = 1, 2 \quad (**)$$

η οποία κανονικά θα ίσχυε μετά το τέλος της φάσης (β) του Βήματος II του αλγορίθμου του Γιαννακάκης. Συνεπώς η φάση (β) δεν αλλάζει καθόλου τις σχέσεις  $p\text{-itrans}_n$ , δηλαδή  $s\text{-itrans}_n = p\text{-itrans}_n$  για κάθε  $n$ . Η ισότητα (\*\*) ισχύει λόγω των εξής ιδιαιτεροτήτων των δεδομένων μας: για κάθε εσωτερικό κόμβο  $n$  έχουμε  $(Y_n \cap Y_{n_1}) \cap (Y_n \cap Y_{n_2}) = Q_{n_1} \cap Q_{n_2} = \emptyset$  και  $\pi_{Q_{n_1} Q_{n_2}} \text{itrans}_n = Q \times Q = \text{dom}(Q_{n_1}) \times \text{dom}(Q_{n_2})$ . Αυτό σημαίνει ότι

$$\pi_{Q_{n_1}} p\text{-itrans}_{n_1} \subseteq \pi_{Q_{n_1}} \text{itrans}_n = Q$$

και

$$\pi_{Q_{n_1} Q_{n_2}} (\text{itrans}_n \bowtie \pi_{Q_{n_1}} p\text{-itrans}_{n_1}) = \pi_{Q_{n_1}} p\text{-itrans}_{n_1} \times Q.$$

**Φάση β. (top-down διάσχιση του  $t$ ).** Σκοπός αυτής της φάσης είναι η διαγραφή από τις σχέσεις  $p\text{-trans}_n$  εκείνων των πλειάδων που δεν μπορούν να συνενωθούν με τις σχέσεις που αντιστοιχούν σε προγόνους τους ως προς το συντακτικό δέντρο. Πιο συγκεκριμένα, η φάση (β) συνίσταται στον υπολογισμό του υποστιγμιότυπου  $(s\text{-trans}_n)_{n \in T}$  του  $(p\text{-trans}_n)_{n \in T}$  που πραγματοποιείται ως εξής: για τη ρίζα  $r$  έχουμε  $s\text{-trans}_r = p\text{-trans}_r$ , ενώ για κάθε κόμβο  $n_i \in T$ ,  $i = 1, 2$  που είναι παιδί του  $n$ , υπολογίζουμε την  $Y_{n_i}$ -σχέση

$$s\text{-trans}_{n_i} = p\text{-trans}_{n_i} \bowtie \pi_{Q_{n_i}} s\text{-trans}_n.$$

Ισχύει από τον αλγόριθμο του Γιαννακάκης ότι οι σχέσεις  $(s\text{-trans}_n)_{n \in T}$  αποτελούν ένα καθολικά συνεπές στιγμιότυπο πάνω στο  $\mathcal{G}_t$  ισοδύναμο με το  $(\text{trans}_n)_{n \in T}$ . Δηλαδή ισχύει η ισότητα (ii), γεγονός που αποδεικνύει ότι κάθε σχέση  $s\text{-trans}_n$  αποτελείται από τις επιτυχημένες μεταβάσεις στον κόμβο  $n$  ως προς τις εκτελέσεις του  $\text{Assign}_\phi$  στο δέντρο εισόδου  $\mathcal{T}$ .

*Πολυπλοκότητα.* Από τον Ορισμό 2.2.3 έπεται ότι  $\|\text{itrans}_n\| = 2^{k+1}$  για  $n$  φύλλο και  $\|\text{itrans}_n\| = s^2 2^{k+2}$  για  $n$  εσωτερικό κόμβο,  $s = |Q|$ . Επιπλέον

$$s\text{-trans}_n \subseteq p\text{-trans}_n \subseteq \text{trans}_n \subseteq \text{itrans}_n.$$

Έτσι, κατά τη φάση (α) και (β), η πολυπλοκότητα σε κάθε κόμβο είναι πάντα  $O(s^2 \cdot 2^k)$ , διαμορφώνοντας τη συνολική πολυπλοκότητα  $m \cdot O(s^2 \cdot 2^k)$ ,  $m = |T|$ .  $\square$

**Πρόταση 2.3.3.** Δεδομένου του καθολικά συνεπούς στιγμιότυπου  $(s\text{-trans}_n)_{n \in T}$ , που είναι ισοδύναμο με το στιγμιότυπο εισόδου  $g_{\phi, \mathcal{T}} = (\text{trans}_n)_{n \in T}$  πάνω στο  $\mathcal{G}_t$ , το Βήμα II του αλγορίθμου του Γιαννακάκης υπολογίζει την έξοδο του ερωτήματος  $Q_t$  στην  $g_{\phi, \mathcal{T}}$ , υπολογίζοντας επαγωγικά (με κατεύθυνση από τα φύλλα προς τη ρίζα) σχέσεις  $a\text{-trans}_n$  έτσι ώστε

$$a\text{-trans}_n = \pi_{Q_n E_{T_n}} \left( \bigotimes_{i \in T} \text{trans}_i \right) \text{ για } n \neq r,$$

δηλαδή σχέσεις  $a\text{-trans}_n$  που αντιστοιχούν σε περιορισμούς, στα υποδέντρα  $T_n$ , των κωδικοποιημένων αναθέσεων που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$ . Τελικά στη ρίζα  $r$ , έχουμε

$$a\text{-trans}_r = \pi_{E_T} \left( \bigotimes_{i \in T} \text{trans}_i \right) = Q_t(g_{\phi, \mathcal{T}}).$$

Η πολυπλοκότητα αποτίμησης του ερωτήματος είναι  $O(\|Q_t(g_{\phi, \mathcal{T}})\|)$ .

**Απόδειξη.** Εφαρμόζουμε το Βήμα II του αλγορίθμου του Γιαννακάκης. Σύμφωνα με αυτό, ξεκινώντας από τα φύλλα με κατεύθυνση προς τη ρίζα, υπολογίζουμε σε κάθε κόμβο  $n$  μία  $Q_n E_{T_n}$ -σχέση  $a\text{-trans}_n$  ως εξής: για κάθε φύλλο  $n$ , έχουμε  $a\text{-trans}_n = s\text{-trans}_n$ , ενώ αν το  $n$  είναι εσωτερικός κόμβος διαφορετικός από τη ρίζα  $r$ , έχουμε

$$a\text{-trans}_n = \pi_{Q_n E_{T_n}} \left( (s\text{-trans}_n \otimes a\text{-trans}_{n_1}) \otimes a\text{-trans}_{n_2} \right).$$

Στη ρίζα υπολογίζουμε την ίδια συνένωση αλλά προβάλλουμε μόνο στο  $E_T$ . Ουσιαστικά αυτό που κάνουμε σε κάθε κόμβο  $n$  είναι: i) προβάλλουμε στο γνώρισμα  $Q_n$ , το οποίο αποτελεί το μοναδικό κοινό σημείο ανάμεσα στα γνωρίσματα του  $T_n$  και του  $T \setminus T_n$ , καθώς ισχύει  $Y_{T_n} \cap Y_{T \setminus T_n} = Q_n$  και ii) προβάλλουμε στο  $E_{T_n}$  που αποτελεί τον περιορισμό στο  $T_n$  του διανύσματος προβολής  $E_T$ .

Από τις ισότητες που ισχύουν για τις σχέσεις που υπολογίζονται από το Βήμα II του αλγορίθμου του Γιαννακάκης, προκύπτει άμεσα ότι οι  $a\text{-trans}_n$  αντιστοιχούν στους περιορισμούς στο  $T_n$  των κωδικοποιημένων αναθέσεων που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$  καθώς έχουμε

$$a\text{-trans}_n = \pi_{Q_n E_{T_n}} \left( \bigotimes_{i \in T} \text{trans}_i \right).$$

Τελικά,

$$\text{a-trans}_r = \pi_{E_T} \left( \bigotimes_{i \in T} \text{s-trans}_i \right) = \pi_{E_T} \left( \bigotimes_{i \in T} \text{trans}_i \right) = \mathcal{Q}_t(g_{\phi, \mathcal{T}}).$$

Πολυπλοκότητα. Θέτουμε  $\text{j-trans}_n = \text{s-trans}_n$  όταν το  $n$  είναι φύλλο και

$$\text{j-trans}_n = (\text{s-trans}_n \otimes \text{a-trans}_{n_1}) \otimes \text{a-trans}_{n_2}$$

όταν το  $n$  είναι εσωτερικός κόμβος με παιδιά  $n_1, n_2$ . Θα αποδείξουμε αρχικά ότι για κάθε εσωτερικό κόμβο  $n \in T$  και κάθε  $n_j \in T_n \setminus \{n\}$ , έχουμε

$$\|\text{j-trans}_{n_j}\| < \|\text{j-trans}_n\|. \quad (*)$$

Γνωρίζουμε ότι  $\text{j-trans}_n = \pi_{U_n} s_{\phi, \mathcal{T}}$ , όπου  $U_n = Q_n E_{T_n}$  ή  $U_n = Q_n Q_{n_1} Q_{n_2} E_{T_n}$ , και από το Λήμμα 2.3.2 θυμίζουμε ότι αυτό αποδεικνύει ότι  $s_{\phi, \mathcal{T}} \models E_{T_n} \rightarrow Y_{T_n}$ — έχουμε ότι αν  $U_n \subseteq Y_{T_n}$ , τότε  $\pi_{U_n} s_{\phi, \mathcal{T}} \models E_{T_n} \rightarrow U_n$ . Επομένως  $|\text{j-trans}_n| = |\pi_{E_{T_n}} s_{\phi, \mathcal{T}}|$  και

$$\|\text{j-trans}_n\| = \begin{cases} \|\pi_{E_{T_n}} s_{\phi, \mathcal{T}}\| + |\pi_{E_{T_n}} s_{\phi, \mathcal{T}}|, & n \text{ φύλλο} \\ \|\pi_{E_{T_n}} s_{\phi, \mathcal{T}}\| + 3|\pi_{E_{T_n}} s_{\phi, \mathcal{T}}|, & n \text{ εσωτερικός κόμβος.} \end{cases}$$

Επειδή για  $n_j \in T_n \setminus \{n\}$  έχουμε  $E_{T_{n_j}} \subset E_{T_n}$ , τελικά αποδεικνύουμε την (\*).

Τώρα, χρησιμοποιώντας την (\*), αποδεικνύουμε με επαγωγή ότι το συνολικό κόστος  $h_n$  για την αποτίμηση της σχέσης  $\text{a-trans}_n$ , δηλαδή το κόστος υπολογισμού της  $\text{a-trans}_n$ , θεωρώντας τις σχέσεις  $(\text{s-trans}_n)_{n \in T}$  ως είσοδο, είναι το πολύ  $c_n \cdot \|\text{j-trans}_n\|$  για κάποια σταθερά  $c_n$ , δηλαδή  $h_n = O(\|\text{j-trans}_n\|)$  (\*\*). Η επαγωγική απόδειξη είναι:

- *Βάση*: για κάθε φύλλο  $n$ , έχουμε  $\text{a-trans}_n = \text{s-trans}_n = \text{j-trans}_n$  και επομένως η (\*\*) ισχύει τετριμμένα.
- *Επαγωγικό βήμα*: έστω ότι ο  $n$  είναι ένας εσωτερικός κόμβος. Αφού η  $\text{a-trans}_n$  είναι μία προβολή της  $\text{j-trans}_n$ , έχουμε άμεσα

$$h_n = o_n + h_{n_1} + h_{n_2}$$

όπου το  $o_n \leq c \cdot \|\text{j-trans}_n\|$  αντιστοιχεί στο κόστος υπολογισμού της συνένωσης  $\text{j-trans}_n$  δεδομένων των  $\text{a-trans}_{n_1}, \text{a-trans}_{n_2}$  (λόγω του ότι προφανώς έχουμε  $\|\text{s-trans}_n\|, \|\text{a-trans}_{n_1}\|, \|\text{a-trans}_{n_2}\| < \|\text{j-trans}_n\|$ ). Συνεπώς, από την επαγωγική υπόθεση και χρησιμοποιώντας την (\*), έχουμε ότι

$$h_n \leq c \cdot \|\text{j-trans}_n\| + c_{n_1} \cdot \|\text{j-trans}_{n_1}\| + c_{n_2} \cdot \|\text{j-trans}_{n_2}\| \leq c_n \cdot \|\text{j-trans}_n\|$$

για  $c_n = c + c_{n_1} + c_{n_2}$ .

Άρα στη ρίζα  $r$  το κόστος  $h_r$  για τον υπολογισμό του  $\mathcal{Q}_t(g_{\phi, \mathcal{T}}) = \text{a-trans}_r$ , με δεδομένη την  $(\text{s-trans}_n)_{n \in T}$ , είναι το πολύ  $c_r \cdot \|\text{j-trans}_r\|$ , δηλαδή

$$h_r = O(\|\text{j-trans}_r\|) = O(\|\mathcal{Q}_t(g_{\phi, \mathcal{T}})\|).$$

□

**Παράδειγμα 2.3.2.** (συνέχεια) Ας υποθέσουμε ότι τα  $\phi(X_1, X_2)$  και  $\mathcal{T}$  είναι όπως στο Παράδειγμα 2.2.3. Ο αλγόριθμος του Yannakakis, εφαρμοσμένος στο ακυκλικό συζευκτικό ερώτημα και τη βάση δεδομένων που αντιστοιχούν στα  $\phi$  και  $\mathcal{T}$ , υπολογίζει τις κωδικοποιημένες αναθέσεις  $\varepsilon_1, \varepsilon_2$  που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$  στην  $E_{n_1}E_{n_2}E_{n_3}$ -σχέση

$$\text{a-trans}_{n_1} = \{\langle 01, 10, 01 \rangle, \langle 00, 10, 01 \rangle\}.$$

Πιο συγκεκριμένα, το Βήμα II του αλγορίθμου του Yannakakis υπολογίζει την  $\text{a-trans}_{n_1}$  υπολογίζοντας επαγωγικά περιορισμούς των κωδικοποιημένων αναθέσεων που ικανοποιούν τον δεδομένο τύπο, δηλαδή την  $Q_{n_2}E_{n_2}$ -σχέση  $\text{a-trans}_{n_2} = \{\langle q_2, 10 \rangle, \langle s_2, 10 \rangle\}$  και την  $Q_{n_3}E_{n_3}$ -σχέση  $\text{a-trans}_{n_3} = \{\langle q_3, 01 \rangle, \langle s_3, 01 \rangle\}$  που αντιστοιχούν στις υποεκτελέσεις  $\rho_1, \rho_2$  του Παραδείγματος 2.2.3.

**Πολυπλοκότητα της αποτίμησης.** Ο προσδιορισμός της συνολικής πολυπλοκότητας (combined complexity) του αλγορίθμου του Yannakakis για τον υπολογισμό του  $Q_t$  είναι άμεση συνέπεια των παραπάνω αποδείξεων πολυπλοκότητας.

**Πόρισμα 2.3.1.** Η συνολική πολυπλοκότητα του αλγορίθμου του Yannakakis για τον υπολογισμό του ακυκλικού συζευκτικού ερωτήματος  $Q_t$  με είσοδο τη βάση δεδομένων  $g_{\phi, \mathcal{T}} = (\text{trans}_n)_{n \in \mathcal{T}}$ , όπου  $\mathcal{T} = (t, c)$ , είναι

$$O(|\mathcal{T}| \cdot |\text{trans}_i| + \|Q_t(g_{\phi, \mathcal{T}})\|),$$

όπου το  $i$  είναι ένας εσωτερικός κόμβος του  $\mathcal{T}$ .

**Σχέση αλγορίθμου του Yannakakis & FFG αλγορίθμου.** Ένα σημαντικό πόρισμα του υπολογισμού του  $Q_t$  με τον αλγόριθμο του Yannakakis είναι η σχέση που προκύπτει με τον FFG αλγόριθμο. Αναλυτικότερη περιγραφή του τελευταίου, δίνεται στο Παράρτημα Β'.

**Πόρισμα 2.3.2.** Ο αλγόριθμος του Yannakakis για τον υπολογισμό του ακυκλικού συζευκτικού ερωτήματος  $Q_t$  με είσοδο τη βάση δεδομένων  $g_{\phi, \mathcal{T}} = (\text{trans}_n)_{n \in \mathcal{T}}$ , όπου  $\mathcal{T} = (t, c)$ , συμπίπτει με τον FFG αλγόριθμο με είσοδο τα  $A_\phi$  και  $\mathcal{T}$ , με την έννοια ότι και οι δύο αλγόριθμοι πραγματοποιούν τους ίδιους υπολογισμούς και παράγουν την ίδια έξοδο (κωδικοποιημένη διαφορετικά από τον κάθε αλγόριθμο).

**Απόδειξη.** Περιγράφουμε αρχικά τους δύο αλγορίθμους και στη συνέχεια κάνουμε μια αναλυτική έκθεση των ομοιοτήτων τους και των μικρών τους διαφορών.

Ξεκινάμε περιγράφοντας τα βήματα του FFG αλγορίθμου [FFG02] για τον υπολογισμό, με είσοδο τα  $A_\phi$  και  $\mathcal{T}$ , των αναθέσεων που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$ . Τα βήματα, που προσαρμόζονται στο τρέχων πλαίσιο, δηλαδή δίνονται ως προς το  $\text{Assign}_\phi$ , είναι τα εξής:

- (i) bottom-up υπολογισμός των πιθανών καταστάσεων του  $\text{Assign}_\phi$  σε κάθε κόμβο,
- (ii) top-down υπολογισμός των επιτυχημένων καταστάσεων του  $\text{Assign}_\phi$  σε κάθε κόμβο,

(ii) bottom-up υπολογισμός των αναθέσεων πραγματοποιώντας κατά-συνιστώσες ενώσεις των περιορισμών, στα υποδέντρα, των αναθέσεων που ικανοποιούν τον  $\phi$ .

Αποδείξαμε στην Πρόταση 2.3.2 και στην Πρόταση 2.3.3 ότι ο αλγόριθμος του Yannakakis εφαρμοσμένος στο ερώτημα  $Q_t$ , με είσοδο τη βάση δεδομένων που αποτελείται από τις έγκυρες μεταβάσεις του αυτομάτου  $\text{Assign}_\phi$  σε ένα δέντρο  $\mathcal{T} = (t, c)$ , διασχίζει το δέντρο  $t$  τρεις φορές ως εξής:

- (i) bottom-up υπολογισμός των πιθανών μεταβάσεων του  $\text{Assign}_\phi$  σε κάθε κόμβο,
- (ii) top-down υπολογισμός των επιτυχημένων μεταβάσεων του  $\text{Assign}_\phi$  σε κάθε κόμβο,
- (iii) bottom-up υπολογισμός των κωδικοποιημένων αναθέσεων πραγματοποιώντας συνενώσεις των περιορισμών, στα υποδέντρα, των κωδικοποιημένων αναθέσεων που ικανοποιούν τον  $\phi$ .

Παρόλο που ο FFG αλγόριθμος υπολογίζει καταστάσεις, ενώ το αλγόριθμος του Yannakakis υπολογίζει μεταβάσεις, και οι δυο αλγόριθμοι πραγματοποιούν στα δύο πρώτα τους βήματα ακριβώς τους ίδιους υπολογισμούς. Στο τρίτο βήμα, ο FFG αλγόριθμος υπολογίζει τις επιτυχημένες μεταβάσεις —τις οποίες έχει ήδη υπολογίσει ο αλγόριθμος του Yannakakis στο βήμα (ii)— χρησιμοποιώντας τις επιτυχημένες καταστάσεις που έχει ήδη υπολογίσει, και τις συνθέτει επαγωγικά με περιορισμούς των αναθέσεων που ικανοποιούν τον τύπο. Η σύνθεση αυτή γίνεται με κατεύθυνση από τα φύλλα προς τη ρίζα, πραγματοποιώντας τους ίδιους υπολογισμούς με τον αλγόριθμο του Yannakakis, με τη μόνη διαφορά ότι ο αλγόριθμος του Yannakakis συνθέτει περιορισμούς κωδικοποιημένων αναθέσεων που ικανοποιούν τον τύπο.  $\square$

### 2.3.4 Αποτελέσματα πολυπλοκότητας & σύνδεση με τον FFG αλγόριθμο

Στην ενότητα αυτή, παρουσιάζονται ενδιαφέροντα πορίσματα της επίλυσης μέσω της σχεσιακής άλγεβρας για το πρόβλημα της αποτίμησης της MSO σε δέντρα.

**Η πολυπλοκότητα του προβλήματος αποτίμησης της MSO σε δέντρα.** Παραθέτουμε αρχικά μία απόδειξη —μέσω της σχεσιακής άλγεβρας— για το πρόβλημα της αποτίμησης της MSO, πόρισμα της οποίας αποτελεί το αντίστοιχο αποτέλεσμα για το πρόβλημα ελέγχου μοντέλου που ακολουθεί στη συνέχεια.

**Θεώρημα 2.3.1.** 1. Η συνολική πολυπλοκότητα του προβλήματος αποτίμησης για έναν MSO τύπο  $\phi(X_1, \dots, X_k)$  σε ένα χρωματισμένο δέντρο  $\mathcal{T}$  είναι

$$O(|T| \cdot f(\phi) + \text{size}(\phi(\mathcal{T})))$$

όπου η  $f$  είναι μία υπολογίσιμη συνάρτηση του μεγέθους του  $\phi$  και  $\text{size}(\phi(\mathcal{T})) = |T| \cdot |\phi(\mathcal{T})|$ .

2. Η  $f$  είναι μία μη-στοιχειώδης συνάρτηση του μεγέθους του τύπου  $\phi$  που αντιστοιχεί στο μέγεθος του  $\text{Assign}_\phi$ .

**Απόδειξη.** 1. Η επίλυση του προβλήματος αποτίμησης της MSO σε χρωματισμένα δέντρα αποτελείται από τα εξής βήματα:

- (i) κατασκευή του αυτομάτου  $\text{Assign}_\phi$  και ορισμός της βάσης δεδομένων  $g_{\phi, \mathcal{T}}$ , με πολυπλοκότητα μία υπολογίσιμη συνάρτηση του μεγέθους του τύπου  $\phi$  που αντιστοιχεί στο μέγεθος του  $\text{Assign}_\phi$ ,
- (ii) ορισμός του ακυκλικού συζευκτικού ερωτήματος  $Q_t$  και κατασκευή του αντίστοιχου συντακτικού δέντρου, με πολυπλοκότητα γραμμική στο  $|T|$ ,
- (iii) αποτίμηση, με τον αλγόριθμο του Yannakakis, του  $Q_t(g_{\phi, \mathcal{T}})$  με στοιχεία τις κωδικοποιημένες αναθέσεις που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$ : από το Πρόρισμα 2.3.1, η πολυπλοκότητα είναι  $O(|T| \cdot |Q|^2 \cdot 2^k + \|Q_t(g_{\phi, \mathcal{T}})\|)$  όπου  $Q$  είναι το σύνολο των  $\phi$ -καταστάσεων,
- (iv) υπολογισμός του  $\phi(\mathcal{T})$  από το  $Q_t(g_{\phi, \mathcal{T}}) = \phi_{\text{enc}}(\mathcal{T})$ , με πολυπλοκότητα γραμμική στο  $\|Q_t(g_{\phi, \mathcal{T}})\| = |T| \cdot |\phi(\mathcal{T})| = \text{size}(\phi(\mathcal{T}))$ .

2. Η υπολογίσιμη συνάρτηση  $f$  είναι μη στοιχειώδης, καθώς το μέγεθος του  $A_\phi$  και συνεπώς και του  $\text{Assign}_\phi$  δεν φράσσεται από καμία στοιχειώδη συνάρτηση [Rei02].  $\square$

Η πολυπλοκότητα που αποδείχθηκε παραπάνω και εκείνη που δίνεται στο [FFG02] είναι ουσιαστικά οι ίδιες, γεγονός που είναι άλλωστε αναμενόμενο καθώς και στις δυο περιπτώσεις πραγματοποιούνται οι ίδιοι υπολογισμοί. Πιο συγκεκριμένα, το πρόβλημα αποτίμησης της MSO σε δέντρα αποδεικνύεται στο [FFG02] να έχει πολυπλοκότητα

$$f(\phi) \cdot (|T| + \|\phi(\mathcal{T})\|)$$

όπου  $\|\phi(\mathcal{T})\| = \sum_{(B_1, \dots, B_k) \in \phi(\mathcal{T})} (\sum_{i=1}^k |B_i|)$ . Η πολυπλοκότητα αυτή επιτυγχάνεται μέσω του ορισμού και της υλοποίησης σύνθετων δομών δεδομένων όπως πίνακες με στοιχεία λίστες, και τελεστών όπως το MERGE. Το βασικό πλεονέκτημα της προσέγγισής μας είναι η μεταφορά της διαίσθησης από γνωστά –και καλά μελετημένα– αλγεβρικά εργαλεία και αλγορίθμους. Συνεπώς, η πολυπλοκότητα που αποδεικνύουμε προκύπτει ιδιαίτερα φυσικά και είναι ιδιαίτερα απλή στην κατανόηση. Αξίζει να σημειωθεί ότι η πολυπλοκότητα αυτή αποτελεί γνωστό κάτω φράγμα λόγω του γεγονότος ότι το μέγεθος του αυτομάτου θα είναι πάντα μη-στοιχειώδης συνάρτηση του μεγέθους του δεδομένου τύπου. Μάλιστα, στο [FG04] αποδεικνύεται ότι, εάν  $P \neq NP$ , δεν υπάρχει καμία άλλη μέθοδος που να μπορεί να επιλύσει το αντίστοιχο πρόβλημα ελέγχου μοντέλου με στοιχειώδες φράγμα.

**Πόρισμα 2.3.3.** Η συνολική πολυπλοκότητα του προβλήματος ελέγχου μοντέλου για μία πρόταση  $\phi$  σε ένα χρωματισμένο δέντρο  $\mathcal{T}$  είναι

$$O(|T| \cdot f(\phi))$$

όπου η  $f$  είναι μία μη-στοιχειώδης συνάρτηση του μεγέθους της  $\phi$  που αντιστοιχεί στο μέγεθος του  $\text{Assign}_\phi$ .

**Η σχέση του αλγορίθμου του Yannakakis με τον FFG αλγόριθμο.** Η αναγωγή του προβλήματος αποτίμησης της MSO σε ένα πρόβλημα αποτίμησης ενός ακυκλικού συζευκτικού ερωτήματος, και η επίλυση του τελευταίου με τον αλγόριθμο του Yannakakis (βλέπε προηγούμενες ενότητες), μας προσέφεραν βαθιά διαίσθηση ως προς τη φύση της σχέσης που συνδέει τον FFG αλγόριθμο και τον αλγόριθμο του Yannakakis: *ο πυρήνας και των δυο αλγορίθμων είναι η απαλοιφή των άχρηστων τμημάτων της εισόδου, που οδηγεί στη συνέχεια σε έναν αποδοτικό υπολογισμό της εξόδου* (βλέπε την απόδειξη του Πορίσματος 2.3.2). Η ύπαρξη σύνδεσης μεταξύ αυτών των δύο αλγορίθμων, επισημάνθηκε αρχικά στο [FFG02]. Μέσω της αναγωγής μας, οδηγούμαστε στο παρακάτω θεώρημα —άμεση συνέπεια του Πορίσματος 2.3.2— που δηλώνει την ακριβή φύση αυτής της σύνδεσης. Πιο συγκεκριμένα, αποδεικνύουμε ότι ο FFG αλγόριθμος αποτελεί ένα συγκεκριμένο «στιγμιότυπο» του αλγορίθμου του Yannakakis, με την έννοια ότι η εφαρμογή του τελευταίου σε ένα συγκεκριμένο τύπο από στιγμιότυπα προβλημάτων μας δίνει τον πρώτο. Τα στιγμιότυπα προβλημάτων είναι εκείνα που προσομοιώνουν αυτόματα και δέντρα.

**Θεώρημα 2.3.2.** Για κάθε αυτόματο  $A_\phi$  και χρωματισμένο δέντρο  $\mathcal{T} = (t, c)$  υπάρχουν ένα ακυκλικό συζευκτικό ερώτημα  $Q_t$  και μία βάση δεδομένων  $g_{\phi, \mathcal{T}}$  τέτοια ώστε ο υπολογισμός του  $Q_t(g_{\phi, \mathcal{T}})$  με τον αλγόριθμο του Yannakakis να παράγει τα βήματα του FFG αλγορίθμου με είσοδο τα  $A_\phi$  και  $\mathcal{T}$ .

**Παράδειγμα 2.3.3.** Ας θεωρήσουμε τον ατομικό MSO τύπο  $\phi(x, Y) \equiv \text{In}(x, Y)$  με αυτόματο

$$A_\phi = (\Gamma \times \{0, 1\}^2, Q, \delta_0, \delta, F)$$

που δίνεται στον Πίνακα 2.2 (α'). Και ας θεωρήσουμε ένα χρωματισμένο δέντρο  $\mathcal{T} = (t, c)$  με πέντε κόμβους  $n_1$  (ρίζα),  $n_2$  (αριστερό παιδί της ρίζας),  $n_3$  (δεξί παιδί της ρίζας),  $n_4$  (αριστερό παιδί του  $n_2$ ),  $n_5$  (δεξί παιδί του  $n_2$ ).

Ορίζουμε το ακυκλικό συζευκτικό ερώτημα  $Q_t$  ως εξής:

$$Q_t = \pi_{E_{n_1} E_{n_2} E_{n_4} E_{n_5} E_{n_3}} (\text{Trans}_{n_1} \bowtie \text{Trans}_{n_2} \bowtie \text{Trans}_{n_3} \bowtie \text{Trans}_{n_4} \bowtie \text{Trans}_{n_5})$$

όπου  $\text{sort}(\text{Trans}_{n_1}) = Q_{n_1} Q_{n_2} Q_{n_3} E_{n_1}$ ,  $\text{sort}(\text{Trans}_{n_2}) = Q_{n_2} Q_{n_4} Q_{n_5} E_{n_2}$  και  $\text{sort}(\text{Trans}_{n_i}) = Q_{n_i} E_{n_i}$  για  $i = 3, 4, 5$ .

Από τον Ορισμό 2.3.1, έχουμε ότι η βάση δεδομένων  $g_{\phi, \mathcal{T}} = \{\text{trans}_{n_i} \mid i = 1, 2, 3, 4, 5\}$  είναι:

$$\begin{aligned} \langle q, e_1 e_2 \rangle &\in \text{trans}_{n_i} \text{ ανν } \delta_0((\gamma, (e_1, e_2))) = q \quad i = 3, 4, 5 \\ \langle q, q_1, q_2 e_1 e_2 \rangle &\in \text{trans}_{n_2} \text{ ανν } \delta(q_1, q_2(\gamma, (e_1, e_2))) = q \\ \langle q, q_1, q_2 e_1 e_2 \rangle &\in \text{trans}_{n_2} \text{ ανν } \delta(q_1, q_2(\gamma, (e_1, e_2))) = q \ \& \ q \in F \end{aligned}$$

όπου θεωρούμε ότι το χρώμα κάθε κόμβου είναι  $\gamma$ .

Οι δυο αλγορίθμων πραγματοποιούν τρεις διασχίσεις στο δέντρο εισόδου ως εξής:

- Βήμα I. Ο αλγόριθμος του Yannakakis υπολογίζει στο πρώτο βήμα του τις σχέσεις

$$\begin{aligned} \text{p-trans}_{n_i} &= \text{trans}_{n_i} = \{\langle q_0, 00 \rangle, \langle q_0, 01 \rangle, \langle q_f, 10 \rangle, \langle q_a, 11 \rangle\} \quad i = 3, 4, 5 \\ \text{p-trans}_{n_2} &= (\text{trans}_{n_2} \bowtie \pi_{Q_{n_4}} \text{p-trans}_{n_4}) \bowtie \pi_{Q_{n_5}} \text{p-trans}_{n_5} = \text{trans}_{n_2} \\ \text{p-trans}_{n_1} &= (\text{trans}_{n_1} \bowtie \pi_{Q_{n_2}} \text{p-trans}_{n_2}) \bowtie \pi_{Q_{n_3}} \text{p-trans}_{n_3} = \text{trans}_{n_1} \end{aligned}$$

που αποτελούνται από τις πιθανές μεταβάσεις σε κάθε κόμβο  $n \neq n_1$ . Κατά συνέπεια, υπολογίζει και τις πιθανές καταστάσεις σε κάθε τέτοιο κόμβο, τις οποίες υπολογίζει το πρώτο βήμα του αλγορίθμου FFG –δίνεται αναλυτικά στο Παράρτημα Β’– με τον παρακάτω τρόπο

$$\begin{aligned} \text{pot-states}(n_i) &= \pi_{Q_{n_i}} \text{p-trans}_{n_i} = Q = \{q_0, q_a, q_f\} \quad i = 3, 4, 5 \\ \text{pot-states}(n_2) &= \pi_{Q_{n_2}} (\text{trans}_{n_2} \bowtie \text{pot-states}(n_4) \bowtie \text{pot-states}(n_5)) \\ &= \pi_{Q_{n_2}} \text{p-trans}_{n_2}. \end{aligned}$$

- Βήμα II. Ο αλγόριθμος του Yannakakis υπολογίζει στο δεύτερο βήμα του τις σχέσεις  $\text{s-trans}_{n_i}$  ως εξής

$$\begin{aligned} \text{s-trans}_{n_1} &= \text{p-trans}_{n_1} = I \\ \text{s-trans}_{n_2} &= \text{p-trans}_{n_2} \bowtie \pi_{Q_{n_2}} \text{s-trans}_{n_1} = I \cup I' \\ \text{s-trans}_{n_3} &= \text{p-trans}_{n_3} \bowtie \pi_{Q_{n_3}} \text{s-trans}_{n_1} = L \\ \text{s-trans}_{n_i} &= \text{p-trans}_{n_i} \bowtie \pi_{Q_{n_i}} \text{s-trans}_{n_2} = L \quad i = 4, 5 \end{aligned}$$

όπου  $I = \{\langle q_a, q_0, q_0, 11 \rangle, \langle q_a, q_a, q_0, 00 \rangle, \langle q_a, q_a, q_0, 01 \rangle, \langle q_a, q_0, q_a, 00 \rangle, \langle q_a, q_0, q_a, 01 \rangle\}$  και  $I' = \{\langle q_0, q_0, q_0, 00 \rangle, \langle q_0, q_0, q_0, 01 \rangle\}$ ,  $L = \{\langle q_0, 00 \rangle, \langle q_0, 01 \rangle, \langle q_a, 11 \rangle\}$ . Οι σχέσεις  $\text{s-trans}_{n_i}$  έχουν ως στοιχεία τις επιτυχημένες μεταβάσεις σε κάθε κόμβο. Κατά συνέπεια, το βήμα αυτό υπολογίζει και τις επιτυχημένες καταστάσεις σε κάθε κόμβο που υπολογίζει το δεύτερο βήμα του αλγορίθμου FFG με ανάλογο τρόπο ως εξής

$$\begin{aligned} \text{succ-states}(n_1) &= \pi_{Q_{n_1}} \text{s-trans}_{n_1} = \{q_a\} \\ \text{succ-states}(n_i) &= \pi_{Q_{n_i}} (\text{p-trans}_{n_i} \bowtie \pi_{Q_{n_i}} \text{s-trans}_{n_1}) \\ &= \pi_{Q_{n_i}} \text{s-trans}_{n_i} = \{q_0, q_a\} \quad i = 2, 3 \\ \text{succ-states}(n_j) &= \pi_{Q_{n_j}} (\text{p-trans}_{n_j} \bowtie \pi_{Q_{n_j}} \text{s-trans}_{n_2}) \\ &= \pi_{Q_{n_j}} \text{s-trans}_{n_j} = \{q_0, q_a\} \quad j = 4, 5 \end{aligned}$$



a-trans <sub>n<sub>1</sub></sub>					sat-assign(n <sub>1</sub> )
E <sub>n<sub>1</sub></sub>	E <sub>n<sub>2</sub></sub>	E <sub>n<sub>4</sub></sub>	E <sub>n<sub>5</sub></sub>	E <sub>n<sub>3</sub></sub>	(x, Y)
11	00	00	00	00	({n <sub>1</sub> }, {n <sub>1</sub> })
11	00	00	01	00	({n <sub>1</sub> }, {n <sub>1</sub> , n <sub>5</sub> })
11	00	01	00	00	({n <sub>1</sub> }, {n <sub>1</sub> , n <sub>4</sub> })
11	00	01	01	00	({n <sub>1</sub> }, {n <sub>1</sub> , n <sub>4</sub> , n <sub>5</sub> })
⋮	⋮	⋮	⋮	⋮	⋮

Πίνακας 2.3: Η αντιστοιχία της σχέσης a-trans<sub>n<sub>1</sub></sub> και του συνόλου sat-assign(n<sub>1</sub>).

- Βήμα III. Ο αλγόριθμος του Yannakakis υπολογίζει στο τρίτο βήμα του τις σχέσεις a-trans<sub>n<sub>i</sub></sub> ως εξής

$$\begin{aligned}
 \text{a-trans}_{n_i} &= \text{s-trans}_{n_i} = L \quad i = 3, 4, 5 \\
 \text{a-trans}_{n_2} &= \pi_{Q_{n_2} E_{n_2} E_{n_4} E_{n_5}} ((\text{s-trans}_{n_2} \bowtie \text{a-trans}_{n_4}) \bowtie \text{a-trans}_{n_5}) \\
 \text{a-trans}_{n_1} &= \pi_{E_{n_1} E_{n_2} E_{n_4} E_{n_5} E_{n_3}} ((\text{s-trans}_{n_1} \bowtie \text{a-trans}_{n_2}) \bowtie \text{a-trans}_{n_3}).
 \end{aligned}$$

Η σχέση a-trans<sub>n<sub>1</sub></sub> αποτελείται από τις κωδικοποιημένες αναθέσεις που ικανοποιούν τον ln(x, Y) στο δεδομένο δέντρο  $\mathcal{T}$  και οι υπόλοιπες σχέσεις a-trans<sub>n<sub>i</sub></sub> αποτελούνται από τους αντίστοιχους περιορισμούς αυτών. Ανάλογα, το Βήμα III του FFG αλγορίθμου υπολογίζει τα σύνολα αναθέσεων sat-assign(n<sub>i</sub>), που κωδικοποιούν οι σχέσεις που υπολογίζονται παραπάνω. Πιο συγκεκριμένα

$$\begin{aligned}
 \text{sat-assign}(n_i) &= \text{dec}_2(\pi_{E_{n_i}} \text{a-trans}_{n_i}) \quad i = 3, 4, 5 \\
 \text{sat-assign}(n_2) &= \text{dec}_2(\pi_{E_{n_2} E_{n_4} E_{n_5}} \text{a-trans}_{n_2}) \\
 \text{sat-assign}(n_1) &= \text{dec}_2(\text{a-trans}_{n_1}) \\
 &= \phi(\mathcal{T})
 \end{aligned}$$

όπου θεωρούμε ότι  $\text{dec}_2(S) = \{\text{dec}_2(s) \mid s \in S\}$  — η  $\text{dec}_k$  είναι η συνάρτηση αποκωδικοποίησης που ορίστηκε στον Ορισμό 2.2.1— και ότι  $\text{sat-assign}(n_i) = \{\text{sat-assign}(n_i, q) \mid q \in \text{succ-states}(n_i)\}$ .

Η αντιστοιχία της σχέσης a-trans<sub>n<sub>2</sub></sub> με το σύνολο sat-assign(n<sub>2</sub>) δίνεται στον Πίνακα 2.4, ενώ στον Πίνακα 2.3 παρουσιάζεται μέρος της αντιστοιχίας της σχέσης a-trans<sub>n<sub>1</sub></sub> και του συνόλου sat-assign(n<sub>1</sub>).

Όπως αναφέραμε παραπάνω, το Θεώρημα 2.3.2 είναι ένα δευτερεύον αποτέλεσμα της αναγωγής του προβλήματος αποτίμησης της MSO, σε ένα πρόβλημα αποτίμησης ενός ακυκλικού συζευκτικού ερωτήματος. Μία αντίστροφη αναγωγή προτείνεται στο [FFG02] προκειμένου να μεταφερθεί η διαίσθηση για τη σύνδεση αυτών των δύο αλγορίθμων: δεδομένου ενός ακυκλικού

a-trans <sub>n<sub>2</sub></sub>				sat-assign(n <sub>2</sub> , q <sub>0</sub> )
Q <sub>n<sub>2</sub></sub>	E <sub>n<sub>2</sub></sub>	E <sub>n<sub>4</sub></sub>	E <sub>n<sub>5</sub></sub>	(x, Y)
q <sub>0</sub>	00	00	00	(∅, ∅)
q <sub>0</sub>	00	00	01	(∅, {n <sub>5</sub> })
q <sub>0</sub>	00	01	00	(∅, {n <sub>4</sub> })
q <sub>0</sub>	00	01	01	(∅, {n <sub>4</sub> , n <sub>5</sub> })
q <sub>0</sub>	01	00	00	(∅, {n <sub>2</sub> })
q <sub>0</sub>	01	00	01	(∅, {n <sub>2</sub> , n <sub>5</sub> })
q <sub>0</sub>	01	01	00	(∅, {n <sub>2</sub> , n <sub>4</sub> })
q <sub>0</sub>	01	01	01	(∅, {n <sub>2</sub> , n <sub>4</sub> , n <sub>5</sub> })
a-trans <sub>n<sub>2</sub></sub>				sat-assign(n <sub>2</sub> , q <sub>a</sub> )
Q <sub>n<sub>2</sub></sub>	E <sub>n<sub>2</sub></sub>	E <sub>n<sub>4</sub></sub>	E <sub>n<sub>5</sub></sub>	(x, Y)
q <sub>a</sub>	11	00	00	({n <sub>2</sub> }, {n <sub>2</sub> })
q <sub>a</sub>	11	00	01	({n <sub>2</sub> }, {n <sub>2</sub> , n <sub>5</sub> })
q <sub>a</sub>	11	01	00	({n <sub>2</sub> }, {n <sub>2</sub> , n <sub>4</sub> })
q <sub>a</sub>	11	01	01	({n <sub>2</sub> }, {n <sub>2</sub> , n <sub>4</sub> , n <sub>5</sub> })
q <sub>a</sub>	00	00	11	({n <sub>5</sub> }, {n <sub>5</sub> })
q <sub>a</sub>	00	01	11	({n <sub>5</sub> }, {n <sub>4</sub> , n <sub>5</sub> })
q <sub>a</sub>	01	00	11	({n <sub>5</sub> }, {n <sub>2</sub> , n <sub>5</sub> })
q <sub>a</sub>	01	01	11	({n <sub>5</sub> }, {n <sub>2</sub> , n <sub>4</sub> , n <sub>5</sub> })
q <sub>a</sub>	00	11	00	({n <sub>4</sub> }, {n <sub>4</sub> })
q <sub>a</sub>	00	11	01	({n <sub>4</sub> }, {n <sub>4</sub> , n <sub>5</sub> })
q <sub>a</sub>	01	11	00	({n <sub>4</sub> }, {n <sub>2</sub> , n <sub>4</sub> })
q <sub>a</sub>	01	11	01	({n <sub>4</sub> }, {n <sub>2</sub> , n <sub>4</sub> , n <sub>5</sub> })

Πίνακας 2.4: Η σχέση a-trans<sub>n<sub>2</sub></sub> και το αντίστοιχο σύνολο sat-assign(n<sub>2</sub>).

συζευκτικού ερωτήματος  $Q$  με είσοδο τη βάση δεδομένων  $d = \{r_1, \dots, r_n\}$ , ορίζεται ένα αυτόματο  $A_d$  με καταστάσεις τις πλειάδες των σχέσεων της  $d$  και μεταβάσεις που προσομοιώνουν συνενώσεις των πλειάδων των σχέσεων  $r_i$ , έτσι ώστε κάθε εκτέλεση του  $A_d$  στο συντακτικό δέντρο  $\mathcal{T}_Q$  του  $Q$  να αντιστοιχεί σε μία πλειάδα της  $r_1 \bowtie \dots \bowtie r_n$ .

Θυμίζουμε ότι το τρίτο βήμα του FFG αλγορίθμου, διασχίζοντας το δέντρο με κατεύθυνση από τα φύλλα προς τη ρίζα, υπολογίζει διανύσματα πάνω σε υποσύνολα κόμβων του δεδομένου δέντρου, αποκωδικοποιώντας κατάλληλα το επιπρόσθετο μέρος των χρωμάτων σε κάθε κόμβο και συνθέτοντάς τα με τα ήδη υπολογισμένα διανύσματα υποσυνόλων. Συνεπώς είναι προφανές ότι ο FFG αλγόριθμος δεν μπορεί να χρησιμοποιηθεί με είσοδο το δέντρο  $\mathcal{T}_Q$  για τον υπολογισμό του  $Q(d)$ , καθώς το τελευταίο αποτελείται από διανύσματα πάνω στο πεδίο ορισμού  $D$  της  $d$ , και το  $D$  με το σύνολο κόμβων του  $\mathcal{T}_Q$  είναι ξένα σύνολα. Παρόλο που το τρίτο βήμα του FFG αλγορίθμου δεν μπορεί να εφαρμοστεί με είσοδο το δέντρο  $\mathcal{T}_Q$

και συνεπώς μία πλήρης αντιστοιχία ανάμεσα στους δύο αλγορίθμους δε μπορεί να επιτευχθεί με αυτή την αναγωγή, αποδεικνύεται η εξής αντιστοιχία: τα δύο πρώτα βήματα του FFG αλγορίθμου, με είσοδο τα  $\mathcal{T}_Q$  και  $A_d$ , ουσιαστικά συμπίπτουν με το Βήμα I του αλγορίθμου του Yannakakis, με είσοδο τα  $Q$  και  $d$ . Δηλαδή το [FFG02] αποδεικνύει ότι το πρώτο βήμα του FFG αλγορίθμου επιλύοντας το πρόβλημα απόφασης για τα  $A_d$  και  $\mathcal{T}_Q$ , αντιστοιχεί πλήρως στη φάση (α) του Βήματος I του αλγορίθμου του Yannakakis που επιλύει το πρόβλημα ελέγχου μοντέλου για τα  $Q$  και  $d$ . Θυμίζουμε ότι για το σύνολο  $P_r$  των πιθανών καταστάσεων στη ρίζα  $r$ , που υπολογίζεται κατά το πρώτο βήμα του FFG αλγορίθμου με είσοδο τα  $A_\phi$  και  $\mathcal{T}$ , έχουμε ότι  $P_r \cap F \neq \emptyset$  αν το  $A_\phi$  έχει επιτυχημένες εκτελέσεις στο  $\mathcal{T}$ . Έτσι, χρησιμοποιώντας αυτό το βήμα μπορούμε επίσης να αποφανθούμε για το αν  $\mathcal{T} \models \phi$ , στην περίπτωση που η  $\phi$  είναι πρόταση.

Αξίζει να σημειωθεί ότι ο λόγος που καταφέρνουμε να έχουμε μία πλήρη αντιστοιχία μέσω της αναγωγής μας είναι επειδή στην περίπτωσή μας τα δέντρα που αποτελούν την είσοδο των δυο αλγορίθμων έχουν ακριβώς το ίδιο υποκείμενο δέντρο. Πιο συγκεκριμένα, το δέντρο  $\mathcal{T}$ , που είναι η είσοδος του FFG αλγορίθμου, και το συντακτικό δέντρο  $\mathcal{P}_t$  του  $\mathcal{G}_t$ , που είναι η είσοδος του αλγορίθμου του Yannakakis, έχουν το ίδιο υποκείμενο δέντρο  $t$ . Έτσι, οι προβολές στο  $Y_{E_n}$  των σχέσεων που υπολογίζει το Βήμα II του αλγορίθμου του Yannakakis, αντιστοιχούν ακριβώς στους περιορισμούς των αναθέσεων που ικανοποιούν τον τύπο στα  $T_n$ .

## 2.4 Η λύση της datalog για το πρόβλημα αποτίμησης της MSO

Ο τετριμμένος τρόπος υπολογισμού των κωδικοποιημένων αναθέσεων που ικανοποιούν έναν MSO τύπο  $\phi(\bar{X})$  σε ένα χρωματισμένο δέντρο  $\mathcal{T}$ , μέσω της επίλυσης του ισοδύναμου προβλήματος αυτομάτων, συνίσταται στα εξής δυο βήματα: αρχικά υπολογίζουμε όλες τις πιθανές εκτελέσεις του  $\text{Assign}_\phi$  στο  $\mathcal{T}$ , και στη συνέχεια επιλέγουμε από αυτές μόνο εκείνες που είναι επιτυχημένες. Αυτή είναι η διαδικασία που υλοποιεί το datalog πρόγραμμα  $\Pi_{\phi,m}$  της Πρότασης 2.4.1 που δίνεται στην Ενότητα 2.4.2, το οποίο τελικά υπολογίζει τα διανύσματα αναθέσεων των επιτυχημένων εκτελέσεων, δηλαδή επιλύει την κωδικοποιημένη εκδοχή του προβλήματος αποτίμησης της MSO.

Στην Ενότητα 2.4.3, δίνουμε δύο βελτιστοποιημένα datalog προγράμματα για το ίδιο πρόβλημα: το δεύτερο πρόγραμμα που δίνεται στην Πρόταση 2.4.3, προκύπτει με περαιτέρω βελτιστοποίηση του πρώτου προγράμματος που δίνεται στην Πρόταση 2.4.2, και αποτελεί μία βέλτιστη datalog λύση για την κωδικοποιημένη εκδοχή του προβλήματος αποτίμησης της MSO. Τα προγράμματά μας υπολογίζουν τα διανύσματα αναθέσεων των εκτελέσεων κατασκευάζοντας εκτελέσεις από υποεκτελέσεις με ενεργοποιήσεις κανόνων που ακολουθούν τη δομή του υποκείμενου δέντρου του δεδομένου δέντρου  $\mathcal{T}$ .

Στην Ενότητα 2.4.4, δίνουμε δύο βελτιστοποιημένα datalog προγράμματα που επιλύουν το πρόβλημα αποτίμησης της MSO: το πρώτο πρόγραμμα είναι μοναδιαίο, ενώ το δεύτερο έχει βαθμό που εξαρτάται από τον αριθμό των ελεύθερων μεταβλητών του δεδομένου MSO τύπου

και αποδεικνύει ότι τα MSO-ορίσιμα ερωτήματα είναι ορίσιμα στη datalog. Στην Ενότητα 2.4.5, δίνουμε λύσεις για ορισμένες σημαντικές ειδικές περιπτώσεις, ανάμεσα στις οποίες είναι το πρόβλημα ελέγχου μοντέλου και η αποτίμηση μονομελών ερωτημάτων της MSO.

Όλα τα προαναφερθέντα προγράμματα ορίζονται ως προς το αυτόματο  $\text{Assign}_\phi$ . Όπως θα δείξουμε στην Ενότητα 2.4.6, μπορούμε να ορίσουμε ισοδύναμα προγράμματα χρησιμοποιώντας απλά τον τύπο  $\phi$ .

### 2.4.1 Προκατακτικά της datalog

Στο περιβάλλον της datalog, μία *βάση δεδομένων* με πεδίο ορισμού ή σύμπαν  $D$  είναι μία πεπερασμένη σχεσιακή δομή

$$\mathcal{D} = (D, \{r_1, \dots, r_n\})$$

με υπογραφή  $\{R_1, \dots, R_n\}$ , δηλαδή το  $D$  είναι ένα πεπερασμένο σύνολο, και κάθε  $R_i$  είναι ένα σύμβολο κατηγορήματος βαθμού  $a_i$ , που αντιστοιχεί στο όνομα της  $r_i$  πάνω στο  $D$ . Η  $D$  θα αναπαριστάται ισοδύναμα ως ένα σύνολο από  $R_i$ -γεγονότα (facts), δηλαδή εκφράσεις της μορφής  $R_i(c_1, \dots, c_n)$ ,  $c_j \in D$ ,  $i = 1, \dots, n$ :

$$\mathcal{D} = \bigcup_i \{R_i(\bar{c}) \mid \bar{c} \in r_i \subseteq D^{a_i}\}.$$

Ένα *datalog πρόγραμμα* είναι μία συλλογή από *κανόνες* της μορφής

$$A_0 \leftarrow A_1, \dots, A_k,$$

όπου το  $A_0$  είναι η *κεφαλή* (head) και τα  $A_1, \dots, A_k$  αποτελούν το *σώμα* (body) του κανόνα. Κάθε  $A_i$  είναι ένα *άτομο* (atom) δηλαδή μία έκφραση της μορφής  $R(x_1, \dots, x_m)$  όπου κάθε  $x_i$  είναι είτε μία *μεταβλητή* ή μία *σταθερά*, και το  $R$  είναι ένα σύμβολο κατηγορήματος (ή κατηγορήμα για συντομία). Κάθε μεταβλητή που εμφανίζεται στην κεφαλή του κανόνα πρέπει να έχει τουλάχιστον μία εμφάνιση στο σώμα του. Τα κατηγορήματα που εμφανίζονται στις κεφαλές των κανόνων ονομάζονται *προθεσιακά κατηγορήματα* ή *IDB κατηγορήματα* (intensional database predicates) ενώ τα κατηγορήματα που δεν εμφανίζονται σε κεφαλές κανόνων ονομάζονται *εκτασιακά κατηγορήματα* ή *EDB κατηγορήματα* (extensional database predicates). Το σύνολο  $\text{IDB}(\Pi)$  (αντίστοιχα  $\text{EDB}(\Pi)$ ) των IDB (αντίστοιχα EDB) κατηγορημάτων του προγράμματος  $\Pi$  αποτελούν το *προθεσιακό* (αντίστοιχα *εκτασιακό*) *σχήμα* του  $\Pi$ . Το σύνολο των σταθερών που εμφανίζονται στο  $\Pi$  συμβολίζεται με  $\text{Const}(\Pi)$ . Μια βάση δεδομένων αποτελεί *είσοδο για το πρόγραμμα*  $\Pi$  όταν η υπογραφή της ταυτίζεται με το  $\text{EDB}(\Pi)$ .

Δεδομένου ενός σύμπαντος  $D$  και ενός πεπερασμένου συνόλου  $V$  με μεταβλητές, ονομάζουμε *D-αποτίμηση* (valuation)  $v$  πάνω στο  $V$  μία ολική απεικόνιση από το  $V$  στο  $D$ . Ονομάζουμε *D-στιγμιότυποποίηση* ή απλά *στιγμιότυποποίηση* (instantiation) ενός κανόνα  $r$  το αποτέλεσμα της εφαρμογής πάνω στον  $r$  μιας  $D$ -αποτίμησης  $v$  πάνω στο σύνολο των μεταβλητών που εμφανίζονται στον  $r$ , δηλαδή το αποτέλεσμα της αντικατάστασης κάθε μεταβλητής  $x$  στον  $r$  με την τιμή  $v(x)$ . Σε ότι ακολουθεί, το  $\Pi$  θα συμβολίζει ένα πρόγραμμα, η  $\mathcal{D}$  θα συμβολίζει

μία βάση δεδομένων με σύμπαν το  $D$  που αποτελεί είσοδο για το  $\Pi$  και θα αναφερόμαστε σε αυτά ως το ζευγάρι  $(\Pi, \mathcal{D})$ . Λέμε ότι το γεγονός  $P(\bar{a})$ ,  $\bar{a} = (a_1, \dots, a_m)$ ,  $a_i \in D \cup \text{Const}(\Pi)$ , συμπεραίνεται (is derivable) από το  $(\Pi, \mathcal{D})$  ή υπολογίζεται από το  $\Pi$  με είσοδο την  $\mathcal{D}$ , και θα γράφουμε

$$(\Pi, \mathcal{D}) \models P(\bar{a})$$

αν είτε  $P(\bar{a}) \in \mathcal{D}$  ή υπάρχει μία  $D$ -στιγμιότυποποίηση  $r'$  ενός κανόνα  $r$  του  $\Pi$  τέτοια ώστε το  $P(\bar{a})$  να είναι η κεφαλή του  $r'$  και κάθε γεγονός στο σώμα του  $r'$  να συμπεραίνεται από το  $(\Pi, \mathcal{D})$ .

Δεδομένου ενός ζευγαριού  $(\Pi, \mathcal{D})$ , ορίζουμε τη σημασιολογία του  $\Pi$  πάνω στην  $\mathcal{D}$ , ως το σύνολο  $\Pi(\mathcal{D})$  των γεγονότων που συμπεραίνονται από το  $(\Pi, \mathcal{D})$ . Το  $\Pi(\mathcal{D})$  είναι πεπερασμένο και αποτελεί μία βάση δεδομένων με σύμπαν το  $D \cup \text{Const}(\Pi)$  και υπογραφή  $\text{EDB}(\Pi) \cup \text{IDB}(\Pi)$ , έτσι ώστε όταν αυτή, η  $\Pi(\mathcal{D})$ , περιορίζεται στην υπογραφή  $\text{EDB}(\Pi)$  να ταυτίζεται με την  $\mathcal{D}$ . Σύμφωνα με τη μοντελοθεωρητική σημασιολογία των προγραμμάτων, οι κανόνες του  $\Pi$  ερμηνεύονται ως λογικές προτάσεις της μορφής  $\forall \bar{x}\phi$  και έτσι το  $\Pi(\mathcal{D})$  είναι ένα μοντέλο του  $\Pi$  που περιέχει το  $\mathcal{D}$ . Το  $P(\bar{a})$  αληθεύει στο μοντέλο  $\Pi(\mathcal{D})$  αν το  $P(\bar{a})$  συμπεραίνεται από το  $(\Pi, \mathcal{D})$ , δηλαδή  $\Pi(\mathcal{D}) \models P(\bar{a})$  αν  $(\Pi, \mathcal{D}) \models P(\bar{a})$ . Έτσι, κάθε πρόγραμμα  $\Pi$  ορίζει μία απεικόνιση από το σύνολο των βάσεων δεδομένων που έχουν υπογραφή  $\text{EDB}(\Pi)$  στο σύνολο των βάσεων δεδομένων με υπογραφή  $\text{EDB}(\Pi) \cup \text{IDB}(\Pi)$  τέτοια ώστε η εικόνα της  $\mathcal{D}$  να είναι η  $\Pi(\mathcal{D})$ .

Ένα datalog ερώτημα  $Q$  είναι ένα ζευγάρι

$$(\Pi, Q)$$

που αποτελείται από ένα datalog πρόγραμμα  $\Pi$  μαζί με ένα IDB κατηγορημα  $Q \in \text{IDB}(\Pi)$  που ονομάζεται στόχος (goal). Το  $Q$  ορίζει μία απεικόνιση από τις βάσεις δεδομένων που αποτελούν είσοδο για το  $\Pi$  στις σχέσεις με όνομα το κατηγορημα  $Q$ , τέτοια ώστε για κάθε βάση δεδομένων  $\mathcal{D}$  η τιμή  $Q(\mathcal{D})$  να είναι το σύνολο των  $Q$ -γεγονότων που συμπεραίνονται από το  $(\Pi, \mathcal{D})$ . Είναι προφανές ότι  $Q(\mathcal{D}) \subseteq \Pi(\mathcal{D})$ . Ισοδύναμα θα θεωρούμε –ειδικά στις διατυπώσεις των προτάσεών μας– ότι

$$Q(\mathcal{D}) = \{\bar{a} \in D \cup \text{Const}(\Pi) \mid (\Pi, \mathcal{D}) \models Q(\bar{a})\}$$

και θα λέμε ότι το  $Q$  με είσοδο την  $\mathcal{D}$  υπολογίζει την  $Q(\mathcal{D})$ . Δύο datalog ερωτήματα  $(\Pi, Q)$  και  $(\Pi', Q)$  είναι ισοδύναμα αν ορίζουν την ίδια απεικόνιση. Ο βαθμός (arity) του προγράμματος  $\Pi$  είναι ο μέγιστος βαθμός ανάμεσα στα IDB κατηγορήματα του  $\Pi$ . Ο βαθμός του ερωτήματος  $(\Pi, Q)$  είναι ο βαθμός του  $\Pi$ . Ένα datalog πρόγραμμα, και αντίστοιχα ένα ερώτημα, είναι μοναδιαίο (monadic) αν ο βαθμός του είναι 1.

Ο παραπάνω αναδρομικός ορισμός των γεγονότων που συμπεραίνονται από το  $(\Pi, \mathcal{D})$  αποδίδει τον ακόλουθο αλγόριθμο, γνωστό ως *bottom-up αποτίμηση*, για τον υπολογισμό του  $\Pi(\mathcal{D})$  σε έναν πεπερασμένο αριθμό βημάτων: ξεκινάμε θέτοντας  $\mathcal{D}^0 = \mathcal{D}$ . Στη συνέχεια, σε κάθε βήμα  $i+1 > 0$ , παράγουμε μία νέα βάση δεδομένων  $\mathcal{D}^{i+1}$ , εμπλουτίζοντας την τρέχουσα βάση δεδομένων  $\mathcal{D}^i$  με τις κεφαλές εκείνων των στιγμιότυποποιήσεων κανόνων του  $\Pi$  που

έχουν στο σώμα τους μόνο γεγονότα που περιέχονται στην  $\mathcal{D}^i$ . Μετά από  $k$  βήματα, για κάποιο  $k$ , θα έχουμε τελικά ότι

$$\mathcal{D}^k = \mathcal{D}^{k+1}$$

και

$$\Pi(\mathcal{D}) = \mathcal{D}^k.$$

Προφανώς  $k \leq |D|^{\text{ar}(\Pi)}$  όπου  $\text{ar}(\Pi)$  είναι ο βαθμός του  $\Pi$ . Ονομάζουμε *ενεργές* (active) στιγμιοτυποποιήσεις του  $(\Pi, \mathcal{D})$  τις στιγμιοτυποποιήσεις του  $\Pi$  που συμμετέχουν στον υπολογισμό του  $\Pi(\mathcal{D})$  κατά την bottom-up αποτίμηση. Μετράμε την *πολυπλοκότητα αποτίμησης του προγράμματος  $\Pi$*  με είσοδο την  $\mathcal{D}$  με τον αριθμό των ενεργών στιγμιοτυποποιήσεων του  $(\Pi, \mathcal{D})$ .

Θεωρούμε ένα ερώτημα  $\mathcal{Q} = (\Pi, \mathcal{Q})$  τέτοιο ώστε για κάθε βάση δεδομένων  $\mathcal{D}$  να έχουμε ότι

$$\mathcal{Q}(\mathcal{D}) \in \mathcal{D}^k \setminus \mathcal{D}^{k-1}$$

όπου το  $\mathcal{D}^k = \Pi(\mathcal{D})$  ορίζεται όπως παραπάνω. Για αυτού του είδους τα ερωτήματα –τα οποία αντιστοιχούν στα ερωτήματα που ορίζονται στην παρούσα διατριβή– υπολογίζουμε το  $\mathcal{Q}(\mathcal{D})$  μέσω της bottom-up αποτίμησης. Έτσι, μετράμε την *πολυπλοκότητα υπολογισμού του  $\mathcal{Q}(\mathcal{D})$*  από τον αριθμό των ενεργών στιγμιοτυποποιήσεων του  $(\Pi, \mathcal{D})$ .

Οι ενεργές στιγμιοτυποποιήσεις του  $(\Pi, \mathcal{D})$  διαμερίζονται σε δυο σύνολα: i) στο σύνολο που αποτελείται από τις  *$\mathcal{Q}$ -συναφείς στιγμιοτυποποιήσεις* ως προς τη  $\mathcal{D}$ , δηλαδή τις στιγμιοτυποποιήσεις που συμμετέχουν στον συμπερασμό των γεγονότων του  $\mathcal{Q}(\mathcal{D})$  και ii) στο σύνολο που αποτελείται από τις *άσκοπες στιγμιοτυποποιήσεις* ως προς τα  $\mathcal{Q}$  και  $\mathcal{D}$ .

Η *βελτιστοποίηση της bottom-up αποτίμησης* συνίσταται στην απαλοιφή όσο το δυνατόν περισσότερων άσκοπων στιγμιοτυποποιήσεων. Αυτό μπορεί να επιτευχθεί μέσω της αναγραφής του αρχικού ερωτήματος  $\mathcal{Q}$  σε ένα ισοδύναμο ερώτημα  $\mathcal{Q}'$  εφαρμόζοντας έναν *αλγόριθμο βελτιστοποίησης που βασίζεται στην αναγραφή* όπως η μέθοδος *φιλτραρίσματος που βασίζεται στην επίλυση* (resolution-based filtering). Όπως έχουμε ήδη αναφέρει, Το «φιλτράρισμα που βασίζεται στην επίλυση» (ή «φιλτράρισμα επίλυσης» για συντομία) είναι η μέθοδος αναγραφής για την datalog η οποία αναφέρεται συχνά στη βιβλιογραφία με το όνομα «magic sets» (βλέπε επίσης το [BMSU86]). Ο αλγόριθμος αυτός εκτελείται σε δυο βήματα:

- Αρχικά ορίζει νέα κατηγορήματα, που ονομάζονται *κατηγορήματα φίλτρου* (filter predicates), χρησιμοποιώντας ένα πρόγραμμα  $F$  που αποτελεί *φίλτρο* για το  $\mathcal{Q} = (\Pi, \mathcal{Q})$ . Ο ορισμός των κατηγορημάτων φίλτρου στη μέθοδο του φιλτραρίσματος που βασίζεται στην επίλυση προσομοιώνει τις top-down μεθόδους αποτίμησης για την datalog. Ο όρος «επίλυση» στο όνομά της μεθόδου, προκύπτει από το γεγονός ότι αυτές οι μέθοδοι αποτίμησης βασίζονται στην αρχή της επίλυσης.
- Στη συνέχεια αναγράφει το αρχικό πρόγραμμα  $\Pi$  σε ένα νέο πρόγραμμα  $\Pi^F$  στο οποίο τα κατηγορήματα φίλτρου εισάγονται στο σώμα των κανόνων του  $\Pi$ . Το πρόγραμμα  $\Pi^F$  φιλτράρει το σύνολο των ενεργών στιγμιοτυποποιήσεων του αρχικού προγράμματος

Π απαλείφοντας στιγμιοτυποποιήσεις οι οποίες δεν είναι  $\mathcal{Q}$ -συναφείς ενώ διατηρεί όλες τις  $\mathcal{Q}$ -συναφείς στιγμιοτυποποιήσεις.

Προφανώς τα  $\mathcal{Q}$  και  $\mathcal{Q}' = (F \cup \Pi^F, \mathcal{Q})$  είναι ισοδύναμα, και λέμε ότι το  $\Pi$  βελτιστοποιείται στο  $F \cup \Pi^F$ .

Ένα φίλτρο  $F$  ονομάζεται βέλτιστο για το  $\mathcal{Q}$  όταν αυτό επιτυγχάνει να απαλείψει όλες τις άσκοπες στιγμιοτυποποιήσεις, δηλαδή όταν για κάθε βάση δεδομένων  $\mathcal{D}$  οι ενεργές στιγμιοτυποποιήσεις του  $(\Pi^F, \mathcal{D} \cup F(\mathcal{D}))$  αντιστοιχούν ακριβώς στις  $\mathcal{Q}$ -συναφείς στιγμιοτυποποιήσεις του  $\Pi$  ως προς την  $\mathcal{D}$ . Όταν το φίλτρο  $F$  είναι βέλτιστο λέμε ότι το  $\mathcal{Q}$  αναγράφεται κατά τον βέλτιστο τρόπο στο  $\mathcal{Q}'$ .

Μία απεικόνιση (αντίστοιχα boolean απεικόνιση)  $f$  πάνω στην κλάση  $\mathcal{C}_{\mathcal{T}}$  των  $\Gamma$ -δέντρων ονομάζεται  $a$ -datalog ορίσιμη αν υπάρχει ένα datalog ερώτημα  $\mathcal{Q}_f$  με βαθμό  $a$  τέτοιο ώστε, για κάθε δέντρο  $\mathcal{T} \in \mathcal{C}_{\mathcal{T}}$ , να ισχύει  $f(\mathcal{T}) = \mathcal{Q}_f(\mathcal{D}_{\mathcal{T}})$  (αντίστοιχα  $f(\mathcal{T}) = 1$  αν  $\mathcal{Q}_f(\mathcal{D}_{\mathcal{T}}) \neq \emptyset$ ), όπου η βάση δεδομένων  $\mathcal{D}_{\mathcal{T}}$  αντιστοιχεί στο δέντρο  $\mathcal{T}$ .

Επιτρέποντας άτομα με άρνηση στο σώμα των κανόνων της datalog, λαμβάνουμε προγράμματα της datalog με άρνηση που ονομάζονται  $\text{datalog}^{\neg}$  προγράμματα (βλέπε τα [AHV95, Bid91] για αναλυτική παρουσίαση). Ένα  $\text{datalog}^{\neg}$  πρόγραμμα  $\Pi$  ονομάζεται διαστρωματωμένο (stratified) αν υπάρχει μία διαμέριση  $\{\Pi_1, \dots, \Pi_n\}$  του  $\Pi$  τέτοια ώστε i) κάθε IDB να ορίζεται από ένα μοναδικό υποπρόγραμμα  $\Pi_i$  και ii) τα  $\Pi_i$ , που ονομάζονται τα στρώματα (strata) του  $\Pi$ , να μπορούν να διαταχθούν με τέτοιο τρόπο ώστε αν το IDB κατηγορημα  $P'$  (αντίστοιχα η άρνηση του  $P'$ ), που ορίζεται στο στρώμα  $\Pi_j$ , χρησιμοποιείται στον ορισμό του IDB κατηγορηματος  $P$ , που ορίζεται στο στρώμα  $\Pi_k$ , τότε να ισχύει οπωσδήποτε  $\Pi_j \leq \Pi_k$  (αντίστοιχα  $\Pi_j < \Pi_k$ ).

Ας υποθέσουμε ότι η  $\Pi_1 < \dots < \Pi_n$  είναι μία διάταξη των στρωμάτων του  $\Pi$ . Η σημασιολογία του διαστρωματωμένου  $\text{datalog}^{\neg}$  προγράμματος  $\Pi$  ορίζεται ως εξής:

$$\Pi(\mathcal{D}) = \mathcal{D} \cup \bigcup_{1 \leq i \leq n} \Pi'_i(\mathcal{D})$$

όπου, για κάθε  $i$ ,  $P(\bar{a}) \in \Pi'_i(\mathcal{D})$  αν υπάρχει μια στιγμιοτυποποίηση  $r'$  ενός κανόνα  $r$  του  $\Pi_i$  τέτοια ώστε το  $P(\bar{a})$  να είναι η κεφαλή του  $r'$  και κάθε γεγονός  $P'(\bar{b})$  (αντίστοιχα γεγονός με άρνηση  $\neg P'(\bar{b})$ ) στο σώμα του  $r'$  να περιέχεται στο  $\mathcal{D} \cup \bigcup_{j < i} \Pi'_j(\mathcal{D})$  (αντίστοιχα να είναι τέτοιο ώστε το  $P'(\bar{b})$  να μην περιέχεται στο  $\mathcal{D} \cup \bigcup_{j < i} \Pi'_j(\mathcal{D})$ ).

## 2.4.2 Εκφράζοντας το πρόβλημα στη datalog μέσω του $\text{Assign}_{\phi}$

Δεδομένου του αυτομάτου

$$\text{Assign}_{\phi} = (\Gamma, \mathcal{Q} \times \{0, 1\}^k, \Delta_0, \Delta, F \times \{0, 1\}^k)$$

ορίζουμε datalog ερωτήματα για τον  $\phi$  χρησιμοποιώντας τις  $\phi$ -μεταβάσεις για να εκφράσουμε τις εκτελέσεις του  $\text{Assign}_{\phi}$  στην datalog. Σε όλα μας τα προγράμματα τα σύνολα  $D'$ ,  $D$ ,  $D'_0$ ,

$D_0$  αντιστοιχούν στα σύνολα του Ορισμού 2.2.3. Ορίζουμε αρχικά τη βάση δεδομένων  $\mathcal{D}_{\mathcal{T}}$  που αποτελεί την είσοδο των προγραμμάτων μας και περιγράφει το δέντρο  $\mathcal{T}$  με τη μορφή γεγονότων. Στη συνέχεια, εκφράζουμε την κωδικοποιημένη εκδοχή του προβλήματος αποτίμησης της MSO ως ένα datalog ερώτημα.

**Ορισμός 2.4.1.** Ας θεωρήσουμε το δέντρο  $\mathcal{T} = (t, c)$ . Η βάση δεδομένων  $\mathcal{D}_{\mathcal{T}}$  ορίζεται ως εξής:

- $\text{leaf}(n) \in \mathcal{D}_{\mathcal{T}}$  αν το  $n$  είναι ένα φύλλο του  $t$
- $\text{root}(n) \in \mathcal{D}_{\mathcal{T}}$  αν το  $n$  είναι η ρίζα του  $t$
- $\text{succ}_0(n, n_1), \text{succ}_1(n, n_2) \in \mathcal{D}_{\mathcal{T}}$  αν  $s_i(n) = n_i$  για  $i = 1, 2$
- $\text{color-}\gamma(n) \in \mathcal{D}_{\mathcal{T}}$  αν  $c(n) = \gamma$ .

Θεωρούμε ότι το  $\text{Pre}_t$  είναι ένα σύνολο από γεγονότα τα οποία ορίζονται ως εξής:

- $\text{pre-}i(n) \in \text{Pre}_t$  αν το  $n$  είναι το  $i$ -οστό στοιχείο ως προς την προδιατεταγμένη διάσχιση του  $t$ .

Ορίζουμε την  $\mathcal{D}_{\mathcal{T}}^{\text{pre}} = \mathcal{D}_{\mathcal{T}} \cup \text{Pre}_t$ .

**Πρόταση 2.4.1.** Ας υποθέσουμε ότι ο  $\phi(X_1, \dots, X_k)$  είναι ένας  $\text{MSO}[\tau_{\Gamma}]$  τύπος και ότι το  $m$  είναι ένας ακέραιος. Υπάρχει ένα datalog ερώτημα  $\mathcal{Q}_{\phi, m} = (\Pi_{\phi, m}, \text{assign})$ , βαθμού  $m + 1$ , το οποίο για κάθε  $\Gamma$ -δέντρο  $\mathcal{T}$ , μεγέθους  $m$ , υπολογίζει, με είσοδο την βάση δεδομένων  $\mathcal{D}_{\mathcal{T}}$ , τις κωδικοποιημένες αναθέσεις που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$ . Δηλαδή

$$\varepsilon_{\bar{B}} \in \mathcal{Q}_{\phi, m}(\mathcal{D}_{\mathcal{T}}) \quad \text{αν} \quad \bar{B} \in \phi(\mathcal{T}).$$

Το  $\mathcal{Q}_{\phi, m}(\mathcal{D}_{\mathcal{T}})$  υπολογίζεται μετά από το πολύ  $m \cdot 2^{m \cdot k} + a$  στιγμιοτυποποιήσεις κανόνων όπου  $a = |\phi(\mathcal{T})|$ .

**Απόδειξη.** Αρχικά ορίζουμε τις οικογένειες κανόνων που ακολουθούν. Σημειώνουμε ότι χρησιμοποιούμε τις συντομεύσεις  $(x_1, \bar{u}) = (x_1, u_1, \dots, u_i)$  και  $(x_2, \bar{v}) = (x_2, v_1, \dots, v_j)$ .

$$\begin{aligned} A. \quad & r_{\gamma q e}^1 : \quad \text{assign-}q^1(x, e) \leftarrow \text{color-}\gamma(x), \text{leaf}(x) \\ & r_{\gamma q q_1 q_2 e}^{i+j+1} : \quad \text{assign-}q^{i+j+1}(x, e, \bar{u}, \bar{v}) \leftarrow \text{color-}\gamma(x), \text{succ}_0(x, x_1), \text{assign-}q_1^i(x_1, \bar{u}), \\ & \quad \quad \quad \text{succ}_1(x, x_2), \text{assign-}q_2^j(x_2, \bar{v}) \end{aligned}$$

όπου  $\gamma q e \in D_0$  και  $\gamma q q_1 q_2 e \in D$ . Θυμίζουμε από τον Ορισμό 2.2.3 ότι τα στοιχεία των  $D_0$  και  $D$  αντιστοιχούν στις μεταβάσεις του  $\text{Assign}_{\phi}$ . Τα  $i, j$  είναι περιττοί αριθμοί τ.ω.  $1 \leq i, j \leq m - 2$  και  $3 \leq i + j + 1 \leq m$ . Κάθε  $e$  αποτελεί σταθερά του προγράμματος και είναι μία λέξη από 0 και 1 μεγέθους  $k$ .

Οι παραπάνω κανόνες προσομοιώνουν τον υπολογισμό των εκτελέσεων του  $\text{Assign}_{\phi}$  σε ένα  $\Gamma$ -δέντρο  $\mathcal{T}$  μεγέθους  $m$ . Πιο συγκεκριμένα, για κάθε γεγονός  $\text{assign-}q^{m_n}(n, \bar{\epsilon})$ ,  $\bar{\epsilon} = (\epsilon_1, \dots, \epsilon_{m_n})$ , που υπολογίζεται από τους παραπάνω κανόνες, υπάρχει μία εκτέλεση  $\zeta; \bar{\epsilon}$  του  $\text{Assign}_{\phi}$  στο



υποδέντρο  $\mathcal{T}_n$  του  $\mathcal{T}$  μεγέθους  $m_n$ , τέτοια ώστε  $\zeta(n) = q$ . Έτσι, κάθε γεγονός  $\text{assign-}q^m(r, \bar{e})$  που υπολογίζεται αντιστοιχεί στο διάνυσμα ανάθεσης  $\bar{e} = (e_1, \dots, e_m)$  μιας εκτέλεσης του  $\text{Assign}_\phi$  που αναθέτει την κατάσταση  $q\bar{e}_1$  στη ρίζα  $r$  του  $\mathcal{T}$ . Θυμίζουμε ότι το διάνυσμα ανάθεσης είναι διατεταγμένο ως προς την προδιατεταγμένη διάσχιση του  $\mathcal{T}$ . Για να επιλέξουμε τις αναθέσεις που ικανοποιούν τον τύπο προσθέτουμε την ακόλουθη οικογένεια κανόνων για κάθε  $q \in F$  (ενώ θεωρούμε ότι  $\bar{u} = (u_1, \dots, u_m)$ )

$$r_q : \quad \text{assign}(\bar{u}) \leftarrow \text{assign-}q^m(x, \bar{u}).$$

Με αυτόν τον τρόπο προκύπτει το ερώτημα  $\mathcal{Q}_{\phi, m} = (\Pi_{\phi, m}, \text{assign})$ , που υπολογίζει τις κωδικοποιημένες αναθέσεις που ικανοποιούν τον τύπο  $\phi$  στο  $\mathcal{T}$ .

*Πολυπλοκότητα.* Από το Λήμμα 2.2.2 έχουμε ότι ο αριθμός των πιθανών εκτελέσεων σε ένα υποδέντρο μεγέθους  $m_n$  είναι  $2^{m_n \cdot k}$ . Έτσι, για κάθε κόμβο  $n$  ενός  $\Gamma$ -δέντρου  $\mathcal{T}$  μεγέθους  $m$ , το πρόγραμμα  $\Pi_{\phi, m}$  παράγει ακριβώς  $2^{m_n \cdot k}$  γεγονότα  $\text{assign-}q^{m_n}(n, \bar{e})$  για κατάλληλα  $q, n, \bar{e}$ . Η παραγωγή αυτή πραγματοποιείται μέσω ισάριθμων στιγμιοτυποποιήσεων κανόνων. Συνεπώς, ο συνολικός αριθμός γεγονότων που παράγονται κατά τον υπολογισμό του ερωτήματος  $(\Pi_{\phi, m}, \text{assign})$  είναι

$$\sum_{n \in T} 2^{m_n \cdot k} + a < m \cdot 2^{m \cdot k} + a$$

όπου το  $a = |\phi(\mathcal{T})|$  είναι ο αριθμός των στιγμιοτυποποιήσεων κανόνων της μορφής  $r_q$ .  $\square$

**Παράδειγμα 2.4.1.** Ορίζουμε το ερώτημα  $\mathcal{Q}_{\phi, 3} = (\Pi_{\phi, 3}, \text{assign})$  της παραπάνω πρότασης για τον ατομικό τύπο  $\phi(X, Y) \equiv E(X, Y)$  και ένα  $\Gamma$ -δέντρο με τρεις κόμβους. Το πρόγραμμα  $\Pi_{\phi, 3}$  αποτελείται από τους παρακάτω κανόνες:

$$\begin{aligned} r_{\gamma q_0 00}^1 & : \quad \text{assign-}q_0^1(x, 00) \leftarrow \text{color-}\gamma(x), \text{leaf}(x) \\ r_{\gamma q_0 01}^1 & : \quad \text{assign-}q_0^1(x, 01) \leftarrow \text{color-}\gamma(x), \text{leaf}(x) \\ r_{\gamma q_f 10}^1 & : \quad \text{assign-}q_f^1(x, 10) \leftarrow \text{color-}\gamma(x), \text{leaf}(x) \\ r_{\gamma q_a 11}^1 & : \quad \text{assign-}q_a^1(x, 11) \leftarrow \text{color-}\gamma(x), \text{leaf}(x) \\ r_{\gamma q_0 q_0 q_0 00}^3 & : \quad \text{assign-}q_0^3(x, 00, u, v) \leftarrow \text{color-}\gamma(x), \text{succ}_0(x, x_1), \text{assign-}q_0^1(x_1, u), \\ & \quad \text{succ}_1(x, x_2), \text{assign-}q_0^1(x_2, v) \\ r_{\gamma q_0 q_0 q_0 01}^3 & : \quad \text{assign-}q_0^3(x, 01, u, v) \leftarrow \text{color-}\gamma(x), \text{succ}_0(x, x_1), \text{assign-}q_0^1(x_1, u), \\ & \quad \text{succ}_1(x, x_2), \text{assign-}q_0^1(x_2, v) \\ r_{\gamma q_f q_0 q_0 10}^3 & : \quad \text{assign-}q_0^3(x, 10, u, v) \leftarrow \text{color-}\gamma(x), \text{succ}_0(x, x_1), \text{assign-}q_0^1(x_1, u), \\ & \quad \text{succ}_1(x, x_2), \text{assign-}q_0^1(x_2, v) \\ r_{\gamma q_a q_0 q_0 11}^3 & : \quad \text{assign-}q_0^3(x, 11, u, v) \leftarrow \text{color-}\gamma(x), \text{succ}_0(x, x_1), \text{assign-}q_0^1(x_1, u), \\ & \quad \text{succ}_1(x, x_2), \text{assign-}q_0^1(x_2, v) \\ r_{\gamma q_a q_a q_0 0i}^3 & : \quad \text{assign-}q_a^3(x, 0i, u, v) \leftarrow \text{color-}\gamma(x), \text{succ}_0(x, x_1), \text{assign-}q_a^1(x_1, u), \end{aligned}$$

$$\begin{aligned}
& \text{succ}_1(x, x_2), \text{assign-}q_0^1(x_2, v) \\
& \text{για } i = 0, 1 \\
r_{\gamma q_f q_a q_0 i 0}^3 : & \text{assign-}q_f^3(x, i0, u, v) \leftarrow \text{color-}\gamma(x), \text{succ}_0(x, x_1), \text{assign-}q_a^1(x_1, u), \\
& \text{succ}_1(x, x_2), \text{assign-}q_0^1(x_2, v) \\
& \text{για } i = 0, 1 \\
r_{\gamma q_a q_0 q_a 0 i}^3 : & \text{assign-}q_a^3(x, 0i, u, v) \leftarrow \text{color-}\gamma(x), \text{succ}_0(x, x_1), \text{assign-}q_0^1(x_1, u), \\
& \text{succ}_1(x, x_2), \text{assign-}q_a^1(x_2, v) \\
& \text{για } i = 0, 1 \\
r_{\gamma q_f q_0 q_a i 0}^3 : & \text{assign-}q_f^3(x, i0, u, v) \leftarrow \text{color-}\gamma(x), \text{succ}_0(x, x_1), \text{assign-}q_0^1(x_1, u), \\
& \text{succ}_1(x, x_2), \text{assign-}q_a^1(x_2, v) \\
& \text{για } i = 0, 1 \\
r_{\gamma q_f q_a q_a e}^3 : & \text{assign-}q_f^3(x, e, u, v) \leftarrow \text{color-}\gamma(x), \text{succ}_0(x, x_1), \text{assign-}q_a^1(x_1, u), \\
& \text{succ}_1(x, x_2), \text{assign-}q_a^1(x_2, v) \\
& \text{για } e = 00, 01, 10, 11 \\
r_{\gamma q_f q_f q_e}^3 : & \text{assign-}q_f^3(x, e, u, v) \leftarrow \text{color-}\gamma(x), \text{succ}_0(x, x_1), \text{assign-}q_f^1(x_1, u), \\
& \text{succ}_1(x, x_2), \text{assign-}q^1(x_2, v) \\
& \text{για } q = q_0, q_a, q_f \quad \& \quad e = 00, 01, 10, 11 \\
r_{\gamma q_f q q_e}^3 : & \text{assign-}q_f^3(x, e, u, v) \leftarrow \text{color-}\gamma(x), \text{succ}_0(x, x_1), \text{assign-}q^1(x_1, u), \\
& \text{succ}_1(x, x_2), \text{assign-}q_f^1(x_2, v) \\
& \text{για } q = q_0, q_a \quad \& \quad e = 00, 01, 10, 11 \\
r_{q_a} : & \text{assign}(u_1, u_2, u_3) \leftarrow \text{assign-}q_a^3(x, u_1, u_2, u_3).
\end{aligned}$$

όπου  $\gamma \in \Gamma$ .

Αποδείξαμε λοιπόν σε αυτή την ενότητα ότι η κωδικοποιημένη εκδοχή του προβλήματος αποτίμησης της MSO σε πεπερασμένα δέντρα ανάγεται σε ένα πρόβλημα αποτίμησης ενός datalog ερωτήματος. Πιο συγκεκριμένα, δείξαμε ότι σε κάθε ζευγάρι  $(\phi, \mathcal{T})$ , για  $\mathcal{T}$  μεγέθους  $m$ , μπορούμε να συσχετίσουμε ένα ερώτημα  $\mathcal{Q}_{\phi, m}$  και μία βάση δεδομένων  $\mathcal{D}_{\mathcal{T}}$  τ.ω.

$$\bar{B} \in \phi(\mathcal{T}) \quad \text{ανν} \quad \varepsilon_{\bar{B}} \in \mathcal{Q}_{\phi, m}(\mathcal{D}_{\mathcal{T}})$$

δηλαδή τ.ω.  $\phi_{\text{enc}}(\mathcal{T}) = \mathcal{Q}_{\phi, m}(\mathcal{D}_{\mathcal{T}})$ .

### 2.4.3 Βελτιστοποιημένες εκδοχές του datalog προγράμματος $\Pi_{\phi, m}$

Το datalog πρόγραμμα  $\Pi_{\phi, m}$  της Πρότασης 2.4.1 παράγει άσκοπες στιγμιτυποποιήσεις ως προς το ερώτημα  $\mathcal{Q}_{\phi, m}$ , και μπορεί να βελτιστοποιηθεί με την εισαγωγή κατηγορημάτων

φίλτρου στο σώμα των κανόνων του. Η βελτιστοποίηση μπορεί να πραγματοποιηθεί με δύο διαφορετικούς τρόπους και οδηγεί σε δύο διαφορετικά datalog προγράμματα: i) στο  $\Pi_{\phi,m}^R$  που προκύπτει μέσω της άμεσης εφαρμογής της μεθόδου αναγραφής που βασίζεται στην επίλυση και ii) στο  $\Pi_{\phi,m}^S$  που αποτελεί ένα επιπρόσθετα βελτιστοποιημένο πρόγραμμα. Αποδεικνύουμε ότι το  $\Pi_{\phi,m}$  βελτιστοποιείται στο  $\Pi_{\phi,m}^R$  το οποίο με τη σειρά του βελτιστοποιείται στο  $\Pi_{\phi,m}^S$ . Επίσης αποδεικνύουμε ότι τα αντίστοιχα ερωτήματα  $\mathcal{Q}_{\phi,m}^R = (\Pi_{\phi,m}^R, \text{assign})$  και  $\mathcal{Q}_{\phi,m}^S = (\Pi_{\phi,m}^S, \text{assign})$  είναι ισοδύναμα με το  $\mathcal{Q}_{\phi,m} = (\Pi_{\phi,m}, \text{assign})$ .

Για να καταλάβουμε καλύτερα τις ιδέες των βελτιστοποιήσεων που πραγματοποιούνται, ας δούμε καλύτερα μερικά βασικά στοιχεία για το πρόγραμμα  $\Pi_{\phi,m}$ . Αρχικά ας σημειώσουμε ότι το  $\Pi_{\phi,m}$ , όπως άλλωστε και τα βελτιστοποιημένα προγράμματα  $\Pi_{\phi,m}^R$  και  $\Pi_{\phi,m}^S$ , υπολογίζουν τα διανύσματα αναθέσεων των  $\mathcal{T}_n$ -υποεκτελέσεων συνενώνοντας κατάλληλα τα διανύσματα αναθέσεων των  $\mathcal{T}_{n_1}, \mathcal{T}_{n_2}$ -υποεκτελέσεων. Επιπλέον, το  $\Pi_{\phi,m}$  υπολογίζει έμμεσα και τα διανύσματα καταστάσεων των υποεκτελέσεων τα οποία μπορούν να ανακτηθούν από τις καταστάσεις  $q$  που εμφανίζονται στο όνομα των IDB κατηγορημάτων  $\text{assign-}q^\ell$  που συμμετέχουν στον συμπερασμό των αντίστοιχων διανυσμάτων αναθέσεων. Συνεπώς, για κάθε  $n$ , το σύνολο  $S(n)$  των καταστάσεων  $q$  για τις οποίες τα γεγονότα  $\text{assign-}q^{m_n}(n, \bar{e})$  παράγονται από το  $\Pi_{\phi,m}$ , αντιστοιχεί ακριβώς στο σύνολο των πιθανών καταστάσεων στον  $n$ .

Στα βελτιστοποιημένα προγράμματα, αυτό το σύνολο  $S(n)$  αντιστοιχεί στην τομή του συνόλου των πιθανών καταστάσεων στον  $n$  με το σύνολο των καταστάσεων που υπολογίζονται στον  $n$  μέσω του φίλτρου. Δηλαδή, το  $S(n)$  είναι ένα υποσύνολο του συνόλου των πιθανών καταστάσεων το οποίο περιέχει όλες τις επιτυχημένες καταστάσεις στον  $n$ . Αποδεικνύουμε ότι στο πλήρως βελτιστοποιημένο πρόγραμμα  $\Pi_{\phi,m}^S$ , το σύνολο  $S(n)$  ταυτίζεται με το σύνολο των επιτυχημένων καταστάσεων στον  $n$ .

**Πρόταση 2.4.2.** Ας υποθέσουμε ότι η εφαρμογή της μεθόδου αναγραφής που βασίζεται στην επίλυση για το ερώτημα  $\mathcal{Q}_{\phi,m} = (\Pi_{\phi,m}, \text{assign})$  οδηγεί στο ισοδύναμο ερώτημα

$$\mathcal{Q}_{\phi,m}^R = (\Pi_{\phi,m}^R, \text{assign}).$$

Το φίλτρο  $R$  του  $\Pi_{\phi,m}^R$  παράγει καταστάσεις που αποτελούν υπερσύνολο των επιτυχημένων καταστάσεων και δεν αποτελεί βέλτιστο φίλτρο.

**Απόδειξη.** Με κατάλληλη προσαρμογή της μεθόδου αναγραφής που βασίζεται στην επίλυση του [BMSU86] για το ερώτημα  $(\Pi_{\phi,m}, \text{assign})$ , λαμβάνουμε το ακόλουθο πρόγραμμα  $\Pi_{\phi,m}^R = R \cup A_R$

$$\begin{array}{lll} R. & r_q : & \text{resol-}q(x) \leftarrow \text{root}(x) \\ & r_{\gamma q q_1 q_2}^1 : & \text{resol-}q_1(x_1) \leftarrow \text{resol-}q(x), \text{succ}_0(x, x_1), \text{color-}\gamma(x) \\ & r_{\gamma q q_1 q_2}^2 : & \text{resol-}q_2(y_1) \leftarrow \text{resol-}q(x), \text{succ}_1(x, y_1), \text{color-}\gamma(x) \end{array}$$

όπου  $q \in F$  στον κανόνα  $r_q$ , και  $\gamma q q_1 q_2 \in D'$  στους υπόλοιπους.

$$A_R. \quad f_{\gamma q e}^1 : \quad \text{assign-}q^1(x, e) \leftarrow \text{resol-}q(x), \text{color-}\gamma(x), \text{leaf}(x)$$

$$\begin{aligned}
f_{\gamma q q_1 q_2 e}^{i+j+1} &: \text{assign-}q^{i+j+1}(x, e, \bar{u}, \bar{v}) \leftarrow \text{resol-}q(x), \text{color-}\gamma(x), \text{succ}_0(x, x_1), \\
&\quad \text{assign-}q_1^i(x_1, \bar{u}), \text{succ}_1(x, x_2), \text{assign-}q_2^j(x_2, \bar{v}) \\
a_q &: \text{assign}(\bar{e}) \leftarrow \text{assign-}q^m(x, \bar{e})
\end{aligned}$$

όπου  $\gamma q e \in D_0$ ,  $\gamma q q_1 q_2 e \in D$ ,  $q \in Q$ . Οι  $i, j$  είναι περιττοί αριθμοί τ.ω.  $1 \leq i, j \leq m - 2$  και  $3 \leq i + j + 1 \leq m$ .

Η ορθότητα του μετασχηματισμένου προγράμματος  $\Pi_{\phi, m}^R$  έπεται άμεσα από την ορθότητα της μεθόδου που εφαρμόσαμε και η οποία διατηρεί την ισοδυναμία ως προς το αρχικό ερώτημα. Συνεπώς, το πρόγραμμα  $\Pi_{\phi, m}^R$  υπολογίζει με τη μορφή γεγονότων  $\text{assign}(\bar{e})$  ακριβώς τα διανύσματα αναθέσεων  $\bar{e}$  όλων των επιτυχημένων εκτελέσεων του  $\text{Assign}_\phi$  στο  $\mathcal{T}$ .

Παρόλα αυτά, θα σκιαγραφήσουμε τώρα μία απόδειξη αυτού του αποτελέσματος ορθότητας, προσαρμοσμένη στο πλαίσιο που οριοθετήσαμε παραπάνω, η οποία θα μας δώσει παράλληλα τη διαίσθηση των βασικών σημείων του αλγορίθμου. Αρχικά δίνουμε τους ορισμούς κάποιων εννοιών που θα διευκολύνουν την παρουσίασή μας. Μία κατάσταση  $q$  ονομάζεται  $R$ -φιλτραρισμένη στον  $n$  αν το γεγονός  $\text{resol-}q(n)$  παράγεται από το φίλτρο  $R$ . Αποδεικνύεται εύκολα ότι σε κάθε κόμβο  $n$  i) το σύνολο των  $R$ -φιλτραρισμένων καταστάσεων περιέχει όλες τις επιτυχημένες καταστάσεις—πράγματι η ιδιότητα αυτή ισχύει στη ρίζα και μεταφέρεται σταδιακά σε ολόκληρο το δέντρο καθώς το διασχίζουμε από πάνω προς τα κάτω— και ii) το σύνολο των  $R$ -πιθανών καταστάσεων, δηλαδή των καταστάσεων που είναι ταυτόχρονα  $R$ -φιλτραρισμένες και πιθανές, περιέχει όλες τις επιτυχημένες καταστάσεις. Η τελευταία αυτή πρόταση είναι προφανής, καθώς μία  $R$ -φιλτραρισμένη κατάσταση η οποία δεν είναι πιθανή δε μπορεί να είναι ούτε επιτυχημένη.

Κατά την bottom-up αποτίμηση του  $A_R$ , οι  $R$ -φιλτραρισμένες καταστάσεις φιλτράρουν τον συμπερασμό των πιθανών υποεκτελέσεων και εκτελέσεων έτσι ώστε να υπολογίζονται μόνο τα διανύσματα αναθέσεων των  $R$ -πιθανών υποεκτελέσεων και εκτελέσεων, δηλαδή εκείνων που συνθέτονται από  $R$ -πιθανές καταστάσεις. Πράγματι, αποδεικνύεται εύκολα με επαγωγή ότι το γεγονός  $\text{assign-}q^{|\mathcal{T}_n|}(n, \bar{e})$  υπολογίζεται από το  $A_R$  αν η  $q$  είναι μία  $R$ -πιθανή κατάσταση στον κόμβο  $n$ . Συνεπώς, το σύνολο των  $R$ -πιθανών  $\mathcal{T}_n$ -υποεκτελέσεων περιέχει τους περιορισμούς στο  $\mathcal{T}_n$  όλων των επιτυχημένων εκτελέσεων. Από το Λήμμα 2.2.2, έχουμε ότι το ίδιο ακριβώς ισχύει και για τα αντίστοιχα διανύσματα αναθέσεων. Επειδή όμως στη ρίζα όλες οι  $R$ -φιλτραρισμένες καταστάσεις είναι τελικές, οι  $R$ -πιθανές εκτελέσεις είναι ακριβώς οι επιτυχημένες εκτελέσεις. Και κατά συνέπεια τα διανύσματα αναθέσεων που υπολογίζονται από το  $A_R$  ως γεγονότα του  $\text{assign}$ , αντιστοιχούν ακριβώς στις κωδικοποιημένες αναθέσεις που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$ . Έτσι ολοκληρώνεται η σκιαγράφηση της απόδειξης ορθότητας.

Το γεγονός ότι το  $R$  δεν είναι ένα βέλτιστο φίλτρο οφείλεται στο ότι δεν ισχύει η παρακάτω ιδιότητα: για κάθε ζευγάρι  $q_1, q_2$   $R$ -πιθανών καταστάσεων στα  $n_1, n_2$  αντίστοιχα, υπάρχει μία  $R$ -πιθανή κατάσταση  $q$  στον  $n$  χρώματος  $\gamma$  τέτοια ώστε  $\gamma q q_1 q_2 \in D'$ . Για παράδειγμα, ας υποθέσουμε ότι, για δεδομένες  $R$ -πιθανές καταστάσεις  $q_1, q_2$  στα  $n_1, n_2$ , κάθε  $q$  τ.ω.  $\gamma q q_1 q_2 \in D'$ , ενώ είναι πιθανή κατάσταση στον κόμβο  $n$  χρώματος  $\gamma$  δεν είναι  $R$ -φιλτραρισμένη στον ίδιο κόμβο. Σε αυτή την περίπτωση η  $R$ -φιλτραρισμένη κατάσταση  $q_1$  (αντίστοιχα η  $q_2$ ) στον  $n_1$  (αντίστοιχα στον  $n_2$ ) παράγεται από κανόνες  $m_{\gamma q' q_1 q_2}^1$ ,  $q_2' \neq q_2$  (αντίστοιχα  $m_{\gamma q'' q_1' q_2}^2$ ,

$q'_1 \neq q_1$ ) όπου  $q \neq q' \neq q''$ . Κατά συνέπεια, δεν ισχύει ότι όλα τα γεγονότα  $\text{assign-}q_1^{|\mathcal{T}_{n_1}|}(n_1, \bar{\epsilon}_1)$ ,  $\text{assign-}q_2^{|\mathcal{T}_{n_2}|}(n_2, \bar{\epsilon}_2)$  που παράγονται από το  $A_M$  θα συμμετέχουν αναγκαστικά στο σώμα μιας στιγμιοτυποποίησης ενός κανόνα  $f_{\gamma q q_1 q_2}^{|\mathcal{T}_{n_1}|+|\mathcal{T}_{n_2}|+1}$  του  $A_M$ . Ένα γεγονός  $\text{assign-}q_i^{|\mathcal{T}_{n_i}|}(n_i, \bar{\epsilon}_i)$  είναι «άχρηστο» όταν η κατάσταση  $q_i$ , παρόλο που είναι  $R$ -φιλτραρισμένη και πιθανή στον  $n_i$ , δεν είναι επιτυχημένη στον κόμβο αυτό. Έτσι, κάθε στιγμιοτυποποίηση που συμμετέχει στον συμπερασμό ενός τέτοιου «άχρηστου» γεγονότος είναι άσκοπη.  $\square$

**Παράδειγμα 2.4.2.** Ας θεωρήσουμε το μη βελτιστοποιημένο ερώτημα  $\mathcal{Q}_{\phi,3} = (\Pi_{\phi,3}, \text{assign})$  του Παραδείγματος 2.4.1 που αντιστοιχούσε στον ατομικό τύπο  $\phi(X, Y) \equiv E(X, Y)$  και σε ένα  $\Gamma$ -δέντρο με τρεις κόμβους. Παραθέτουμε στη συνέχεια το φίλτρο  $R$  που προκύπτει μετά την εφαρμογή της μεθόδου αναγραφής που βασίζεται στην επίλυση του [BMSU86] για το ερώτημα  $\mathcal{Q}_{\phi,3}$ :

$$\begin{aligned} \text{resol-}q_a(x) &\leftarrow \text{root}(x) \\ \text{resol-}q_0(y) &\leftarrow \text{resol-}q_0(x), \text{succ}_i(x, y), \text{color-}\gamma(x) \\ \text{resol-}q_0(y) &\leftarrow \text{resol-}q_a(x), \text{succ}_i(x, y), \text{color-}\gamma(x) \\ \text{resol-}q_a(y) &\leftarrow \text{resol-}q_a(x), \text{succ}_i(x, y), \text{color-}\gamma(x) \\ \text{resol-}q_0(y) &\leftarrow \text{resol-}q_f(x), \text{succ}_i(x, y), \text{color-}\gamma(x) \\ \text{resol-}q_a(y) &\leftarrow \text{resol-}q_f(x), \text{succ}_i(x, y), \text{color-}\gamma(x) \\ \text{resol-}q_f(y) &\leftarrow \text{resol-}q_f(x), \text{succ}_i(x, y), \text{color-}\gamma(x) \end{aligned}$$

όπου  $i = 0, 1$  και  $\gamma \in \Gamma$ .

**Πρόταση 2.4.3.** Το ερώτημα  $\mathcal{Q}_{\phi,m} = (\Pi_{\phi,m}, \text{assign})$  μπορεί να αναγραφεί κατά βέλτιστο τρόπο σε ένα ισοδύναμο ερώτημα

$$\mathcal{Q}_{\phi,m}^S = (\Pi_{\phi,m}^S, \text{assign}).$$

Το βέλτιστο φίλτρο  $P \cup S$  του  $\Pi_{\phi,m}^S$  παράγει ακριβώς όλες τις επιτυχημένες καταστάσεις.

Η πολυπλοκότητα αποτίμησης του ερωτήματος  $\mathcal{Q}_{\phi,m}^S$  με είσοδο τη βάση δεδομένων  $\mathcal{D}_{\mathcal{T}}$ , που αντιστοιχεί σε ένα δέντρο  $\mathcal{T}$  μεγέθους  $m$ , είναι το πολύ

$$m \cdot (s^2 \cdot 2^{k+1} + a) + a$$

όπου το  $s$  είναι το μέγεθος του συνόλου  $Q$  των  $\phi$ -καταστάσεων και το  $a = |\phi(\mathcal{T})|$ .

**Απόδειξη.** Αποδεικνύουμε ότι μπορούμε να «ξαναφιλτράρουμε» το φίλτρο  $R$  της Πρότασης 2.4.2 μέσω ενός νέου φίλτρου  $P$  και έτσι να έχουμε τελικά ένα βέλτιστο φίλτρο  $P \cup S$  με το  $S$  να προκύπτει από την ενσωμάτωση του φίλτρου  $P$  στο  $R$ .

$$P. \quad p_{\gamma q} : \quad \text{pot-}q(x) \leftarrow \text{leaf}(x), \text{color-}\gamma(x)$$

$$\begin{aligned}
P_{\gamma q q_1 q_2} &: \text{pot-}q(x) \leftarrow \text{pot-}q_1(x_1), \text{pot-}q_2(x_2), \text{succ}_0(x, x_1), \text{succ}_1(x, x_2), \text{color-}\gamma(x) \\
S. \quad s_q &: \text{suc-}q(x) \leftarrow \text{pot-}q(x), \text{root}(x) \\
s_{\gamma q q_1 q_2}^i &: \text{suc-}q_i(x_i) \leftarrow \text{suc-}q(x), \text{pot-}q_1(x_1), \text{pot-}q_2(x_2), \text{succ}_0(x, x_1), \text{succ}_1(x, x_2), \text{color-}\gamma(x)
\end{aligned}$$

όπου  $i = 1, 2$ , το  $q \in F$  στον  $s_q$ ,  $\gamma q \in D'_0$  και  $\gamma q q_1 q_2 \in D'$ .

Είναι προφανές ότι οι κανόνες του  $P$  παράγουν σε κάθε κόμβο  $n$  τις αντίστοιχες πιθανές καταστάσεις. Οι κανόνες  $s_q$  (αντίστοιχα  $s_{\gamma q q_1 q_2}^i$ ) του  $S$  προκύπτουν από τους κανόνες  $r_q$  (αντίστοιχα  $r_{\gamma q q_1 q_2}^i$ ) του  $R$  της Πρότασης 2.4.2 εισάγοντας στο σώμα τους, με κατάλληλο τρόπο, τα κατηγορήματα φίλτρου  $\text{pot-}q$  του  $P$ . Έτσι, το  $S$  παράγει σε κάθε κόμβο  $n$  ένα υποσύνολο των  $R$ -πιθανών καταστάσεων στον  $n$ .

Θα δείξουμε τώρα ότι αυτό το υποσύνολο είναι το σύνολο των επιτυχημένων καταστάσεων στον  $n$ , θα αποδείξουμε δηλαδή ότι το  $P \cup S$  είναι ένα βέλτιστο φίλτρο. Είναι προφανές ότι οι κανόνες  $s_q$  παράγουν όλες τις επιτυχημένες καταστάσεις στη ρίζα  $r$  καθώς μία πιθανή κατάσταση στην ρίζα  $r$  είναι επιτυχημένη αν είναι τελική. Για να αποδείξουμε ότι οι κανόνες  $s_{\gamma q q_1 q_2}^i$  παράγουν όλες τις επιτυχημένες καταστάσεις σε κάθε κόμβο  $n \neq r$ , αρκεί να αποδείξουμε, χρησιμοποιώντας τον Ορισμό 2.2.4, την ακόλουθη πρόταση:

«για κάθε  $\gamma q q_1 q_2 \in D'$  και κόμβο  $n$  χρώματος  $\gamma$  τ.ω. η  $q$  να είναι επιτυχημένη κατάσταση στον  $n$  και η  $q_i$  να είναι μία πιθανή κατάσταση στο παιδί  $n_i$ ,  $i = 1, 2$ , του  $n$ , ισχύει ότι η  $q_i$  είναι επιτυχημένη κατάσταση στον  $n_i$ ».

Είναι προφανές ότι υπάρχει ένα  $e$  τ.ω.  $q q_1 q_2 e \in D|_\gamma$  και η  $q q_1 q_2 e$  να είναι μία πιθανή μετάβαση στον  $n$ . Αυτό ισχύει γιατί οι  $q$ ,  $q_1$ ,  $q_2$  είναι πιθανές καταστάσεις στους κόμβους  $n$ ,  $n_1$ ,  $n_2$  αντίστοιχα και  $\gamma q q_1 q_2 \in D'$ . Αφού όμως η  $q$  είναι επιτυχημένη κατάσταση στον  $n$ , έχουμε ότι η  $q q_1 q_2 e$  είναι επιτυχημένη μετάβαση στον  $n$ , το οποίο συνεπάγεται ότι η  $q_i$  είναι επιτυχημένη κατάσταση στον  $n_i$ ,  $i = 1, 2$ , και έτσι ολοκληρώνεται η απόδειξη της παραπάνω πρότασης.

Το  $\Pi_{\phi, m}^S$  αποτελεί επέκταση του φίλτρου  $P \cup S$  με το πρόγραμμα  $A_S$ . Το τελευταίο είναι εκείνο το πρόγραμμα που προκύπτει από την αντικατάσταση κάθε ατόμου  $\text{resol-}q(x)$  στο πρόγραμμα  $A_R$  της Πρότασης 2.4.2 με το άτομο  $\text{suc-}q(x)$ . Πράγματι, δεν είναι δύσκολο να επιβεβαιωθεί ότι το  $A_S$  υπολογίζει ακριβώς τους περιορισμούς των κωδικοποιημένων αναθέσεων που ικανοποιούν τον τύπο ως άτομα  $\text{assign-}q^{|T_n|}(n, \bar{e})$ , όπου το  $\bar{e}$  είναι περιορισμός στο  $T_n$  μία κωδικοποιημένη ανάθεση που ικανοποιεί τον τύπο. Έτσι, η αποτίμηση του  $A_S$  δεν περιέχει άσκοπες στιγμιοτυποποιήσεις.

**Πολυπλοκότητα.**  $P \cup S$ : Σε κάθε φύλλο  $n$  (αντίστοιχα εσωτερικό κόμβο  $n$ ) έχουμε το πολύ  $2^k$  (αντίστοιχα  $s^2 \cdot 2^k$ ) στιγμιοτυποποιήσεις κανόνων  $p_{\gamma q}$  (αντίστοιχα  $p_{\gamma q q_1 q_2}$ ), δηλαδή τόσες όσα και τα στοιχεία  $\gamma q \in D'_0$  (αντίστοιχα  $\gamma q q_1 q_2 \in D'$ ) για δεδομένο  $\gamma = c(n)$ . Συνεπώς, ο συνολικός αριθμός των στιγμιοτυποποιήσεων του  $P$  είναι αυστηρά μικρότερος από  $m \cdot s^2 \cdot 2^k$ . Ανάλογα, ο συνολικός αριθμός στιγμιοτυποποιήσεων του  $S$  είναι αυστηρά μικρότερος από  $m \cdot s^2 \cdot 2^k$ .

$A_S$ : Ας υποθέσουμε ότι  $a = |\phi(\mathcal{T})|$  και ότι το  $a_n$  είναι ο πληθάρημος του περιορισμού του  $\phi(\mathcal{T})$  στον υποδέντρο  $\mathcal{T}_n$ . Είναι προφανές ότι  $a_n \leq a$  για κάθε  $n \in T$ . Κάθε μία από

τις  $a_n$  αναθέσεις στο υποδέντρο  $\mathcal{T}_n$  παράγεται από μία μοναδική στιγμιοτυποποίηση κανόνα του  $A_S$ . Συνεπώς ο συνολικός αριθμός ενεργών στιγμιοτυποποιήσεων του  $A_S$  είναι

$$\sum_{n \in T} a_n \leq a \cdot m.$$

Συνοψίζοντας, το ερώτημα  $(\Pi_{\phi, m}^S, \text{assign})$  υπολογίζεται μετά από το πολύ  $m \cdot (s^2 \cdot 2^{k+1} + a) + a$  στιγμιοτυποποιήσεις κανόνων.  $\square$

**Παράδειγμα 2.4.3.** (συνέχεια) Ας υποθέσουμε ότι τα  $\phi(X_1, X_2)$  και  $\mathcal{T}$  είναι όπως στο Παράδειγμα 2.2.3. Το πρόγραμμα  $\Pi_{\phi, m}^S$ , όπως επίσης και τα προγράμματα  $\Pi_{\phi, m}$  και  $\Pi_{\phi, m}^R$ , υπολογίζουν τις κωδικοποιημένες αναθέσεις  $\varepsilon_1$  και  $\varepsilon_2$  που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$  ως γεγονότα

$$\text{assign}(01, 10, 01), \quad \text{assign}(00, 10, 01) \quad (*)$$

τα οποία αποτελούν την έξοδο του ερωτήματος  $(\Pi_{\phi, m}^S, \text{assign})$  με είσοδο τη βάση δεδομένων  $\mathcal{D}_{\mathcal{T}}$ . Πιο συγκεκριμένα, οι κανόνες του  $A_S$  υπολογίζουν επαγωγικά περιορισμούς των κωδικοποιημένων αναθέσεων που ικανοποιούν τον δεδομένο τύπο με τη μορφή γεγονότων

$$\text{assign-}q_2^1(n_2, 10), \quad \text{assign-}s_2^1(n_2, 10), \quad \text{assign-}q_3^1(n_3, 01), \quad \text{assign-}s_3^1(n_3, 01),$$

και τις συνθέτουν στα γεγονότα

$$\text{assign-}q_1^3(n_1, 01, 10, 01), \quad \text{assign-}s_1^3(n_1, 00, 10, 01)$$

από τα οποία προκύπτει η έξοδος  $(*)$  μέσω των κανόνων  $a_{q_1}$  και  $a_{s_1}$  αντίστοιχα.

Μία συγκριτική παρατήρηση των Παραδειγμάτων 2.4.3 και 2.3.2 μας δείχνει την αντιστοιχία ανάμεσα στο βέλτιστο datalog πρόγραμμα  $\Pi_{\phi, m}^S$ , τον αλγόριθμο του Γιαννακάκης και τον FFG αλγόριθμο. Η αντιστοιχία αυτή διατυπώνεται στο πόρισμα που ακολουθεί το οποίο ουσιαστικά επεκτείνει το Πόρισμα 2.3.2 της προηγούμενης ενότητας.

**Πόρισμα 2.4.1.** Η μέθοδος αναγραφής με βέλτιστο φίλτρο του datalog ερωτήματος  $\mathcal{Q}_{\phi, m}$  με είσοδο τη βάση δεδομένων  $\mathcal{D}_{\mathcal{T}}$ , ο FFG αλγόριθμος με είσοδο τα  $A_{\phi}$  και  $\mathcal{T}$ , και ο αλγόριθμος του Γιαννακάκης για τον υπολογισμό του ακυκλικού συζευκτικού ερωτήματος  $\mathcal{Q}_t$  με είσοδο τη βάση δεδομένων  $g_{\phi, \mathcal{T}} = (\text{trans}_n)_{n \in T}$  συμπίπτουν. Δηλαδή, και οι τρεις αυτοί αλγόριθμοι πραγματοποιούν τους ίδιους υπολογισμούς και παράγουν την ίδια έξοδο.

**Απόδειξη.** Ανακαλώντας τα βήματα του FFG αλγορίθμου από την απόδειξη του Πορίσματος 2.3.2 ή το Παράρτημα Β', παρατηρούμε ότι το βέλτιστο φίλτρο  $P \cup S$  για το  $\mathcal{Q}_{\phi, m}$  λειτουργεί ακριβώς όπως τα πρώτα δύο βήματα του FFG αλγορίθμου. Δηλαδή υπολογίζει τις πιθανές και επιτυχημένες καταστάσεις σε κάθε κόμβο. Επιπλέον, το πρόγραμμα  $A_S$  προσομοιώνει το τελευταίο βήμα του FFG αλγορίθμου το οποίο υπολογίζει τις αναθέσεις που ικανοποιούν τον τύπο συνθέτοντας, με κατεύθυνση από τα φύλλα προς τη ρίζα, περιορισμούς των αναθέσεων αυτών στα υποδέντρα.  $\square$

### 2.4.4 Επίλυση μέσω βελτιστοποιημένης $(k + 1)$ -datalog & μοναδιαίας datalog

Στην ενότητα αυτή ορίζουμε datalog ερωτήματα που εκφράζουν το πρόβλημα αποτίμησης της MSO μέσω προγραμμάτων που πραγματοποιούν υπολογισμούς ανάλογους με εκείνους του προγράμματος  $\Pi_{\phi, m}^S$ , και με αυτή την έννοια είναι και αυτά βέλτιστα, με βασική διαφορά ότι έχουν μικρότερο βαθμό.

Αρχικά, στην Πρόταση 2.4.4, δίνουμε ένα πρόγραμμα  $\Pi_{\phi}^k$  βαθμού  $k + 1$  το οποίο, σε αντίθεση με τα προγράμματα των προηγούμενων ενοτήτων, είναι ανεξάρτητο από το μέγεθος  $m$  του δεδομένου δέντρου. Το πρόγραμμα αυτό πραγματοποιεί κατά-συνιστώσες ενώσεις περιορισμών των αναθέσεων αντί για συνενώσεις περιορισμών των κωδικοποιημένων αναθέσεων όπως κάνει το πρόγραμμα  $\Pi_{\phi, m}^S$ . Πιο συγκεκριμένα, το πρόγραμμα  $\Pi_{\phi}^k$  υπολογίζει, ως γεγονότα του κατηγορήματος-στόχου assign, τις αναθέσεις που ικανοποιούν τον  $\phi$  χρησιμοποιώντας μία επεκτεταμένη βάση δεδομένων πάνω στο δυναμοσύνολο  $\mathcal{P}(T)$  του συνόλου  $T$  των κόμβων του δεδομένου δέντρου, η οποία ουσιαστικά ορίζει τις πιθανές ενώσεις ανάμεσα στα στοιχεία του  $\mathcal{P}(T)$ .

Στη συνέχεια, στην Πρόταση 2.4.5, δίνουμε ένα μοναδιαίο datalog πρόγραμμα  $\Pi_{\phi, m}^{\text{mon}}$  που υπολογίζει τα στοιχεία των αναθέσεων που ικανοποιούν τον τύπο μέσω μιας οικογένειας κατηγορημάτων-στόχων χρησιμοποιώντας τη βάση δεδομένων  $\mathcal{D}_{\mathcal{T}}^{\text{pre}}$  που επεκτείνει τη  $\mathcal{D}_{\mathcal{T}}$  με γεγονότα που περιγράφουν την προδιατεταγμένη διάσχιση. Στην περίπτωση αυτή οι κωδικοποιημένες αναθέσεις που ικανοποιούν τον τύπο, δεν αντιστοιχούν σε ορίσματα κάποιου κατηγορήματος όπως ισχύει στο  $\Pi_{\phi, m}^S$ , αλλά αποτελούν τμήμα των συμβόλων κατηγορημάτων. Είναι δηλαδή σα να έχουμε μία «ήμι-στιγμιότυποποιημένη» εκδοχή της τελευταίας οικογένειας κανόνων του  $\Pi_{\phi, m}^S$ . Οι αναθέσεις που υπολογίζονται με αυτό τον τρόπο, στη συνέχεια αποδομούνται –πάλι σε επίπεδο κατηγορημάτων– παράγοντας τελικά τα στοιχεία  $B_i$  κάθε μιας από τις αναθέσεις  $\bar{B}$  που ικανοποιούν τον δεδομένο τύπο  $\phi$ .

Η σύγκριση των προγραμμάτων αυτής της παραγράφου με εκείνα των προηγούμενων μας οδηγεί σε ενδιαφέροντα συμπεράσματα. Παρατηρούμε λοιπόν ότι καθώς προσπαθούμε να βελτιώσουμε τα αρχικά μας προγράμματα ως προς το βαθμό τους, δηλαδή να μειώσουμε το βαθμό, άλλες παράμετροι των προγραμμάτων καθίστανται προβληματικοί. Πιο συγκεκριμένα, αυξάνει κατά εκθετικό τρόπο το μέγεθος των βάσεων δεδομένων που αποτελούν είσοδο για τα προγράμματα καθώς και τα ίδια τα προγράμματα. Αναλυτικότερα, το πρόγραμμα  $\Pi_{\phi}^k$  έχει ως είσοδο μία βάση δεδομένων πάνω στο δυναμοσύνολο του  $T$  μεγέθους  $O(2^m)$  καθώς περιέχει γεγονότα που περιγράφουν τις ενώσεις των αντίστοιχων υποσυνόλων. Από την άλλη, το πρόγραμμα  $\Pi_{\phi, m}^{\text{mon}}$  έχει παραπάνω από  $2^{k \cdot m}$  IDB κατηγορήματα καθώς χρησιμοποιεί τουλάχιστον τόσα όσες και οι πιθανές εκτελέσεις του αυτομάτου στο δεδομένο δέντρο.

**Αποτίμηση της MSO με datalog βαθμού  $k + 1$ .** Ας υποθέσουμε ότι το  $G_{\mathcal{T}}$  είναι ένα σύνολο γεγονότων που ορίζεται ως εξής:

- για κάθε φύλλο  $n$ ,

$$\text{mul-0}(\emptyset, n), \text{mul-1}(\{n\}, n) \in G_{\mathcal{T}}$$



- για κάθε εσωτερικό κόμβο  $n$  και σύνολα  $T_1 \subseteq T_{n_1}, T_2 \subseteq T_{n_2}$ ,

$$\text{sum-1}(T_1 \cup T_2 \cup \{n\}, T_1, T_2, n), \text{sum-0}(T_1 \cup T_2, T_1, T_2, n) \in G_{\mathcal{T}}.$$

Ορίζουμε την βάση δεδομένων

$$\mathcal{D}'_{\mathcal{T}} = \mathcal{D}_{\mathcal{T}} \cup G_{\mathcal{T}}.$$

**Παράδειγμα 2.4.4.** (συνέχεια) Θεωρούμε ότι το  $\mathcal{T}$  είναι όπως στο Παράδειγμα 2.2.3. Σε αυτή την περίπτωση η  $G_{\mathcal{T}}$  αποτελείται από τα ακόλουθα σύνολα γεγονότων:

- $\text{mul-0}(\emptyset, n_i), \text{mul-1}(\{n_i\}, n_i)$  για  $i = 2, 3$
- $\text{sum-0}(\emptyset, \emptyset, \emptyset, n_1), \text{sum-0}(\{n_2\}, \{n_2\}, \emptyset, n_1), \text{sum-0}(\{n_3\}, \emptyset, \{n_3\}, n_1),$   
 $\text{sum-0}(\{n_2, n_3\}, \{n_2\}, \{n_3\}, n_1) \text{sum-1}(\{n_1\}, \emptyset, \emptyset, n_1), \text{sum-1}(\{n_1, n_2\}, \{n_2\}, \emptyset, n_1),$   
 $\text{sum-1}(\{n_1, n_3\}, \emptyset, \{n_3\}, n_1), \text{sum-1}(\{n_1, n_2, n_3\}, \{n_2\}, \{n_3\}, n_1).$

**Πρόταση 2.4.4.** Για κάθε  $\text{MSO}[\tau_{\Gamma}]$  τύπο  $\phi(X_1, \dots, X_k)$ , υπάρχει ένα datalog ερώτημα  $Q_{\phi}^k = (\Pi_{\phi}^k, \text{assign})$  βαθμού  $k + 1$  τέτοιο ώστε, για κάθε  $\Gamma$ -δέντρο  $\mathcal{T}$

$$\phi(\mathcal{T}) = Q_{\phi}^k(\mathcal{D}'_{\mathcal{T}}).$$

Η πολυπλοκότητα αποτίμησης του ερωτήματος ισούται με την πολυπλοκότητα αποτίμησης του  $Q_{\phi, m}^S(\mathcal{D}_{\mathcal{T}})$ , δηλαδή είναι το πολύ  $m \cdot (s^2 \cdot 2^{k+1} + a) + a$ .

**Απόδειξη.** Το πρόγραμμα  $\Pi_{\phi}^k$  ορίζεται ως η ένωση  $P \cup S \cup A'_S$ , όπου το  $P \cup S$  είναι το βέλτιστο φίλτρο που ορίστηκε στην απόδειξη της Πρότασης 2.4.3 και το  $A'_S$  ορίζεται από τις ακόλουθες οικογένειες κανόνων για κάθε  $\gamma q e \in D_0, \gamma q q_1 q_2 e \in D, q \in Q$ , όπου  $e = e_1 \cdots e_k$ .

$$\begin{aligned} A'_S. \quad g_{\gamma q e} : \quad & \text{assign-}q(x, x_1, \dots, x_k) \leftarrow \text{suc-}q(x), \text{color-}\gamma(x), \text{leaf}(x), \\ & \text{mul-}e_1(x_1, x), \dots, \text{mul-}e_k(x_k, x) \\ g_{\gamma q q_1 q_2 e} : \quad & \text{assign-}q(x, x_1, \dots, x_k) \leftarrow \text{suc-}q(x), \text{color-}\gamma(x), \\ & \text{succ}_0(x, u_1), \text{assign-}q_1(u_1, y_1, \dots, y_k), \\ & \text{succ}_1(x, u_2), \text{assign-}q_2(u_2, z_1, \dots, z_k), \\ & \text{sum-}e_1(x_1, y_1, z_1, x), \dots, \text{sum-}e_k(x_k, y_k, z_k, x) \\ a_q : \quad & \text{assign}(x_1, \dots, x_k) \leftarrow \text{assign-}q(x, x_1, \dots, x_k), \text{root}(x) \end{aligned}$$

□

Άμεση συνέπεια της Πρότασης 2.4.4 είναι το ακόλουθο αποτέλεσμα ορισιμότητας.

**Θεώρημα 2.4.1.** Η κλάση των  $k$ -μελών  $\text{MSO}$ -ορισίμων ερωτημάτων σε δέντρα είναι ορίσιμη στην  $(k + 1)$ -datalog.

**Παράδειγμα 2.4.5.** (συνέχεια) Ας υποθέσουμε ότι τα  $\phi(X_1, X_2)$  και  $\mathcal{T}$  είναι όπως στο Παράδειγμα 2.2.3. Οι κανόνες του  $A'_S$  υπολογίζουν επαγωγικά περιορισμούς των αναθέσεων που ικανοποιούν τον δεδομένο τύπο με τη μορφή των γεγονότων

$$\text{assign-}q_2(n_2, \{n_2\}, \emptyset), \text{assign-}s_2(n_2, \{n_2\}, \emptyset), \text{assign-}q_3(n_3, \emptyset, \{n_3\}), \text{assign-}s_3(n_3, \emptyset, \{n_3\}),$$

τα οποία συνθέτουν στα γεγονότα

$$\text{assign-}q_1(n_1, \{n_2\}, \{n_1, n_3\}), \text{assign-}s_1(n_1, \{n_2\}, \{n_3\}).$$

Από τα αυτά, μέσω των κανόνων  $a_{q_1}$  και  $a_{s_1}$ , προκύπτει η έξοδος του ερωτήματος  $(\Pi_\phi^3, \text{assign})$  με είσοδο την βάση δεδομένων  $\mathcal{D}'_{\mathcal{T}}$ . Πιο συγκεκριμένα, οι κανόνες αυτοί υπολογίζουν τις αναθέσεις

$$\bar{B}_1 = (\{n_2\}, \{n_1, n_3\}), \bar{B}_2 = (\{n_2\}, \{n_3\}),$$

που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$ , με τη μορφή των γεγονότων

$$\text{assign}(\{n_2\}, \{n_1, n_3\}), \text{assign}(\{n_2\}, \{n_3\})$$

αντίστοιχα.

Αξίζει να σημειωθεί ότι με μία μικρή τροποποίηση του  $\Pi_\phi^k$  μπορούμε εύκολα να ορίσουμε ένα νέο πρόγραμμα  $\Pi_{\phi, m}^k$  βαθμού  $k$  χρησιμοποιώντας τα γεγονότα της  $\mathcal{D}_{\mathcal{T}}^{\text{pre}}$  που κωδικοποιούν την προδιατεταγμένη διάσχιση του  $\mathcal{T}$ . Πιο συγκεκριμένα, αρκεί να αντικαταστήσουμε κάθε εμφάνιση του ατόμου  $\text{assign-}q(x, x_1, \dots, x_k)$  στο σώμα ενός κανόνα με τα άτομα  $\text{assign-}q^i(x_1, \dots, x_k)$  και  $\text{pre-}i(x)$ , ενώ στην περίπτωση που το  $\text{assign-}q(x, x_1, \dots, x_k)$  εμφανίζεται στην κεφαλή το αντικαθιστούμε με το  $\text{assign-}q^i(x_1, \dots, x_k)$  και προσθέτουμε το  $\text{pre-}i(x)$  στο σώμα του κανόνα. Όπως είναι εμφανές, σε αυτή την περίπτωση το  $\Pi_{\phi, m}^k$  εξαρτάται από το  $m$ , δηλαδή ορίζεται αποκλειστικά για δέντρα μεγέθους  $m$ .

**Αποτίμηση της MSO μέσω της μοναδιαίας datalog.** Το μοναδιαίο πρόγραμμα που παρουσιάζεται παρακάτω, παράγει αρχικά για κάθε ανάθεση  $\bar{B} = (B_1, \dots, B_k)$  που ικανοποιεί τον  $\phi$  στο  $\mathcal{T}$  ένα γεγονός

$$\text{assign-}\varepsilon_{\bar{B}}(r).$$

Στη συνέχεια, λαμβάνοντας υπόψη την προδιατεταγμένη διάταξη των κόμβων του  $\mathcal{T}$ , «αποδομεί» τα ήδη υπολογισμένα γεγονότα σε νέα γεγονότα

$$j\text{-assign-}\varepsilon_{\bar{B}}(n) \text{ για κάθε } n \in B_j, j = 1, \dots, k$$

υπολογίζοντας τελικά κάθε ανάθεση  $\bar{B}$  που ικανοποιεί τον δεδομένο τύπο μέσω των στοιχείων του  $B_j$ .

**Παράδειγμα 2.4.6.** (συνέχεια) Ας ανακαλέσουμε από το Παράδειγμα 2.4.3 ότι η έξοδος του  $\Pi_{\phi,m}^S$  ως προς το assign είναι τα γεγονότα  $\text{assign}(\varepsilon_1)$  και  $\text{assign}(\varepsilon_2)$ . Το μοναδιαίο πρόγραμμα που παρουσιάζεται σε αυτή την ενότητα υπολογίζει αρχικά την ίδια έξοδο αλλά με διαφορετική μορφή, δηλαδή με τη μορφή γεγονότων

$$\text{assign-}\varepsilon_1(n_1), \text{assign-}\varepsilon_2(n_1), \text{ με } \varepsilon_1 = 01, 10, 01 \text{ και } \varepsilon_2 = 00, 10, 01$$

όπου τα  $\varepsilon_1$  και  $\varepsilon_2$  είναι οι κωδικοποιημένες αναθέσεις που ικανοποιούν τον τύπο  $\phi$  στο δεδομένο δέντρο  $\mathcal{T}$ . Στη συνέχεια το πρόγραμμα αποδομεί τα  $\varepsilon_1, \varepsilon_2$ , σε επίπεδο κατηγορημάτων πάλι, υπολογίζοντας τα ακόλουθα  $j$ -assign- $\varepsilon_1$ - και  $j$ -assign- $\varepsilon_2$ -γεγονότα,  $j = 1, 2$ :

$$2\text{-assign-}\varepsilon_1(n_1), 1\text{-assign-}\varepsilon_1(n_2), 2\text{-assign-}\varepsilon_1(n_3), 1\text{-assign-}\varepsilon_2(n_2), 2\text{-assign-}\varepsilon_2(n_3)$$

από τα οποία προσδιορίζουμε τις αναθέσεις  $\overline{B}_1 = (\{n_2\}, \{n_1, n_3\})$  και  $\overline{B}_2 = (\{n_2\}, \{n_3\})$  που αντιστοιχούν στα  $\varepsilon_1, \varepsilon_2$ .

Σημειώνουμε ότι στην πρόταση που ακολουθεί επιβάλλεται να διακρίνουμε ως ειδική περίπτωση το διάνυσμα ανάθεσης  $(\emptyset, \dots, \emptyset)$  γιατί στην αντίθετη περίπτωση το πρόγραμμα θα υπολόγιζε πάντα την  $(\emptyset, \dots, \emptyset)$  ως ανάθεση που ικανοποιεί τον δεδομένο τύπο, το οποίο, όπως είναι προφανές, δεν ισχύει για κάθε τύπο και οδηγεί σε λανθασμένη έξοδο. Θυμίζουμε ότι η σημασία των  $\overline{B}_\varepsilon$  και  $\varepsilon_{(\emptyset, \dots, \emptyset)}$  δίνεται στον Ορισμό 2.2.1.

**Πρόταση 2.4.5.** Ας υποθέσουμε ότι ο  $\phi(X_1, \dots, X_k)$  είναι ένας  $\text{MSO}[\mathcal{T}_\Gamma]$  τύπος και ότι το  $m$  είναι ένας ακέραιος. Για κάθε διάνυσμα  $(k, m)$ -ανάθεσης  $\varepsilon$ , υπάρχουν ερωτήματα

$$Q_{\phi,m}^{\varepsilon,j} = (\Pi_{\phi,m}^{\text{mon}}, j\text{-assign-}\varepsilon), \quad j = 1, \dots, k$$

της μοναδιαίας datalog τέτοια ώστε για κάθε  $\Gamma$ -δέντρο  $\mathcal{T}$  μεγέθους  $m$  να ισχύουν τα εξής:

- για κάθε  $\varepsilon \neq \varepsilon_{(\emptyset, \dots, \emptyset)}$ ,

$$\overline{B}_\varepsilon = (B_\varepsilon^1, \dots, B_\varepsilon^k) \in \phi(\mathcal{T}) \quad \text{ανν} \quad Q_{\phi,m}^{\varepsilon,j}(\mathcal{D}_{\mathcal{T}}^{\text{pre}}) = B_\varepsilon^j \text{ για κάθε } j$$

- για  $\varepsilon = \varepsilon_{(\emptyset, \dots, \emptyset)}$ ,

$$\overline{B}_\varepsilon = (\emptyset, \dots, \emptyset) \in \phi(\mathcal{T}) \quad \text{ανν} \quad Q_{\phi,m}^{\varepsilon,j}(\mathcal{D}_{\mathcal{T}}^{\text{pre}}) = \{0\} \text{ για κάθε } j.$$

Η πολυπλοκότητα είναι το πολύ  $m \cdot (s^2 \cdot 2^{k+1} + a) + a + b$ , όπου  $b \leq a \cdot k \cdot m$ .

**Απόδειξη.** Το μοναδιαίο πρόγραμμα  $\Pi_{\phi,m}^{\text{mon}}$  είναι η ένωση των προγραμμάτων  $\Pi_{\phi,m}^{\text{mon I}}$  και  $\Pi_{\phi,m}^{\text{mon II}}$ . Το πρόγραμμα  $\Pi_{\phi,m}^{\text{mon I}} = P \cup S \cup A_S^{\text{mon}}$  αποτελεί μία «μοναδιαία εκδοχή» του προγράμματος  $\Pi_{\phi,m}^S = P \cup S \cup A_S$  της Πρότασης 2.4.3, όπου το  $A_S^{\text{mon}}$  διαφέρει από το  $A_S$  στο ότι

τα διανύσματα αναθέσεων των εκτελέσεων δεν εμφανίζονται ως ορίσματα αλλά βρίσκονται κωδικοποιημένα σε νέα μοναδιαία σύμβολα κατηγορημάτων.

$$\begin{aligned}
A_S^{\text{mon}}. \quad & a_{\gamma qe}^1 : \quad \text{assign}^1\text{-}qe(x) \leftarrow \text{suc-}q(x), \text{color-}\gamma(x), \text{leaf}(x) \\
& a_{\gamma q q_1 q_2 e \alpha \beta}^\ell : \quad \text{assign}^\ell\text{-}qe\alpha\beta(x) \leftarrow \text{suc-}q(x), \text{color-}\gamma(x), \text{succ}_0(x, x_1), \text{assign}^i\text{-}q_1\alpha(x_1), \\
& \quad \quad \quad \text{succ}_1(x, x_2), \text{assign}^j\text{-}q_2\beta(x_2) \\
& a_\varepsilon : \quad \text{assign-}\varepsilon(x) \leftarrow \text{assign}^m\text{-}q\varepsilon(x)
\end{aligned}$$

όπου  $\ell = i + j + 1$  και  $3 \leq \ell \leq m$ ,  $1 \leq i, j \leq m - 2$ . Επίσης  $\gamma qe \in D_0$ ,  $\gamma q q_1 q_2 e \in D$ , και τα  $\alpha, \beta, \varepsilon$  είναι διανύσματα  $(k, i)$ -,  $(k, j)$ -,  $(k, m)$ -αναθέσεων αντίστοιχα.

Είναι προφανές ότι το γεγονός  $\text{assign}^\ell\text{-}qe(n)$  υπολογίζεται από το  $A_S^{\text{mon}}$  αν το  $A_S$  υπολογίζει, με την ίδια είσοδο, το γεγονός  $\text{assign-}q^\ell(n, \varepsilon)$ .

Έτσι, το πρόγραμμα  $\Pi_{\phi, m}^{\text{mon I}}$  υπολογίζει κάθε ανάθεση  $\bar{B}$ , που ικανοποιεί τον  $\phi$  στο  $\mathcal{T}$ , κωδικοποιημένη στο σύμβολο του IDB κατηγορήματος του γεγονότος  $\text{assign-}\varepsilon_{\bar{B}}(r)$ , όπου  $r$  είναι η ρίζα και το  $\varepsilon_{\bar{B}}$  διατάσσεται σύμφωνα με τη προδιατεταγμένη διάσχιση του  $\mathcal{T}$ .

Με το πρόγραμμα  $\Pi_{\phi, m}^{\text{mon II}}$ , κάθε διάνυσμα  $(k, m)$ -ανάθεσης  $\varepsilon$  που εμφανίζεται στα IDB κατηγορήματα  $\text{assign-}\varepsilon$  του  $A_S^{\text{mon}}$ , κωδικοποιώντας την ανάθεση  $\bar{B}_\varepsilon = (B_1, \dots, B_k)$ , αποδομείται σε  $k$  IDB κατηγορήματα

$$1\text{-assign-}\varepsilon, \dots, k\text{-assign-}\varepsilon$$

που ορίζονται ως εξής:

- Αν  $\varepsilon = \bar{e}_1, \dots, \bar{e}_m \neq \varepsilon_{(\emptyset, \dots, \emptyset)}$ , με  $\bar{e}_i = e_{i1}, \dots, e_{ik}$ ,  $i = 1, \dots, m$ , τότε
  - (i) για κάθε  $e_{ij}$ ,  $j = 1, \dots, k$ , τ.ω.  $e_{ij} = 1$ , γράφουμε τον κανόνα  $r_{\varepsilon_{ij}}$  στο  $\Pi_{\phi, m}^{\text{mon II}}$
  - (ii) για κάθε  $j = 1, \dots, k$ , γράφουμε τον κανόνα  $r_{\varepsilon_j}$  στο  $\Pi_{\phi, m}^{\text{mon II}}$ .
- Αν  $\varepsilon = \varepsilon_{(\emptyset, \dots, \emptyset)}$ , τότε για κάθε  $j = 1, \dots, k$ , γράφουμε τον κανόνα  $r_{\varepsilon_{(\emptyset, \dots, \emptyset)j}}$  στο  $\Pi_{\phi, m}^{\text{mon II}}$ .

$$\begin{aligned}
\Pi_{\phi, m}^{\text{mon II}}. \quad & r_{\varepsilon_{ij}} : \quad j\text{-assign-}\varepsilon(y) \leftarrow \text{assign-}\varepsilon(x), \text{pre-}i(y) \\
& r_{\varepsilon_j} : \quad j\text{-assign-}\varepsilon(y) \leftarrow \text{succ}_0(x, y), \text{succ}_0(y, x) \\
& r_{\varepsilon_{(\emptyset, \dots, \emptyset)j}} : \quad j\text{-assign-}\varepsilon_{(\emptyset, \dots, \emptyset)}(0) \leftarrow \text{assign-}\varepsilon_{(\emptyset, \dots, \emptyset)}(x)
\end{aligned}$$

Έτσι, για κάθε ανάθεση  $\bar{B} = (B_1, \dots, B_k) \neq (\emptyset, \dots, \emptyset)$  που ικανοποιεί τον τύπο  $\phi$ , το γεγονός  $j\text{-assign-}\varepsilon_{\bar{B}}(n_i)$  συμπεραίνεται από το  $\Pi_{\phi, m}^{\text{mon II}}$  μέσω του κανόνα  $r_{\varepsilon_{\bar{B}ij}}$  αν  $n_i \in B_j$ . Στην ειδική περίπτωση που υπάρχει κάποιο  $B_j = \emptyset$  στο  $\bar{B} = (B_1, \dots, B_j, \dots, B_k) \neq (\emptyset, \dots, \emptyset)$ ,

θα έχουμε  $e_{ij} = 0$  για κάθε  $i = 1, \dots, m$ . Τότε όμως δεν υπάρχει κάποιος κανόνας  $r_{e_{ij}}$  ο οποίος να ορίζει το  $j$ -assign- $\varepsilon_{\bar{B}}$ . Αυτή είναι η χρησιμότητα των κανόνων  $r_{\varepsilon_j}$ . Καθώς αυτοί δεν στιγμιοτυποποιούνται ποτέ, θα έχουμε ορθά ότι  $Q_{\phi, m}^{\varepsilon, j}(\mathcal{D}_{\mathcal{T}}^{\text{pre}}) = B^j = \emptyset$ .

Στην περίπτωση όπου έχουμε  $\bar{B} = (\emptyset, \dots, \emptyset)$ , κανένας από τους κανόνες  $r_{\varepsilon_{\bar{B}ij}}$  και  $r_{\varepsilon_j}$  δεν στιγμιοτυποποιείται. Έτσι, αν δεν είχαμε τους κανόνες  $r_{\varepsilon_{(\emptyset, \dots, \emptyset)j}}$ , το πρόγραμμα θα έδειχνε πάντα την ανάθεση  $(\emptyset, \dots, \emptyset)$  ως ανάθεση που ικανοποιεί τον τύπο, δηλαδή ακόμα και όταν αυτό δε θα ίσχυε.

Αυτός είναι ο λόγος που εισάγουμε αυτούς τους κανόνες και θεωρούμε την ανάθεση  $(\emptyset, \dots, \emptyset)$  ως ειδική περίπτωση. Πιο συγκεκριμένα, έχουμε ότι η  $(\emptyset, \dots, \emptyset)$  είναι μία ανάθεση που ικανοποιεί τον  $\phi$  ανν για κάθε  $j = 1, \dots, k$  το γεγονός  $j$ -assign- $\varepsilon_{(\emptyset, \dots, \emptyset)}(0)$  παράγεται από το  $\Pi_{\phi, m}^{\text{mon II}}$  μέσω του κανόνα  $r_{\varepsilon_{(\emptyset, \dots, \emptyset)j}}$ .

**Πολυπλοκότητα.** Η πολυπλοκότητα αποτίμησης του  $\Pi_{\phi, m}^{\text{mon I}}$  είναι προφανώς ίση με την πολυπλοκότητα αποτίμησης του  $\Pi_{\phi, m}^S$ , δηλαδή το πολύ  $m \cdot (s^2 \cdot 2^{k+1} + a) + a$ .

Κατά την αποτίμηση του  $\Pi_{\phi, m}^{\text{mon II}}$ , για κάθε ανάθεση  $\bar{B} = (B_1, \dots, B_k)$  που ικανοποιεί τον τύπο  $\phi$  έχουμε ακριβώς  $|\bar{B}|$  στιγμιοτυποποιήσεις κανόνων όπου  $|\bar{B}| = \sum_{i=1}^k |B_i| \leq k \cdot m$  για  $\bar{B} \neq (\emptyset, \dots, \emptyset)$ , και  $|\bar{B}| = k$  διαφορετικά. Έτσι, ο συνολικός αριθμός στιγμιοτυποποιήσεων του  $\Pi_{\phi, m}^{\text{mon II}}$  είναι

$$b = \sum_{\bar{B} \in \phi(\mathcal{T})} |\bar{B}| \leq a \cdot k \cdot m.$$

Συνοψίζοντας, η συνολική πολυπλοκότητα της αποτίμησης είναι το πολύ

$$m \cdot (s^2 \cdot 2^{k+1} + a) + a + b.$$

□

Αποδείξαμε λοιπόν ότι το πρόβλημα αποτίμησης της MSO σε πεπερασμένα δέντρα ανάγεται σε ένα πρόβλημα αποτίμησης της μοναδιαίας datalog.

### 2.4.5 Δύο ειδικές περιπτώσεις: το πρόβλημα ελέγχου μοντέλου & η αποτίμηση μονομελών ερωτημάτων

Η αποτίμηση MSO προτάσεων και MSO τύπων της μορφής  $\phi(x)$ , όπου η  $x$  είναι πρωτοβάθμια μεταβλητή, είναι εγγενώς λιγότερο σύνθετη από την αποτίμηση τύπων που ανήκουν σε ολόκληρη την MSO. Πράγματι, οι αλγόριθμοι αυτομάτων που επιλύουν το πρόβλημα αποτίμησης για αυτές τις δύο περιορισμένες κλάσεις MSO τύπων, απαιτούν αντίστοιχα μία και δύο διασχίσεις του αυτομάτου πάνω στο δεδομένο δέντρο. Θυμίζουμε ότι στη γενική λύση, που παρουσιάστηκε στις προηγούμενες ενότητες και αφορά σε οποιονδήποτε άλλο MSO τύπο, είναι πάντα απαραίτητη και μία τρίτη διάσχιση του δεδομένου δέντρου, η οποία μάλιστα είναι πολύ πιο σύνθετη από τις προηγούμενες. Η βασική αυτή διαφορά έχει ως αποτέλεσμα η λύση που παρουσιάζεται εδώ —για αυτές τις δύο κλάσεις— να έχει μία φυσική έκφραση στη μοναδιαία datalog, και τα αντίστοιχα προγράμματα να οδηγούν σε ανάλογα αποτελέσματα ορισιμότητας για τη μοναδιαία datalog.

**Το πρόβλημα ελέγχου μοντέλου – Αποτίμηση boolean MSO-ορίσιμων ερωτημάτων.** Παραθέτουμε τη λύση της datalog για το πρόβλημα ελέγχου μοντέλου της MSO.

**Πρόταση 2.4.6.** Για κάθε  $MSO[\tau_\Gamma]$  πρόταση  $\phi$  υπάρχει ένα μοναδιαίο datalog πρόγραμμα  $\Pi'_\phi$  με στόχο  $\phi\text{-sat}$ , τέτοιο ώστε για κάθε  $\Gamma$ -χρωματισμένο δέντρο  $\mathcal{T}$  μεγέθους  $m$

$$\mathcal{T} \models \phi \quad \text{ανν} \quad (\Pi'_\phi, \mathcal{D}_\mathcal{T}) \models \phi\text{-sat}.$$

Η πολυπλοκότητα της αποτίμησης είναι το πολύ  $m + 1$ .

**Απόδειξη.** Ορίζουμε το πρόγραμμα  $\Pi'_\phi = P \cup (r_q)_{q \in F}$ , όπου το  $P$  συμβολίζει την οικογένεια κανόνων που ορίστηκε στην Πρόταση 2.4.3 και η  $(r_q)_{q \in F}$  είναι η ακόλουθη οικογένεια κανόνων:

$$r_q : \quad \phi\text{-sat} \leftarrow \text{rot-}q(x), \text{root}(x).$$

Το  $P$  υπολογίζει τις πιθανές καταστάσεις στη ρίζα. Πιο συγκεκριμένα, έχουμε ότι  $(P, \mathcal{D}_\mathcal{T}) \models \text{rot-}q(r)$  ανν υπάρχει μία εκτέλεση του  $\text{Assign}_\phi$  στο  $\mathcal{T}$  που αποδίδει την κατάσταση  $q$  στη ρίζα  $r$ . Οι κανόνες  $(r_q)_{q \in F}$  ελέγχουν αν η  $q$  είναι μία τελική κατάσταση. Η θετική απάντηση σημαίνει την ύπαρξη μιας επιτυχημένης εκτέλεσης, η οποία δηλώνεται μέσω του συμπερασμού του γεγονότος  $\phi\text{-sat}$ .

**Πολυπλοκότητα.** Θυμίζουμε ότι, όταν ο  $\phi$  είναι πρόταση, το  $\text{Assign}_\phi$  είναι στην ουσία ένα ντετερμινιστικό αυτόματο (Παρατήρηση 2.2.2). Κατά συνέπεια, υπάρχει μία και μοναδική εκτέλεση. Έτσι, κατά την αποτίμηση του προγράμματος έχουμε για κάθε κόμβο ακριβώς μία στιγμιοτυποποίηση κανόνα του  $P$ , και το πολύ μία επιπλέον στιγμιοτυποποίηση ενός κανόνα  $r_q$  στη ρίζα. Η τελευταία πραγματοποιείται όταν η εν λόγω εκτέλεση είναι επιτυχημένη. Επομένως, η πολυπλοκότητα της αποτίμησης είναι το πολύ  $m + 1$ .  $\square$

**Πόρισμα 2.4.2.** Η κλάση των boolean MSO-ορίσιμων ερωτημάτων σε δέντρα είναι ορίσιμη στη μοναδιαία datalog.

**Αποτίμηση μονομελών MSO-ορίσιμων ερωτημάτων.** Θυμίζουμε ότι τα μοναδιαία MSO ερωτήματα είναι εκείνα τα ερωτήματα που ορίζονται από MSO τύπους  $\phi(x)$ , δηλαδή τύπους  $\phi(X)$  που εκφράζουν μία ιδιότητα για ένα μονοσύνολο  $X$ .

Είναι προφανές ότι για κάθε τέτοιο τύπο  $\phi$ , τα διανύσματα αναθέσεων των επιτυχημένων εκτελέσεων του  $\text{Assign}_\phi$  στο δέντρο  $\mathcal{T}$  μεγέθους  $m$  έχουν την πολύ απλή μορφή

$$(0, \dots, 0, 1, 0, \dots, 0)$$

δηλαδή όλα τα στοιχεία τους, εκτός από ένα, είναι ίσα με 0. Ας υποθέσουμε ότι τα στοιχεία των διανύσματα αναθέσεων ακολουθούν τη διάταξη  $(n_1, \dots, n_m)$  των κόμβων του  $\mathcal{T}$ . Τότε, το  $\{n_i\}$  υπολογίζεται από το  $\text{Assign}_\phi$  ανν υπάρχει μία επιτυχημένη εκτέλεση του  $\text{Assign}_\phi$  τ.ω. το μοναδικό 1 να εμφανίζεται στη θέση  $i$ . Έτσι, στην περίπτωση αυτή ο υπολογισμός του  $\phi(\mathcal{T})$  καθίσταται ιδιαίτερα απλός καθώς ισχύει η παρακάτω πρόταση.

Το  $\{n\}$  είναι μία ανάθεση που ικανοποιεί τον τύπο  $\phi(X)$ , όπου το  $X$  συμβολίζει ένα μονοσύνολο, αν και μόνο αν υπάρχει μία επιτυχημένη κατάσταση της μορφής  $(q, 1)$  στον κόμβο  $n$ .

Συνεπώς, αρκεί να ορίσουμε datalog κανόνες που i) υπολογίζουν τις επιτυχημένες καταστάσεις κάθε κόμβου και ii) επιλέγουν εκείνους τους κόμβους που έχουν μία επιτυχημένη κατάσταση της μορφής  $(q, 1)$ . Πράγματι, αν οι  $n_1, \dots, n_j$  είναι οι κόμβοι που επιλέγονται, τότε οι  $\{n_{i_1}\}, \dots, \{n_{i_j}\}$  είναι ακριβώς οι αναθέσεις που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$ . Οδηγούμαστε έτσι σε ένα βελτιστοποιημένο datalog πρόγραμμα για τα μονομελή ερωτήματα. Το πρόγραμμα αυτό, που δίνεται παρακάτω, είναι μοναδιαίο.

**Πρόταση 2.4.7.** Για κάθε  $MSO[\mathcal{T}_\Gamma]$  τύπο  $\phi(X)$ , όπου το  $X$  συμβολίζει ένα μονοσύνολο, υπάρχει ένα ερώτημα  $Q_\phi = (\Pi_\phi^1, \text{assign})$  της μοναδιαίας datalog τέτοιο ώστε για κάθε  $\Gamma$ -χρωματισμένο δέντρο  $\mathcal{T}$  να έχουμε

$$\{n\} \in \phi(\mathcal{T}) \quad \text{ανν} \quad n \in Q_\phi(\mathcal{D}_\mathcal{T}).$$

Η πολυπλοκότητα της αποτίμησης είναι το πολύ  $4 \cdot s^2 \cdot m + a$  όπου το  $s$  είναι ο πληθάριθμος του συνόλου  $Q$  των  $\phi$ -καταστάσεων, το  $m$  είναι το μέγεθος του  $\mathcal{T}$ , και  $a = |\phi(\mathcal{T})|$ .

**Απόδειξη.** Το πρόγραμμα  $\Pi_\phi^1$  είναι η ένωση μιας μικρής τροποποίησης  $P' \cup S'$  του φίλτρου  $P \cup S$ , που ορίστηκε στην απόδειξη της Πρότασης 2.4.3, και της οικογένειας κανόνων  $(r_q)_{q \in Q}$ . Αναλυτικότερα, το  $P' \cup S'$  προκύπτει από το  $P \cup S$  με κατάλληλη αντικατάσταση κάθε IDB κατηγορήματος της μορφής  $\text{pot-}q_i$  (αντίστοιχα  $\text{suc-}q_i$ ) με ένα νέο κατηγορήμα της μορφής  $\text{pot-}q_i e_i$  (αντίστοιχα  $\text{suc-}q_i e_i$ ). Πιο συγκεκριμένα έχουμε ότι ένα γεγονός  $\text{suc-}qe(n)$  συμπεραίνεται από το  $P' \cup S'$  ανν υπάρχει μία επιτυχημένη εκτέλεση που αναθέτει την κατάσταση  $qe$  στον κόμβο  $n$ .

Όταν σε ένα συμπερασμένο γεγονός  $\text{suc-}qe(n)$  έχουμε ότι το  $e = 1$ , τότε το  $\{n\}$  είναι μία ανάθεση που ικανοποιεί τον δεδομένο τύπο. Μέσω των κανόνων  $r_q$  επιλέγουμε εκείνους τους κόμβους που έχουν μία επιτυχημένη κατάσταση της μορφής  $q1$ , δηλαδή εκείνα τα  $n$  για τα οποία το  $P' \cup S'$  έχει υπολογίσει ένα γεγονός  $\text{suc-}q1(n)$  για κάποιο  $q$ .

$$\begin{aligned} P'. \quad & p'_{\gamma qe} : \quad \text{pot-}qe(x) \leftarrow \text{leaf}(x), \text{color-}\gamma(x) \\ & p'_{\gamma q q_1 q_2 e} : \quad \text{pot-}qe(x) \leftarrow \text{pot-}q_1 e_1(x_1), \text{pot-}q_2 e_2(x_2), \text{succ}_0(x, x_1), \text{succ}_1(x, x_2), \text{color-}\gamma(x) \\ S'. \quad & s'_{qe} : \quad \text{suc-}qe(x) \leftarrow \text{pot-}qe(x), \text{root}(x) \\ & s'^i_{\gamma q q_1 q_2 e} : \quad \text{suc-}q_i e_i(x_i) \leftarrow \text{suc-}qe(x), \text{pot-}q_1 e_1(x_1), \text{pot-}q_2 e_2(x_2), \text{succ}_0(x, x_1), \text{succ}_1(x, x_2), \\ & \quad \quad \quad \text{color-}\gamma(x) \quad i = 1, 2 \\ & r_q : \quad \text{assign}(x) \leftarrow \text{suc-}q1(x) \end{aligned}$$

όπου  $q \in F$  (αντίστοιχα  $q \in Q$ ) στον κανόνα  $s'_{qe}$  (αντίστοιχα  $r_q$ ). Επίσης,  $\gamma qe \in D_0$ ,  $\gamma q q_1 q_2 e \in D$ , και  $e_i \in \{0, 1\}$ ,  $i = 1, 2$ .

**Πολυπλοκότητα.** Κατά την αποτίμηση του  $P'$ , έχουμε για κάθε φύλλο ακριβώς 2 στιγμιοτυποποιήσεις, και για κάθε εσωτερικό κόμβο το πολύ  $2 \cdot s^2$  στιγμιοτυποποιήσεις. Ανάλογος είναι ο αριθμός των στιγμιοτυποποιήσεων του  $S'$ , γεγονός που προκύπτει άμεσα και μέσω της πολυπλοκότητας του  $P \cup S$  της Πρότασης 2.4.3 για  $k = 1$ . Επιπρόσθετα, έχουμε  $a = |\phi(\mathcal{T})|$  στιγμιοτυποποιήσεις κανόνων της μορφής  $r_q$ . Συνοψίζοντας, η πολυπλοκότητα αποτίμησης του προγράμματος είναι το πολύ  $4 \cdot s^2 \cdot m + a$ .  $\square$

**Πόρισμα 2.4.3.** Η κλάση των μονομελών MSO-ορίσιμων ερωτημάτων σε δέντρα είναι ορίσιμη στη μοναδιαία datalog.

**Παρατήρηση 2.4.1.** Είναι ενδιαφέρον να σημειώσουμε εδώ ότι ο η αποτίμηση των  $k$ -μελών MSO-ορίσιμων ερωτημάτων επιλογής κόμβων είναι υπολογιστικά το ίδιο δύσκολη με την αποτίμηση της πλήρους κλάσης των MSO-ορίσιμων ερωτημάτων. Θυμίζουμε ότι κάθε τέτοιο  $k$ -μελές ερώτημα ορίζεται μέσω ενός τύπου  $\phi(X_1, \dots, X_k)$ ,  $k > 1$ , όπου κάθε  $X_i$  συμβολίζει ένα μονοσύνολο. Οι αναθέσεις που ικανοποιούν έναν τέτοιο τύπο μπορούν να υπολογιστούν από το datalog ερώτημα  $Q_\phi^k = (\Pi_\phi^k, \text{assign})$  της Πρότασης 2.4.4, χρησιμοποιώντας, αντί για την εκθετικού μεγέθους βάση δεδομένων  $\mathcal{D}'_{\mathcal{T}}$  πάνω στο  $\mathcal{P}(T)$ , μία πιο απλή βάση δεδομένων  $\mathcal{D}''_{\mathcal{T}}$  πάνω στο  $T$ , μεγέθους  $O(m^2)$  όπου  $|T| = m$ . Πιο συγκεκριμένα, η  $\mathcal{D}''_{\mathcal{T}} = \mathcal{D}_{\mathcal{T}} \cup G_T$ , όπου το  $G_T$  είναι το ακόλουθο σύνολο γεγονότων:

- mul-0(0,  $n$ ), mul-1( $n$ ,  $n$ ) για κάθε φύλλο  $n$
- sum-1( $n$ , 0, 0,  $n$ ), sum-0(0, 0, 0,  $n$ ) για κάθε εσωτερικό κόμβο  $n$  & sum-0( $n'$ ,  $n'$ , 0,  $n$ ) για  $n' \in T_{n_1}$ , sum-0( $n'$ , 0,  $n'$ ,  $n$ ) για  $n' \in T_{n_2}$

δηλαδή αποτελείται από γεγονότα της μορφής sum- $e_i(x_i + y_i + e_i \cdot x, x_i, y_i, x)$  και mul- $e_i(e_i \cdot x, x)$ . Έτσι, ισχύει ότι

$$\mathcal{T} \models \phi(\{n_1\}, \dots, \{n_k\}) \quad \text{ανν} \quad Q_\phi^k(\mathcal{D}''_{\mathcal{T}}) \models \text{assign}(n_1, \dots, n_k).$$

Έπεται ότι η πολυπλοκότητα αποτίμησης του  $Q_\phi^k(\mathcal{D}''_{\mathcal{T}})$  είναι ίδια με την πολυπλοκότητα αποτίμησης του  $Q_\phi^k(\mathcal{D}'_{\mathcal{T}})$ , είναι δηλαδή το πολύ  $m \cdot (s^2 \cdot 2^{k+1} + a) + a$ .

### 2.4.6 Ορίζοντας datalog ερωτήματα μέσω του $\phi$

Όλα τα datalog προγράμματα που παρουσιάστηκαν στις προηγούμενες ενότητες ορίζονται χρησιμοποιώντας ένα αυτόματο για τον  $\phi$  και επιλύουν το πρόβλημα αποτίμησης της MSO, επιλύοντας ουσιαστικά το ισοδύναμο πρόβλημα αυτομάτων στο οποίο ανάγεται το αρχικό. Αποδεικνύουμε στην Πρόταση 2.4.8 παρακάτω ότι αυτή η αναγωγή μπορεί να εκφραστεί στην διαστρωματωμένη datalog<sup>-</sup> μέσω των προγραμμάτων  $\Pi_\phi$  και  $\Pi_\phi^\varepsilon$  που προσομοιώνουν την επαγωγική κατασκευή ενός ντετερμινιστικού αυτομάτου  $A_\phi$  για τον  $\phi$ . Επιπλέον, αποδεικνύουμε



ότι αυτά τα προγράμματα μπορούν να επεκταθούν κατάλληλα έτσι ώστε να κατασκευάζουν το ελάχιστο τέτοιο αυτόματο.

Στην Πρόταση 2.4.10 αποδεικνύουμε ότι το πρόβλημα της MSO αποτίμησης μπορεί να επιλυθεί από νέα προγράμματα τα οποία ορίζονται άμεσα από τον  $\phi$  και προκύπτουν μέσω της κατάλληλης σύνθεσης των προγραμμάτων που εκφράζουν την αναγωγή και των προγραμμάτων των προηγούμενων ενοτήτων που επιλύουν την ανηγμένη μορφή του αρχικού προβλήματος. Άμεση συνέπεια της Πρότασης 2.4.10 είναι το Θεώρημα 2.4.2 που επεκτείνει το αποτέλεσμα, που αποδείχθηκε στις προηγούμενες ενότητες, ότι το πρόβλημα της αποτίμησης της MSO πάνω σε δέντρα ανάγεται σε ένα πρόβλημα αποτίμησης για την datalog. Πιο συγκεκριμένα, αποδεικνύεται ότι υπάρχει μία λύση που βασίζεται πλήρως στην datalog για το πρόβλημα αποτίμησης της MSO σε δέντρα.

**Εκφράζοντας την κατασκευή του (ελάχιστου) αυτομάτου στην datalog.** Ορίζουμε αρχικά ένα datalog πρόγραμμα  $\Pi_\phi$  που παράγει επαγωγικά το αυτόματο ενός δεδομένου τύπου  $\phi$  και στη συνέχεια το επεκτείνουμε με κανόνες που υλοποιούν την ελαχιστοποίηση ενός αυτομάτου.

**Πρόταση 2.4.8.** Για κάθε MSO τύπο  $\phi$ , υπάρχει ένα πρόγραμμα  $\Pi_\phi$  (αντίστοιχα πρόγραμμα  $\Pi_\phi^\varepsilon$ ) που παράγει τις μεταβάσεις και τις καταστάσεις ενός ντετερμινιστικού αυτομάτου  $A_\phi$  για τον  $\phi$  με τη μορφή γεγονότων ως εξής:

- $\phi\text{-}\gamma q(e) \in \Pi_\phi(\emptyset)$  (αντίστοιχα  $\phi\text{-}\gamma qe \in \Pi_\phi^\varepsilon(\emptyset)$ ) ανν  $\gamma qe \in D_0$ ,
- $\phi\text{-}\gamma q q_1 q_2(e) \in \Pi_\phi(\emptyset)$  (αντίστοιχα  $\phi\text{-}\gamma q q_1 q_2 e \in \Pi_\phi^\varepsilon(\emptyset)$ ) ανν  $\gamma q q_1 q_2 e \in D$ ,
- $\phi\text{-state-}q \in \Pi_\phi(\emptyset)$  (αντίστοιχα  $\phi\text{-state-}q \in \Pi_\phi^\varepsilon(\emptyset)$ ) ανν  $q \in Q$ ,
- $\phi\text{-final-}q \in \Pi_\phi(\emptyset)$  (αντίστοιχα  $\phi\text{-final-}q \in \Pi_\phi^\varepsilon(\emptyset)$ ) ανν  $q \in F$ .

Τα  $\Pi_\phi$  και  $\Pi_\phi^\varepsilon$  είναι διαστρωματωμένα datalog<sup>-</sup> προγράμματα και η πολυπλοκότητα αποτίμησης τους φράσσεται από μία μη-στοιχειώδη συνάρτηση του μεγέθους του  $\phi$ .

**Απόδειξη.** Πρόγραμμα  $\Pi_\phi$ . Ορίζουμε το πρόγραμμα  $\Pi_\phi$  το οποίο προσομοιώνει την κατασκευή ενός ντετερμινιστικού αυτομάτου  $A_\phi$  από τον  $\phi$ .

Πιο συγκεκριμένα, όταν ο  $\phi$  είναι ένα ατομικός τύπος, τότε το  $\Pi_\phi$  αποτελείται από κανόνες με κενό σώμα και ορίσματα που είναι όλα σταθερές: οι κανόνες αυτοί αντιστοιχούν πλήρως στις μεταβάσεις του αυτομάτου  $A_\phi$  που δίνεται στον Πίνακα 2.2 στη σελίδα 30. Όταν ο  $\phi$  δεν είναι ατομικός, τότε το  $\Pi_\phi$  προσομοιώνει την επαγωγική κατασκευή του  $A_\phi$  από τα ντετερμινιστικά αυτόματα για τους υποτύπους του  $\phi$ , όπως αυτή δίνεται στο Παράρτημα Α'. Στην περίπτωση αυτή έχουμε ένα διαστρωματωμένο datalog<sup>-</sup> πρόγραμμα με στρώματα  $\Pi_{\phi_i}$  για κάθε  $\phi_i$  υποτύπο του  $\phi$ . Το  $\Pi_\phi$  δεν έχει εκτασιακά κατηγορήματα και κατά συνέπεια η κενή βάση δεδομένων  $\emptyset$  αποτελεί τη μοναδική είσοδό του.

Ας υποθέσουμε ότι το  $I_\phi$  είναι το σύνολο των κανόνων του  $\Pi_\phi$  με κενό σώμα και έστω ότι το  $\mathcal{E}_\phi$  είναι το σύνολο των γεγονότων που εμφανίζονται στην κεφαλή των κανόνων του  $I_\phi$ . Τότε, θέτοντας  $\Pi_\phi^- = \Pi_\phi \setminus I_\phi$ , έχουμε  $\Pi_\phi(\emptyset) = \Pi_\phi^-(\mathcal{E}_\phi)$  αν  $\Pi_\phi^- \neq \emptyset$ , και  $\Pi_\phi(\emptyset) = \mathcal{E}_\phi$  διαφορετικά.

I. Αν ο  $\phi$  είναι ατομικός τύπος, τότε διακρίνουμε τις εξής περιπτώσεις:

- Έστω ότι  $\phi(X, Y) = E(X, Y)$ .

Τότε  $Q_\phi = \{q_0, q_f, q_a\}$  και το  $\Pi_\phi$  αποτελείται από τους παρακάτω κανόνες με κενό σώμα για κάθε  $\gamma \in \Gamma$  και  $e \in \{0, 1\}$ .

$$\begin{array}{lll} \phi-\gamma q_0(0, e) \leftarrow & \phi-\gamma q_f(1, 0) \leftarrow & \phi-\gamma q_a(1, 1) \leftarrow \\ \phi-\gamma q_0 q_0 q_0(0, e) \leftarrow & \phi-\gamma q_a q_0 q_0(1, 1) \leftarrow & \phi-\gamma q_a q_a q_0(0, e) \leftarrow \\ \phi-\gamma q_a q_0 q_a(0, e) \leftarrow & \phi-\gamma q_f q_1 q_2(e_1, e_2) \leftarrow & \phi\text{-final-}q_a \leftarrow \\ \phi\text{-state-}q_0 \leftarrow & \phi\text{-state-}q_f \leftarrow & \phi\text{-state-}q_a \leftarrow \end{array}$$

Σημειώνουμε ότι οι κανόνες της μορφής  $\phi-\gamma q_f q_1 q_2(e_1, e_2) \leftarrow$  ορίζονται για κάθε τετράδα  $(q_1, q_2, e_1, e_2) \in Q_\phi \times Q_\phi \times \{0, 1\} \times \{0, 1\}$  για την οποία δεν έχουν ήδη οριστεί προηγούμενα κανόνες  $\phi-\gamma q_1 q_2(e_1, e_2) \leftarrow$ , δηλαδή για κάθε  $(q_1, q_2, e_1, e_2) \in (Q_\phi \times Q_\phi \times \{0, 1\} \times \{0, 1\}) \setminus \{(q_0, q_0, 0, 0), (q_0, q_0, 0, 1), (q_0, q_0, 1, 1), (q_a, q_0, 0, 0), (q_a, q_0, 0, 1), (q_0, q_a, 0, 1), (q_0, q_a, 0, 0)\}$ .

- Έστω ότι  $\phi(X, Y) = S_1(X, Y)$ .

Τότε  $Q_\phi = \{s_0, s_1, s_f, s_a\}$  και το  $\Pi_\phi$  αποτελείται από τους παρακάτω κανόνες με κενό σώμα για κάθε  $\gamma \in \Gamma$  και  $e \in \{0, 1\}$ .

$$\begin{array}{llll} \phi-\gamma s_0(0, 0) \leftarrow & \phi-\gamma s_1(0, 1) \leftarrow & \phi-\gamma s_f(1, e) \leftarrow & \\ \phi-\gamma s_0 s_0 s_0(0, 0) \leftarrow & \phi-\gamma s_1 s_0 s_0(0, 1) \leftarrow & \phi-\gamma s_a s_1 s_0(1, 0) \leftarrow & \\ \phi-\gamma s_a s_a s_0(0, 0) \leftarrow & \phi-\gamma s_a s_0 s_a(0, 0) \leftarrow & \phi-\gamma s_f q_1 q_2(e_1, e_2) \leftarrow & \phi\text{-final-}s_a \leftarrow \\ \phi\text{-state-}s_0 \leftarrow & \phi\text{-state-}s_1 \leftarrow & \phi\text{-state-}s_f \leftarrow & \phi\text{-state-}s_a \leftarrow \end{array}$$

Σημειώνουμε ότι οι κανόνες της μορφής  $\phi-\gamma s_f q_1 q_2(e_1, e_2) \leftarrow$  ορίζονται για κάθε τετράδα  $(q_1, q_2, e_1, e_2) \in Q_\phi \times Q_\phi \times \{0, 1\} \times \{0, 1\}$  τέτοια ώστε κανόνες της μορφής  $\phi-\gamma q_1 q_2(e_1, e_2) \leftarrow$  να μην έχουν ήδη οριστεί.

- Έστω ότι  $\phi(X) = P_\alpha(X)$ .

Τότε  $Q_\phi = \{q_0, q_f, q_a\}$  και το  $\Pi_\phi$  αποτελείται από τους παρακάτω κανόνες με κενό σώμα για κάθε  $\gamma \in \Gamma$ ,  $\alpha' \in \Gamma \setminus \{\alpha\}$  και  $e \in \{0, 1\}$ .

$$\begin{array}{llll} \phi-\alpha q_0(0) \leftarrow & \phi-\alpha q_a(1) \leftarrow & \phi-\alpha' q_0(0) \leftarrow & \phi-\alpha' q_f(1) \leftarrow \\ \phi-\alpha q_0 q_0 q_0(0) \leftarrow & \phi-\alpha q_a q_0 q_0(1) \leftarrow & \phi-\alpha' q_0 q_0 q_0(0) \leftarrow & \phi-\alpha' q_f q_0 q_0(1) \leftarrow \\ \phi-\gamma q_a q_a q_0(0) \leftarrow & \phi-\gamma q_a q_0 q_a(0) \leftarrow & \phi-\gamma q_f q_1 q_2(e) \leftarrow & \phi\text{-final-}q_a \leftarrow \\ \phi\text{-state-}q_0 \leftarrow & \phi\text{-state-}q_f \leftarrow & \phi\text{-state-}q_a \leftarrow & \end{array}$$

Σημειώνουμε ότι οι κανόνες της μορφής  $\phi-\gamma q_f q_1 q_2(e) \leftarrow$  ορίζονται για κάθε  $(q_1, q_2, e) \in Q_\phi \times Q_\phi \times \{0, 1\}$  τέτοια ώστε κανόνες της μορφής  $\phi-\gamma q_1 q_2(e) \leftarrow$  να μην έχουν ήδη οριστεί.

II. Αν ο  $\phi$  δεν είναι ατομικός, τότε ορίζουμε το  $\Pi_\phi$  με επαγωγή στη δομή του  $\phi$  ως εξής:

- Έστω ότι  $\phi(\bar{X}, \bar{Y}, \bar{Z}) = \phi_1(\bar{X}, \bar{Y}) \wedge \phi_2(\bar{X}, \bar{Z})$ , δηλαδή το  $\bar{X}$  είναι το σύνολο των κοινών μεταβλητών ανάμεσα στους τύπους  $\phi_1$  και  $\phi_2$ .

Τότε  $Q_\phi = Q_{\phi_1} \times Q_{\phi_2}$  και  $\Pi_\phi = \Pi_{\phi_1} \cup \Pi_{\phi_2} \cup R_\phi$ .

Το  $R_\phi$  αποτελείται από τις παρακάτω τέσσερις οικογένειες κανόνων για όλα τα  $\gamma \in \Gamma$  και  $(q_1, q_2), (q_{1i}, q_{2i}) \in Q_\phi, i = 1, 2$ . Σημειώνουμε ότι χρησιμοποιούμε τις συντομεύσεις  $\langle q_i q_j \rangle$  για τα  $(q_i, q_j) \in Q_\phi$  και  $\bar{q}$  για το  $q_1 q_2, q_{11} q_{21}, q_{12} q_{22}$ .

$$\begin{aligned} r_{\gamma q_1 q_2} : & \quad \phi\text{-}\gamma\langle q_1 q_2 \rangle(\bar{x}, \bar{y}, \bar{z}) \leftarrow \phi_1\text{-}\gamma q_1(\bar{x}, \bar{y}), \phi_2\text{-}\gamma q_2(\bar{x}, \bar{z}) \\ r_{\gamma \bar{q}} : & \quad \phi\text{-}\gamma\langle q_1 q_2 \rangle\langle q_{11} q_{21} \rangle\langle q_{12} q_{22} \rangle(\bar{x}, \bar{y}, \bar{z}) \leftarrow \phi_1\text{-}\gamma q_1 q_{11} q_{12}(\bar{x}, \bar{y}), \phi_2\text{-}\gamma q_2 q_{21} q_{22}(\bar{x}, \bar{z}) \\ r_{q_1 q_2} : & \quad \phi\text{-state-}\langle q_1 q_2 \rangle \leftarrow \phi_1\text{-state-}q_1, \phi_2\text{-state-}q_2 \\ f_{q_1 q_2} : & \quad \phi\text{-final-}\langle q_1 q_2 \rangle \leftarrow \phi_1\text{-final-}q_1, \phi_2\text{-final-}q_2 \end{aligned}$$

- Έστω ότι  $\phi(\bar{X}) = \exists X_{k+1} \phi_1(\bar{X}, X_{k+1})$ , δηλαδή  $\bar{X} = (X_1, \dots, X_k)$  και  $(\bar{X}, X_{k+1}) = (X_1, \dots, X_{k+1})$ .

Τότε  $Q_\phi = \{Q \mid Q \subseteq Q_{\phi_1}\}$  και  $\Pi_\phi = \Pi_{\phi_1} \cup R_\phi$  όπου  $R_\phi = R_\phi^1 \cup R_\phi^2$ . Οι κανόνες του  $R_\phi^1$  ορίζουν τις μεταβάσεις του (ενδεχομένως) μη-ντετερμινιστικού αυτομάτου  $A'_\phi$  του  $\phi$ , ενώ οι κανόνες του  $R_\phi^2$  περιγράφουν τη μετατροπή του μη-ντετερμινιστικού  $A'_\phi$  σε ένα ισοδύναμο ντετερμινιστικό αυτόματο  $A_\phi$ .

Αναλυτικότερα, το  $R_\phi^1$  αποτελείται από τις παρακάτω οικογένειες κανόνων, που γράφονται για κάθε  $q, q_i \in Q_{\phi_1}, \gamma \in \Gamma$ .

$$\begin{aligned} r_{\gamma q} : & \quad \phi'\text{-}\gamma q(\bar{x}) \leftarrow \phi_1\text{-}\gamma q(\bar{x}, x_{k+1}) \\ r_{\gamma q q_1 q_2} : & \quad \phi'\text{-}\gamma q q_1 q_2(\bar{x}) \leftarrow \phi_1\text{-}\gamma q q_1 q_2(\bar{x}, x_{k+1}) \end{aligned}$$

Το πρόγραμμα  $R_\phi^2$  αποτελείται από τις τέσσερις οικογένειες κανόνων που δίνονται παρακάτω: η οικογένεια  $r_{\gamma Q}$  κανόνων του  $R_\phi^2$  γράφεται για κάθε  $Q = \{q_1, \dots, q_m\} \subseteq Q_{\phi_1}$  τέτοιο ώστε  $Q_{\phi_1} \setminus Q = \{q_{m+1}, \dots, q_{m+n}\}$ ,  $\gamma \in \Gamma$ . Η οικογένεια  $r_{\gamma Q Q_1 Q_2}$  γράφεται για όλα τα σύνολα  $Q \times Q_1 \times Q_2 = \{(q_1, q_1^1, q_1^2), \dots, (q_m, q_m^1, q_m^2)\} \subseteq Q_{\phi_1} \times Q_{\phi_1} \times Q_{\phi_1}$  και  $(Q_{\phi_1} \times Q_{\phi_1} \times Q_{\phi_1}) \setminus (Q \times Q_1 \times Q_2) = \{(q_{m+1}, q_{m+1}^1, q_{m+1}^2), \dots, (q_{m+n}, q_{m+n}^1, q_{m+n}^2)\}$ . Οι κανόνες  $r_Q$  γράφονται για κάθε  $Q \in Q_\phi$  και οι κανόνες  $r_{Qq}$  για κάθε  $Q \in Q_\phi$  και  $q \in Q$ .

$$r_{\gamma Q} : \quad \phi\text{-}\gamma Q(\bar{x}) \leftarrow \phi'\text{-}\gamma q_1(\bar{x}), \dots, \phi'\text{-}\gamma q_m(\bar{x}),$$

$$\begin{aligned}
r_{\gamma Q Q_1 Q_2} : & \quad \neg\phi' - \gamma q_{m+1}(\bar{x}), \dots, \neg\phi' - \gamma q_{m+n}(\bar{x}) \\
& \quad \phi - \gamma Q Q_1 Q_2(\bar{x}) \leftarrow \phi' - \gamma q_1 q_1^1 q_1^2(\bar{x}), \dots, \phi' - \gamma q_m q_m^1 q_m^2(\bar{x}), \\
& \quad \neg\phi' - \gamma q_{m+1} q_{m+1}^1 q_{m+1}^2(\bar{x}), \dots, \neg\phi' - \gamma q_{m+n} q_{m+n}^1 q_{m+n}^2(\bar{x}) \\
r_Q : & \quad \phi\text{-state-}Q \leftarrow \\
r_{Qq} : & \quad \phi\text{-final-}Q \leftarrow \phi\text{-state-}Q, \phi_1\text{-final-}q
\end{aligned}$$

Οι παραπάνω κανόνες του  $R_\phi^2$  προσομοιώνουν την εξής κατασκευή δυναμοσυνόλου: ως θεωρήσουμε ότι

$$A'_\phi = \{\Gamma \times \{0, 1\}^k, Q_{\phi_1}, \Delta_0, \Delta, F_{\phi_1}\}.$$

Τότε, για το ισοδύναμο με το  $A'_\phi$  ντετερμινιστικό αυτόματο

$$A_\phi = \{\Gamma \times \{0, 1\}^k, Q_\phi, \delta_0, \delta, F_\phi\}$$

έχουμε

$$\begin{aligned}
\delta_0(\gamma\bar{e}) &= \{q \mid (\gamma\bar{e}, q) \in \Delta_0\} = Q_{\gamma\bar{e}} \\
\delta(Q_1, Q_2, \gamma\bar{e}) &= \{q \mid \text{υπάρχουν } q_1 \in Q_1, q_2 \in Q_2 \text{ τ.ω. } (q_1, q_2, \gamma\bar{e}, q) \in \Delta\} = Q_{\gamma\bar{e}Q_1Q_2}
\end{aligned}$$

Έτσι, συγκεκριμένοι κανόνες της μορφής  $r_{\gamma Q}$  παράγουν γεγονότα  $\phi - \gamma Q_{\gamma\bar{e}}(\bar{e})$ , για όλα τα  $\gamma\bar{e} \in \Gamma \times \{0, 1\}^k$  και συγκεκριμένοι κανόνες της μορφής  $r_{\gamma Q Q_1 Q_2}$  παράγουν γεγονότα  $\phi - \gamma Q_{\gamma\bar{e}Q_1 Q_2} Q_1 Q_2(\bar{e})$ , για κάθε  $(Q_1, Q_2, \gamma\bar{e}) \in Q_{\phi_1} \times Q_{\phi_1} \times \Gamma \times \{0, 1\}^k$ .

- Έστω ότι  $\phi = \neg\phi_1$ .

Τότε,  $Q_\phi = Q_{\phi_1}$  και  $\Pi_\phi = \Pi_{\phi_1} \cup R_\phi$ .

Το πρόγραμμα  $R_\phi$  αποτελείται από τις παρακάτω οικογένειες κανόνων για κάθε  $q, q_i \in Q_\phi$ .

$$\begin{aligned}
\phi - \gamma q(\bar{x}) &\leftarrow \phi_1 - \gamma q(\bar{x}) & \phi\text{-state-}q &\leftarrow \phi_1\text{-state-}q \\
\phi - \gamma q q_1 q_2(\bar{x}) &\leftarrow \phi_1 - \gamma q q_1 q_2(\bar{x}) & \phi\text{-final-}q &\leftarrow \neg\phi_1\text{-final-}q
\end{aligned}$$

*Πρόγραμμα  $\Pi_\phi^\varepsilon$ .* Το πρόγραμμα  $\Pi_\phi^\varepsilon$  δεν έχει EDB κατηγορήματα και τα IDB κατηγορήματά του είναι μηδενικού βαθμού. Πιο συγκεκριμένα, το  $\Pi_\phi^\varepsilon$  μπορεί να ανακτηθεί από το σύνολο των ενεργών στιγμιοτυποποιήσεων του  $(\Pi_\phi, \emptyset)$ , αρκεί κάθε όρισμα να γίνει τμήμα του αντίστοιχου συμβόλου κατηγορήματος. Για παράδειγμα, όταν ο  $\phi$  είναι ατομικός τύπος, κάθε κανόνας του  $\Pi_\phi$  που έχει τη μορφή  $\phi - \gamma q(e_1, e_2) \leftarrow$ , όπου  $e_1, e_2 \in \{0, 1\}$ , αντικαθίσταται στο  $\Pi_\phi^\varepsilon$  από τον κανόνα  $\phi - \gamma q e_1 e_2 \leftarrow$ . Το  $\Pi_\phi^\varepsilon$  κατασκευάζεται, ανεξάρτητα από το  $\Pi_\phi$ , ως εξής: όταν ο  $\phi$  δεν είναι ατομικός και έχει ως υποτύπους τα  $\phi_i$ , τότε το πρόγραμμα  $\Pi_\phi^\varepsilon$  κατασκευάζεται επαγωγικά από τα προγράμματα  $\Pi_{\phi_i}^\varepsilon$  με τρόπο ανάλογο με αυτόν που το  $\Pi_\phi$  κατασκευάζεται από τα  $\Pi_{\phi_i}$ .

*Πολυπλοκότητα.* Ο αριθμός των ενεργών στιγμιοτυποποιήσεων του  $(\Pi_\phi, \emptyset)$  (αντίστοιχα του  $(\Pi_\phi^\varepsilon, \emptyset)$ ) φράσσεται από το μέγεθος του ντετερμινιστικού αυτομάτου που ορίζεται στην απόδειξη του θεωρήματος των Doner - Thatcher & Wright και το οποίο αποτελεί μη-στοιχειώδη συνάρτηση του μεγέθους του τύπου  $\phi$ .  $\square$

**Πρόταση 2.4.9.** Το πρόγραμμα  $\Pi_\phi$  μπορεί να επεκταθεί έτσι ώστε να παράγει τις μεταβάσεις του ελάχιστου ντετερμινιστικού αυτομάτου για τον  $\phi$ .

**Απόδειξη.** Δεδομένου του προγράμματος  $\Pi_\phi$  της Πρότασης 2.4.8, που παράγει ένα ντετερμινιστικό αυτόματο για δέντρα  $A_\phi$ , ορίζουμε ένα διαστρωματωμένο datalog<sup>-</sup> πρόγραμμα  $M_\phi$  τέτοιο ώστε το πρόγραμμα  $\Pi_\phi \cup M_\phi$  να παράγει το ελάχιστο αυτόματο του  $\phi$ , δηλαδή ένα ντετερμινιστικό αυτόματο ισοδύναμο με το  $A_\phi$  αλλά με τον ελάχιστο δυνατό αριθμό καταστάσεων.

Έστω ότι το  $S_\phi = \{q_1, \dots, q_n\}$  είναι το σύνολο των προσβάσιμων καταστάσεων του αυτομάτου  $A_\phi$ . Θυμίζουμε ότι μία κατάσταση  $q$  του  $A_\phi$  ονομάζεται προσβάσιμη αν υπάρχει τουλάχιστον ένα δέντρο  $\mathcal{T}$  τέτοιο ώστε η εκτέλεση  $\rho$  του  $A_\phi$  στο  $\mathcal{T}$  να απεικονίζει τη ρίζα στην  $q$ . Μπορούμε, χωρίς απώλεια της γενικότητας, να υποθέσουμε ότι όλες οι καταστάσεις είναι προσβάσιμες, δηλαδή ότι  $S_\phi = Q_\phi$ . Ας θεωρήσουμε ότι το  $S_\phi$  είναι διατεταγμένο ως εξής:  $q_1 < \dots < q_n$ . Τότε, ορίζουμε το σύνολο  $S'_\phi \subset S_\phi \times S_\phi$  με τον ακόλουθο τρόπο

$$S'_\phi = \{(q_1, q_2), \dots, (q_1, q_n), (q_2, q_3), \dots, (q_2, q_n), \dots, (q_i, q_{i+1}), \dots, (q_i, q_n), \dots, (q_{n-1}, q_n)\}$$

δηλαδή  $S'_\phi = \cup_{1 \leq i \leq n-1} I_i$ , όπου  $I_i = \{(q_i, q_{i+1}), (q_i, q_{i+2}), \dots, (q_i, q_n)\}$ .

Το πρόγραμμα  $M_\phi$  αποτελείται από δύο στρώματα I, II τα οποία ορίζονται παρακάτω.

- Στρώμα I. Κατασκευή ζευγαριών διαχωρίσιμων καταστάσεων του  $S_\phi$  ως προς το  $A_\phi$ : δύο καταστάσεις  $p, q$  ονομάζονται διαχωρίσιμες (distinguishable) ως προς το  $A_\phi$  αν υπάρχει ένα δέντρο  $\mathcal{T}$  με ρίζα  $r$  τέτοιο ώστε αν σε ένα από τα φύλλα του –ας πούμε στο φύλλο  $n_i$ – αποδίδεται κατάσταση  $p$  τότε το  $A_\phi$  αποδίδει στη ρίζα  $r$  μία τελική (αντίστοιχα μία μη τελική) κατάσταση, ενώ αν η  $q$  αποδοθεί στο  $n_i$  τότε μία μη τελική (αντίστοιχα τελική) κατάσταση αποδίδεται στη ρίζα  $r$ . Οι διαχωρίσιμες καταστάσεις παράγονται μέσω των παρακάτω κανόνων: για κάθε  $(p, q) \in S'_\phi$  γράφουμε τους εξής κανόνες

$$\text{dish-}pq \leftarrow \text{final-}p, \neg \text{final-}q$$

$$\text{dish-}pq \leftarrow \neg \text{final-}p, \text{final-}q$$

και για κάθε  $(p, q) \in S'_\phi$  και  $(\gamma, r, p, p') \in \Gamma \times S_\phi^3$ ,  $(\gamma, s, q, p') \in \Gamma \times S_\phi^3$ ,  $(\gamma, r, p', p) \in \Gamma \times S_\phi^3$  και  $(\gamma, s, p', q) \in \Gamma \times S_\phi^3$  γράφουμε τους εξής κανόνες

$$\text{dish-}pq \leftarrow \text{dish-}rs, \phi\text{-}\gamma r p p'(\bar{x}), \phi\text{-}\gamma s q p'(\bar{x})$$

$$\begin{aligned} \text{dish-}pq &\leftarrow \text{dish-}sr, \phi\text{-}\gamma rpp'(\bar{x}), \phi\text{-}\gamma sqp'(\bar{x}) \\ \text{dish-}pq &\leftarrow \text{dish-}rs, \phi\text{-}\gamma rp'p(\bar{x}), \phi\text{-}\gamma sp'q(\bar{x}) \\ \text{dish-}pq &\leftarrow \text{dish-}sr, \phi\text{-}\gamma rp'p(\bar{x}), \phi\text{-}\gamma sp'q(\bar{x}) \end{aligned}$$

- Στρώμα II. α. Κατασκευή των κλάσεων ισοδυναμίας που αποτελούν τις καταστάσεις του ελάχιστου αυτομάτου. Ονομάζουμε δυο καταστάσεις  $p$  και  $q$  ισοδύναμες, και το συμβολίζουμε με  $p \equiv q$ , όταν αυτές δεν είναι διαχωρίσιμες. Η σχέση  $\equiv$  είναι μία σχέση ισοδυναμίας πάνω στο  $S_\phi$  και οι κλάσεις ισοδυναμίας ως προς την  $\equiv$  είναι οι καταστάσεις του ελάχιστου αυτομάτου. Αυτές οι κλάσεις ορίζονται από τις παρακάτω οικογένειες κανόνων: για κάθε κατάσταση  $q_i$  του  $S_\phi$ , γράφουμε τους κανόνες

$$\begin{aligned} \text{C-}q_i &\leftarrow \text{dish-}q_1q_i, \dots, \text{dish-}q_{i-1}q_i, \\ &\quad \text{dish-}q_iq_{i+1}, \dots, \text{dish-}q_iq_n \\ \text{m-state-}q_i &\leftarrow \text{C-}q_i \end{aligned}$$

και για κάθε υποσύνολο  $S = \{s_1, \dots, s_j\}$  του  $S_\phi$  τ.ω.  $s_1 < \dots < s_j$  και  $S_\phi \setminus S = \{s_{j+1}, \dots, s_n\}$  τ.ω.  $s_{j+1} < \dots < s_n$  γράφουμε τους κανόνες

$$\begin{aligned} \text{C-}s_1 \dots s_j &\leftarrow \neg \text{dish-}s_1s_2, \dots, \neg \text{dish-}s_1s_j, \\ &\quad \text{dish-}s_1s_{j+1}, \dots, \text{dish-}s_1s_n \\ \text{m-state-}s_1 \dots s_j &\leftarrow \text{C-}s_1 \dots s_j \end{aligned}$$

β. Το ελάχιστο αυτόματο. Ορίζουμε αρχικά τους κανόνες που παράγουν τις τελικές καταστάσεις του ελάχιστου αυτομάτου: για κάθε κατάσταση  $q$  στο  $S_\phi$ , γράφουμε τον κανόνα

$$\text{m-final-}q \leftarrow \text{m-state-}q, \text{final-}q.$$

Στη συνέχεια, γράφουμε κανόνες που παράγουν τις μεταβάσεις του ελάχιστου αυτομάτου: ας υποθέσουμε ότι το  $[p]$  συμβολίζει την κλάση ισοδυναμίας μιας κατάστασης  $p \in S_\phi$  και ότι οι  $\delta'_0, \delta'$  είναι οι συναρτήσεις μετάβασης του ελάχιστου αυτομάτου. Τότε έχουμε  $\delta'_0(\gamma) = [\delta'_0(\gamma)]$  και  $\delta'([p_1], [p_2], \gamma) = [\delta(p_1, p_2, \gamma)]$ . Έτσι, για κάθε  $(s, s_1, s_2) \in S_\phi \times S_\phi \times S_\phi$  και κάθε υποσύνολο  $P = \{p, p_1, \dots, p_i\}$  του  $S_\phi$  που περιέχει την  $s$  και για το οποίο ισχύει  $p < p_1 < \dots < p_i$ , γράφουμε έναν κανόνα

$$\text{m-}\phi\text{-}\gamma ps_1s_2(\bar{x}) \leftarrow \text{C-}pp_1 \dots p_i, \text{m-state-}s_1, \text{m-state-}s_2, \phi\text{-}\gamma ss_1s_2(\bar{x})$$

και για κάθε κατάσταση  $s \in S_\phi$  και κάθε υποσύνολο  $P$  όπως το παραπάνω, γράφουμε έναν κανόνα

$$\text{m-}\phi\text{-}\gamma p(\bar{x}) \leftarrow \text{C-}pp_1 \dots p_i, \phi\text{-}\gamma s(\bar{x}). \quad \square$$

**Η άμεση αναγωγή.** Είμαστε πλέον έτοιμοι να ορίσουμε την datalog λύση απευθείας μέσω του δεδομένου τύπου  $\phi$ , δηλαδή χωρίς να απαιτείται να έχουμε δεδομένο το αυτόματο  $\text{Assign}_\phi$  όπως ίσχυε για τις λύσεις που παρουσιάστηκαν στις προηγούμενες ενότητες.

**Πρόταση 2.4.10.** Ας υποθέσουμε ότι το  $\Pi$  είναι ένα από τα προγράμματα  $\Pi_{\phi,m}$  ή  $\Pi_{\phi,m}^R$  ή  $\Pi_{\phi,m}^S$  ή  $\Pi_\phi^k$  ή  $\Pi_{\phi,m}^{\text{mon}}$  ή  $\Pi'_\phi$  ή  $\Pi_\phi^1$  που ορίστηκαν στις Προτάσεις 2.4.1, 2.4.2, 2.4.3, 2.4.4, 2.4.5, 2.4.6, 2.4.7 αντίστοιχα. Υπάρχει ένα  $\text{datalog}^-$  πρόγραμμα  $\tilde{\Pi}$ , ισοδύναμο με το  $\Pi$  ως προς το κατηγορημα-στόχο, το οποίο ορίζεται άμεσα από τον τύπο  $\phi$ .

Αν το  $C_\Pi$  είναι ο αριθμός των ενεργών στιγμιοτυποποιήσεων του  $\Pi$ , τότε έχουμε

$$C_{\tilde{\Pi}} = C_\Pi + f(\phi)$$

όπου η  $f$  είναι μία μη-στοιχειώδης συνάρτηση του μεγέθους του τύπου  $\phi$ .

**Απόδειξη.** Έστω ότι το  $\Pi$  είναι είτε το  $\Pi_{\phi,m}$  ή το  $\Pi_{\phi,m}^R$  ή το  $\Pi_{\phi,m}^S$  ή το  $\Pi'_\phi$  που ορίστηκαν στις Προτάσεις 2.4.1, 2.4.2, 2.4.3, 2.4.6 αντίστοιχα. Στην περίπτωση αυτή έχουμε ότι  $\tilde{\Pi} = \Pi_\phi \cup \Pi'$ , όπου το  $\Pi_\phi$  είναι το πρόγραμμα της Πρότασης 2.4.8 ενώ για το  $\Pi'$  ισχύουν όσα ακολουθούν.

Αν  $\Pi = \Pi_{\phi,m}$ , τότε το  $\Pi'$  ορίζεται ως εξής

$$\begin{aligned} \Pi'_{\phi,m} \cdot \quad r_{\gamma q}^{\prime 1} &: \quad \text{assign-}q^1(x, y) \leftarrow \text{color-}\gamma(x), \text{leaf}(x), \phi\text{-}\gamma q(y) \\ r_{\gamma q q_1 q_2}^{\prime i+j+1} &: \quad \text{assign-}q^{i+j+1}(x, y, \bar{u}, \bar{v}) \leftarrow \text{color-}\gamma(x), \text{succ}_0(x, x_1), \text{assign-}q_1^i(x_1, \bar{u}), \\ & \quad \text{succ}_1(x, x_2), \text{assign-}q_2^j(x_2, \bar{v}), \phi\text{-}\gamma q q_1 q_2(y) \\ r'_q &: \quad \text{assign}(y) \leftarrow \text{assign-}q^m(x, y), \phi\text{-final-}q \end{aligned}$$

όπου  $\gamma \in \Gamma$ ,  $q, q_1, q_2 \in Q_\phi$  και τα  $i, j$  είναι περιττοί αριθμοί τ.ω.  $1 \leq i, j \leq m - 2$  και  $3 \leq i + j + 1 \leq m$ . Τα κατηγορήματα  $\phi\text{-}\gamma q$  (αντίστοιχα  $\phi\text{-}\gamma q q_1 q_2$ ) βαθμού 1 αντιστοιχούν στα κατηγορήματα  $\phi\text{-}\gamma q$  (αντίστοιχα  $\phi\text{-}\gamma q q_1 q_2$ ) βαθμού  $k$  του προγράμματος  $\Pi_\phi$ . Πιο συγκεκριμένα, αυτά ορίζονται εύκολα από κανόνες που εμφανίζουν στο σώμα τους άτομα  $w(x, \bar{x})$ , όπου  $\bar{x} = (x_1, \dots, x_k)$ , βαθμού  $k + 1$  τα οποία στιγμιοτυποποιούνται μέσω της βάσης δεδομένων

$$W_k = \{(e, \bar{e}) \mid \bar{e} \in \{0, 1\}^k\}$$

που περιγράφει τη μετατροπή κάθε  $k$ -διανύσματος  $\bar{e} = e_1, \dots, e_k$  σε μία λέξη  $e = e_1 \dots e_k$ . Για κάθε ενεργή στιγμιοτυποποίηση ενός κανόνα  $r_{\gamma q e}^1$  (αντίστοιχα  $r_{\gamma q q_1 q_2 e}^\ell$ ) του προγράμματος  $\Pi_{\phi,m}$  υπάρχει μία ενεργή στιγμιοτυποποίηση του αντίστοιχου κανόνα  $r_{\gamma q}^{\prime 1}$  (αντίστοιχα  $r_{\gamma q q_1 q_2}^{\prime \ell}$ ) του προγράμματος  $\Pi'_{\phi,m}$  όπου κάθε μεταβλητή  $y$  αντικαθίσταται από τη σταθερά  $e$ .

Αν το  $\Pi$  είναι είτε το  $\Pi_{\phi,m}^R$  ή το  $\Pi_{\phi,m}^S$  ή το  $\Pi'_\phi$ , οι κανόνες του  $\Pi'$  είναι παρόμοιοι με τους κανόνες του  $\Pi$ . Έτσι, παρόλο που προφανώς το  $\Pi'$  μπορεί να οριστεί ανεξάρτητα από το  $\Pi$ , δηλαδή όπως το  $\Pi'_{\phi,m}$  παραπάνω, επιλέγουμε έναν πιο εύκολο τρόπο να περιγράψουμε

την κατασκευή του  $\Pi'$  μέσω του προγράμματος  $\Pi$ . Αρκεί για κάθε οικογένεια κανόνων  $R_q$  του  $\Pi$  που ορίζεται για  $q \in F$ , να κατασκευάσουμε μία νέα οικογένεια κανόνων  $R'_q$  που προκύπτει από την αρχική αν προσθέσουμε το γεγονός  $\phi$ -final- $q$  στο σώμα των αντίστοιχων κανόνων. Επίσης, για κάθε μία από τις οικογένειες κανόνων  $R_{\gamma q e}^1$  και  $R_{\gamma q}$  του  $\Pi$ , θα πρέπει να κατασκευάσουμε αντίστοιχα νέες οικογένειες κανόνων  $R_{\gamma q}^1$  και  $R'_{\gamma q}$ , όπου  $\gamma \in \Gamma$  και  $q \in Q_\phi$ , που προκύπτουν από τους αρχικούς κανόνες με την προσθήκη στο σώμα τους των ατόμων  $\phi$ - $\gamma q(y)$ . Τέλος, για κάθε μία από τις οικογένειες κανόνων  $R_{\gamma q_1 q_2 e}^\ell$  και  $R_{\gamma q_1 q_2}$  του  $\Pi$ , κατασκευάζουμε αντίστοιχα νέες οικογένειες κανόνων  $R_{\gamma q_1 q_2}^\ell$  και  $R'_{\gamma q_1 q_2}$ , όπου  $\gamma \in \Gamma$  και  $q, q_1, q_2 \in Q_\phi$ , που προκύπτουν από τους αρχικούς κανόνες με την προσθήκη στο σώμα τους των ατόμων  $\phi$ - $\gamma q_1 q_2(y)$ .

Ας υποθέσουμε τώρα ότι το  $\Pi$  είναι ένα από τα  $\Pi_\phi^k$  ή  $\Pi_{\phi, m}^{\text{mon}}$  ή  $\Pi_\phi^1$  που ορίστηκαν στις Προτάσεις 2.4.4, 2.4.5, 2.4.7 αντίστοιχα. Τότε  $\tilde{\Pi} = \Pi_\phi^\varepsilon \cup \Pi'$ , όπου το  $\Pi_\phi^\varepsilon$  είναι το πρόγραμμα της Πρότασης 2.4.8, ενώ η κατασκευή του  $\Pi'$  από το  $\Pi$  είναι ανάλογη των βημάτων που περιγράφηκαν παραπάνω. Πιο συγκεκριμένα, καθώς το  $e$  δεν αποτελεί πλέον σταθερά του προγράμματος αλλά είναι τμήμα των IDB κατηγορημάτων, κατασκευάζουμε νέους κανόνες με την προσθήκη στο σώμα των αρχικών κανόνων γεγονότων της μορφής  $\phi$ - $\gamma q e$  και  $\phi$ - $\gamma q_1 q_2 e$ .

*Πολυπλοκότητα.* Είναι προφανές ότι, για κάθε βάση δεδομένων  $\mathcal{D}$  που αποτελεί είσοδο για τα προγράμματά μας, ο αριθμός  $C_\Pi$  των ενεργών στιγμιτυποποιήσεων του  $(\Pi, \mathcal{D})$  είναι ίσος με τον αριθμό  $C_{\Pi'}$  των ενεργών στιγμιτυποποιήσεων του  $(\Pi', \mathcal{D} \cup \Pi_\phi(\emptyset))$ . Επίσης, από την Πρόταση 2.4.8, έχουμε ότι ο αριθμός των ενεργών στιγμιτυποποιήσεων του  $(\Pi_\phi, \emptyset)$  είναι  $f(\phi)$ , όπου η  $f$  μια μη-στοιχειώδης συνάρτηση του μεγέθους του  $\phi$ . Επομένως,  $C_{\tilde{\Pi}} = C_\Pi + f(\phi)$ .  $\square$

Θυμίζουμε ότι ο ορισμός της MSO χωρίς άρνηση δίνεται στη σελίδα 19.

**Θεώρημα 2.4.2.** Το πρόβλημα αποτίμησης της MSO σε πεπερασμένα δέντρα ανάγεται άμεσα σε ένα πρόβλημα αποτίμησης της datalog για τους MSO τύπους χωρίς άρνηση, και σε ένα πρόβλημα αποτίμησης της διαστρωματωμένης datalog<sup>-</sup> διαφορετικά.

**Απόδειξη.** Το πρόβλημα αποτίμησης της MSO για κάθε τύπο  $\phi$  ανάγεται άμεσα σε ένα πρόβλημα αποτίμησης της διαστρωματωμένης datalog<sup>-</sup> μέσω του προγράμματος

$$\Pi_\phi \cup \Pi_\phi^{k'}$$

όπου το  $\Pi_\phi^{k'}$  είναι το πρόγραμμα της Πρότασης 2.4.10 που προκύπτει από το πρόγραμμα  $\Pi_\phi^k$  της Πρότασης 2.4.4.

Ειδικά όταν ο  $\phi$  είναι ένας τύπος στον οποίο δεν εμφανίζεται άρνηση, το πρόβλημα αποτίμησης της MSO μπορεί να αναχθεί άμεσα σε ένα πρόβλημα αποτίμησης της datalog μέσω του προγράμματος

$$\Pi'_\phi \cup \Pi_\phi^{k'}.$$

Το πρόγραμμα  $\Pi'_\phi$  ορίζεται μόνο για τύπους χωρίς άρνηση μέσω μιας μικρής τροποποίησης του επαγωγικού ορισμού του προγράμματος  $\Pi_\phi$  της Πρότασης 2.4.8 για την περίπτωση των



τύπων της μορφής  $\phi(\bar{X}) = \exists X_{k+1} \phi_1(\bar{X}, X_{k+1})$ . Στην περίπτωση αυτή ορίζουμε το  $\Pi'_\phi = \Pi'_{\phi_1} \cup R'_\phi$ , όπου με  $R'_\phi = R_{\phi_1}^1 \cup S_\phi$ . Το  $R_{\phi_1}^1$  προκύπτει από το  $R_{\phi_1}^1$  αντικαθιστώντας τα άτομα  $\phi' \cdot \gamma q(\bar{x})$ ,  $\phi' \cdot \gamma q q_1 q_2(\bar{x})$  με τα άτομα  $\phi \cdot \gamma q(\bar{x})$ ,  $\phi \cdot \gamma q q_1 q_2(\bar{x})$  αντίστοιχα. Το  $S_\phi$  αποτελείται από τους κανόνες  $\phi$ -state- $q \leftarrow \phi_1$ -state- $q$  και  $\phi$ -final- $q \leftarrow \phi_1$ -final- $q$  για κάθε  $q \in Q_{\phi_1}$ . Ουσιαστικά αφαιρούμε τους κανόνες που μετατρέπουν το αυτόματο σε ντετερμινιστικό. Η αλλαγή αυτή θα δώσει ένα ορθό αυτόματο καθώς ο τύπος μας δεν εμφανίζει άρνηση.  $\square$

**Η πολυπλοκότητα αποτίμησης της MSO μέσω της datalog.** Ολοκληρώνουμε αυτή την ενότητα με το παρακάτω αποτέλεσμα πολυπλοκότητας.

**Πόρισμα 2.4.4.** Το πρόβλημα αποτίμησης της MSO για έναν τύπο  $\phi(X_1, \dots, X_k)$  και ένα δέντρο  $\mathcal{T}$  μπορεί να επιλυθεί μέσω ενός datalog<sup>-</sup> προγράμματος που αποτιμάται πραγματοποιώντας το πολύ

$$m \cdot (f(\phi) + a)$$

στιγμιοτυποποιήσεις κανόνων, όπου  $m = |\mathcal{T}|$ ,  $a = |\phi(\mathcal{T})|$  και η  $f$  είναι μία μη-στοιχειώδης συνάρτηση του μεγέθους του τύπου  $\phi$ .



### 3.

---

## Προβλήματα Αποτίμησης για τη Μοναδιαία Δευτεροβάθμια Λογική σε Δομές Φραγμένου Δεντροπλάτους

Στο κεφάλαιο αυτό γενικεύουμε τις λύσεις του προβλήματος αποτίμησης της MSO που βασίζονται στα αυτόματα και στην datalog, και παρουσιάστηκαν στο προηγούμενο κεφάλαιο για την περίπτωση των δέντρων, στις δομές φραγμένου δεντροπλάτους. Επιπρόσθετα, επεκτείνουμε την προσέγγισή μας έτσι ώστε αυτή να επιλύει MSO-ορίσιμα προβλήματα βελτιστοποίησης.

Πιο συγκεκριμένα, εισάγουμε αρχικά την έννοια των αυτόματων αποδόμησης τα οποία μπορούν να ειδωθούν ως γενίκευση της έννοιας των αυτομάτων αναθέσεων που ορίστηκε στο προηγούμενο κεφάλαιο. Αποδεικνύουμε ότι τα αυτόματα αυτά, όταν εκτελούνται πάνω σε δεντροδιασπάσεις μιας δεδομένης δομής, υπολογίζουν άμεσα τις λύσεις των αντίστοιχων προβλημάτων αποτίμησης της MSO. Η απόδειξη είναι κατασκευαστική —με την έννοια ότι δίνονται αναλυτικοί ορισμοί των αυτομάτων αποδόμησης για κάθε MSO τύπο— και αποτελεί μία διαφορετική απόδειξη της επέκτασης του θεωρήματος του Courcelle που δίνεται στο [FFG02]. Θυμίζουμε ότι στην απόδειξη του [FFG02] ο αρχικός MSO τύπος, μετατρέπεται σε έναν ισοδύναμο τύπο —πολύ πιο σύνθετο όμως και κατά συνέπεια με πολύ πιο σύνθετη κατασκευή αυτομάτου— πάνω στα δέντρα: η λύση στο αρχικό πρόβλημα αποτίμησης δίνεται μέσω της αναγωγής του σε ένα πρόβλημα αποτίμησης της MSO σε δέντρα το οποίο στη συνέχεια επιλύεται μέσω της αναγωγής του σε πρόβλημα αυτομάτων. Αντίθετα, στην προσέγγιση που παρουσιάζεται εδώ, όπου η κατασκευή των αυτομάτων αποδόμησης γίνεται απευθείας από τον αρχικό MSO τύπο, παρέχουμε μία *άμεση αναγωγή του αρχικού προβλήματος αποτίμησης της MSO σε ένα πρόβλημα αποτίμησης για τα αυτόματα αποδόμησης*.

Όπως κάναμε και στο προηγούμενο κεφάλαιο, χρησιμοποιούμε μεθόδους βελτιστοποίησης της datalog για να αποτιμήσουμε αποτελεσματικά τα αυτόματα αποδόμησης. Δίνουμε έτσι βελτιστοποιημένες λύσεις που βασίζονται στην datalog για τα αρχικά προβλήματα αποτίμησης

της MSO που θεωρούμε. Οι datalog λύσεις που παρουσιάζουμε, επεκτείνουν μη-τετριμμένα τη datalog λύση του [GPW10], που αφορούσε αποκλειστικά στην αποτίμηση MSO τύπων  $\phi(x)$  με  $x$  πρωτοβάθμια μεταβλητή, σε όλους τους MSO τύπους  $\phi(x_1, \dots, x_\ell, Y_1, \dots, Y_k)$ . Τα αυτόματα αποδόμησης, ακριβώς όπως και τα αυτόματα αναθέσεων, είναι μη-ντετερμινιστικά αυτόματα για δέντρα που υπολογίζουν τις αναθέσεις κωδικοποιημένες στις επιτυχημένες εκτελέσεις τους. Συνεπώς, η λύση, μέσω της datalog, για το πρόβλημα αποτίμησης της MSO τόσο στα δέντρα όσο και στις δομές φραγμένου δεντροπλάτους διαθέτει στην ουσία τον ίδιο βασικό πυρήνα. Έτσι, μπορούμε να ανακτήσουμε μία datalog λύση για το πρόβλημα στις δομές φραγμένου δεντροπλάτους μέσω μιας κατάλληλης προσαρμογής και επέκτασης της αντίστοιχης λύσης στα δέντρα. Οι διαφορές προέρχονται από το γεγονός ότι τώρα έχουμε να διαχειριστούμε ως είσοδο δύο βάσεις δεδομένων, μία που αντιστοιχεί στη δομή και μία που αντιστοιχεί στη δεντροδιάσπαση. Κατά συνέπεια, θα πρέπει επιπρόσθετα i) να ορίσουμε στην datalog το δέντρο που αποτελεί είσοδο για τα αυτόματά μας και ii) να ενσωματώσουμε έναν ιδιαίτερα σύνθετο μηχανισμό αποκωδικοποίησης που έχουν τα αυτόματα αποδόμησης. Αξίζει εδώ να σημειωθεί ότι οι λόγοι που επιτυγχάνουμε μία άμεση λύση που βασίζεται στην datalog είναι i) ο ορισμός των αυτόματων αποδόμησης –με δομικό παράγοντα τη συνάρτηση κωδικοποίησής τους– και ο ορισμός της αντίστοιχης αναγωγής του αρχικού προβλήματος μέσω αυτών των αυτομάτων, και ii) η κατάλληλη αναπαράσταση της δεδομένης δεντροδιάσπασης ως βάση δεδομένων.

Με τα datalog προγράμματα που δίνουμε, αποδεικνύουμε ότι *κάθε  $k$ -μελές MSO-ορισμό ερώτημα είναι ορισμό στην  $(k + 1)$ -datalog πάνω σε δομές φραγμένου δεντροπλάτους*. Γενικεύουμε με αυτόν τον τρόπο το αντίστοιχο αποτέλεσμα του προηγούμενου κεφαλαίου για την περίπτωση των δέντρων, και επεκτείνουμε το αποτέλεσμα του [GPW10] που αφορά στην ορισμότητα των μονομελών MSO-ερωτημάτων στην μοναδιαία datalog πάνω σε δομές φραγμένου δεντροπλάτους. Θυμίζουμε ότι στην Πρόταση 2.4.8 του προηγούμενου κεφαλαίου, αποδείξαμε ότι η κατασκευή αυτομάτων μπορεί να εκφραστεί πλήρως στην datalog. Κατά συνέπεια, συνδυάζοντας εκείνο το αποτέλεσμα, καταλήγουμε ότι η datalog λύση που δίνουμε στα προβλήματα αυτού του κεφαλαίου αποτελεί ουσιαστικά *μία λύση που βασίζεται αποκλειστικά στις βάσεις δεδομένων*.

Τέλος, θεωρούμε ανεξάρτητα την περίπτωση των MSO-ορισμών προβλημάτων βελτιστοποίησης πάνω σε δομές φραγμένου δεντροπλάτους. Πιο συγκεκριμένα, δίνουμε και για αυτά τα προβλήματα *βελτιστοποιημένες λύσεις που βασίζονται στη datalog*. Οι λύσεις αυτές είναι το αποτέλεσμα της εφαρμογής της μεθόδου φιλτραρίσματος που βασίζεται στην επίλυση. Τα φίλτρα εδώ, σε αντίθεση με την περίπτωση των δέντρων, έχουν πολύ πιο σύνθετους ορισμούς. Μάλιστα, η γενική λύση για αυτά τα προβλήματα βελτιστοποίησης εκφράζεται μέσω της επέκτασης της datalog με *συναθροιστικές συναρτήσεις*. Ενδιαφέροντα πορίσματα της datalog επίλυσης των προβλημάτων αποτίμησης που μελετάμε, είναι αντίστοιχα αποτελέσματα ορισμότητας.

Καταλήγουμε το κεφάλαιο με την εφαρμογή της προσέγγισής μας στην επίλυση του VERTEX COVER και των συναφών προβλημάτων βελτιστοποίησης.

Σε όλα τα προβλήματα που εξετάζουμε σε αυτό το κεφάλαιο, εκτός από έναν MSO τύπο

$\phi$  και μία πεπερασμένη δομή  $\mathcal{A}$ , θεωρούμε ως μέρος της εισόδου μία δεντροδιάσπαση  $\mathcal{I}$  της  $\mathcal{A}$ .

### 3.1 Προκαταρκτικά

Δίνουμε τους ορισμούς των βασικών εννοιών που εισάγονται σε αυτό το κεφάλαιο. Παραπέμπουμε τον αναγνώστη στην ενότητα με τα προκαταρκτικά του προηγούμενου κεφαλαίου, Ενότητα 2.1, για τους ορισμούς που έχουν ήδη χρησιμοποιηθεί εκεί όπως, για παράδειγμα, αυτούς που αφορούν στα χρωματισμένα δυαδικά δέντρα, το πρόβλημα αποτίμησης της MSO, τα MSO-ορίσιμα ερωτήματα.

**Δομές & δεντροδιασπάσεις** Μία (σχεσιακή) υπογραφή  $\tau$  είναι ένα πεπερασμένο σύνολο από σύμβολα κατηγορημάτων  $R_1, \dots, R_\ell$ . Σε κάθε σύμβολο κατηγορήματος  $R_i$ , για  $i \in \{1, \dots, \ell\}$ , αντιστοιχεί ένας φυσικός αριθμός  $a_i \geq 0$  που ονομάζεται ο βαθμός του. Μία  $\tau$ -δομή  $\mathcal{A}$  είναι ένα ζευγάρι  $(A, (R_i^{\mathcal{A}})_{i \in \{1, \dots, \ell\}})$ , όπου το  $A$  είναι ένα μη-κενό σύνολο που ονομάζεται το *σύμπαν* της δομής, και κάθε  $R_i^{\mathcal{A}}$  είναι μία σχέση βαθμού  $r_i$  πάνω στο  $A$ . Μία δομή  $\mathcal{A}$  είναι *πεπερασμένη* αν το  $A$  είναι ένα πεπερασμένο σύνολο.

Μία *δεντροδιάσπαση* μιας  $\tau$ -δομής  $\mathcal{A}$  είναι ένα ζευγάρι  $(\mathcal{T}, (A_n)_{n \in T})$  όπου το  $\mathcal{T}$  είναι ένα δέντρο με σύνολο κόμβων  $T$ , και κάθε  $A_n$ , που ονομάζεται *σάκος* (bag) της δεντροδιάσπασης, είναι ένα υποσύνολο του σύμπαντος  $A$  της  $\mathcal{A}$  τέτοιο ώστε

- (i) για κάθε  $a \in A$ , το σύνολο  $T_a = \{n \in T \mid a \in A_n\}$  να είναι μη-κενό και συνδεδεμένο στο  $\mathcal{T}$ , και
- (ii) για κάθε  $(a_1, \dots, a_{r_i}) \in R_i^{\mathcal{A}}$ ,  $1 \leq i \leq \ell$ , να υπάρχει ένα  $n \in T$  τέτοιο ώστε  $\{a_1, \dots, a_{r_i}\} \subseteq A_n$ .

Δηλαδή αντιστοιχεί σε μία δεντροδιάσπαση του υπεργράφου  $\mathcal{H}_{\mathcal{A}}$  της  $\mathcal{A}$  που ορίζεται από το ζευγάρι  $(A, \{\{a_1, \dots, a_r\} \mid \text{υπάρχει } R \in \tau \text{ βαθμού } r \text{ τ.ω. } (a_1, \dots, a_r) \in R^{\mathcal{A}}\})$ .

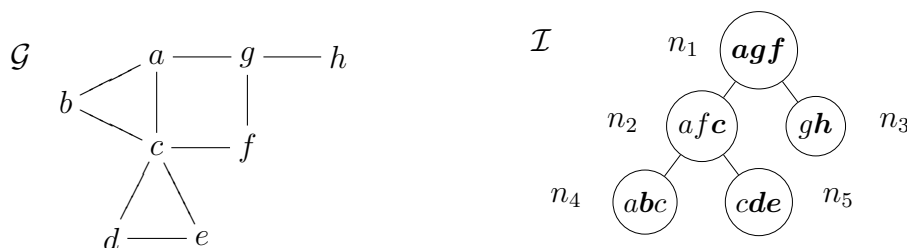
Το *πλάτος* μιας δεντροδιάσπασης  $(\mathcal{T}, (A_n)_{n \in T})$  είναι ίσο με τον αριθμό

$$\max\{|A_n| \mid n \in T\} - 1.$$

Το *δεντροπλάτος*  $\text{tw}(\mathcal{A})$  της δομής  $\mathcal{A}$  είναι το ελάχιστο ανάμεσα στα πλάτη των δεντροδιασπάσεων της  $\mathcal{A}$ .

Είναι πιο βολικό στο πλαίσιο μας να αναπαριστούμε τους σάκους των δεντροδιασπάσεων ως πλειάδες αντί για σύνολα. Έτσι, χρησιμοποιούμε μία ελάχιστη διαφοροποιημένη μορφή για τη δεντροδιάσπαση, που ονομάζεται *ειδική δεντροδιάσπαση* και θεωρεί διατεταγμένα τα στοιχεία των σάκων. Θεωρούμε επίσης ότι το πλάτος μιας ειδικής δεντροδιάσπασης μιας δομής ισούται με το δεντροπλάτος της. Αναλυτικότερα, ας υποθέσουμε ότι η  $\mathcal{A}$  είναι μία δομή που έχει δεντροπλάτος  $w$ , και ότι το  $\bar{a}_n$  συμβολίζει μία  $u$ -πλειάδα  $(a_n^1, \dots, a_n^u)$ ,  $1 \leq u \leq w + 1$ , με στοιχεία από το  $A$ . Μία ειδική δεντροδιάσπαση  $\mathcal{I}$  της  $\mathcal{A}$  ορίζεται από ένα ζευγάρι

$$(\mathcal{T}, (\bar{a}_n)_{n \in T})$$



Σχήμα 3.1: Ο γράφος  $\mathcal{G}$  και μία ειδική δέντροδιάσπαση  $\mathcal{I}$  του  $\mathcal{G}$ .

όπου το  $\mathcal{T}$  είναι ένα δυαδικό δέντρο και η  $(\mathcal{T}, (a_n)_{n \in \mathcal{T}})$ ,  $a_n = \{a_n^1, \dots, a_n^u\}$ , είναι μία δέντροδιάσπαση τη  $\mathcal{A}$  πλάτους  $w$ .

Είναι προφανές ότι ο μετασχηματισμός μιας δεδομένης δέντροδιάσπασης σε μία ειδική δέντροδιάσπαση με το ίδιο πλάτος πραγματοποιείται σε γραμμικό χρόνο.

**Παράδειγμα 3.1.1.** Ας υποθέσουμε ότι ο  $\mathcal{G}$  είναι μία  $E$ -δομή που αντιστοιχεί σε στον μη-κατευθυνόμενο γράφο του Σχήματος 3.1 με σύνολο κορυφών  $V = \{a, b, c, d, e, f, g, h\}$ . Δηλαδή το  $E$  είναι ένα δυαδικό σύμβολο κατηγορήματος που ερμηνεύεται ως η συμμετρική σχέση ακμών και έχουμε  $E^{\mathcal{G}} = \{(a, b), (a, c), (a, g), (b, c), (c, d), (c, e), (c, f), (d, e), (f, g), (g, h), (b, a), \dots, (h, g)\}$ . Η δέντροδιάσπαση  $\mathcal{I}$  που δίνεται στο Σχήμα 3.1 είναι μία ειδική δέντροδιάσπαση πλάτους 2 της  $\mathcal{G}$ . Σημειώνουμε ότι τα στοιχεία των σάκων θεωρούνται διατεταγμένα σύμφωνα με τη σειρά εμφάνισής τους.

**MSO-ορίσιμα προβλήματα βελτιστοποίησης.** Ονομάζουμε MSO-ορίσιμα προβλήματα βελτιστοποίησης πάνω σε πεπερασμένες δομές φραγμένου δεντροπλάτους, τα εξής τρία προβλήματα αποτίμησης που ορίζονται ως προς MSO τύπους  $\phi(X)$  σε τέτοιες δομές:

1. το  $k$ -card-MSO πρόβλημα ελέγχου μοντέλου, δηλαδή το πρόβλημα απόφασης για το αν, για έναν MSO-τύπο  $\phi(X)$  και μία δομή  $\mathcal{A}$ , το σύνολο  $\{B \in \phi(\mathcal{A}) \mid |B| \leq k\}$  είναι κενό ή όχι
2. το min-card-MSO (αντίστοιχα max-card-MSO) πρόβλημα αποτίμησης, δηλαδή το πρόβλημα υπολογισμού, για έναν MSO-τύπο  $\phi(X)$  και μία δομή  $\mathcal{A}$ , του πληθάριθμου εκείνων των στοιχείων του  $\phi(\mathcal{A})$  που έχουν ελάχιστο (αντίστοιχα μέγιστο) μέγεθος και
3. το min-MSO (αντίστοιχα max-MSO) πρόβλημα αποτίμησης, δηλαδή το πρόβλημα υπολογισμού, για έναν MSO-τύπο  $\phi(X)$  και μία δομή  $\mathcal{A}$ , των στοιχείων του  $\phi(\mathcal{A})$  με ελάχιστο (αντίστοιχα μέγιστο) μέγεθος.

Θα λέμε ότι ένα ερώτημα  $q$  πάνω σε μία κλάση  $\mathcal{C}$  πεπερασμένων δομών είναι *min-MSO-ορίσιμο* πάνω στη  $\mathcal{C}$  αν για κάθε δομή  $\mathcal{A} \in \mathcal{C}$  υπάρχει ένας MSO τύπος  $\phi(X)$  τέτοιος ώστε η τιμή του  $q$  στην  $\mathcal{A}$  να συμπίπτει με τη λύση στο min-MSO πρόβλημα αποτίμησης για τον

$\phi$  και την  $\mathcal{A}$ . Ανάλογοι ορισμοί ισχύουν για κάθε μία από τις περιπτώσεις που θεωρήθηκαν παραπάνω.

## 3.2 Η σχέση MSO–αυτομάτων στις δομές φραγμένου δεντροπλάτους

Στην ενότητα αυτή παρουσιάζουμε μία άμεση αναγωγή του προβλήματος αποτίμησης της MSO πάνω σε δομές φραγμένου δεντροπλάτους, σε ένα ισοδύναμο πρόβλημα αποτίμησης για αυτόματα δέντρων. Τα βασικά σημεία αυτής της αναγωγής είναι i) ο ορισμός μιας δεντρικής αναπαράστασης της δεδομένης δομής και ii) ο ορισμός μιας κατάλληλης κωδικοποίησης των αναθέσεων πάνω σε αυτή τη δεντρική αναπαράσταση της δομής. Η δεντρική αναπαράσταση τη δεδομένης δομής βασίζεται στη δεντροδιάσπαση, η οποία επεκτείνεται κατάλληλα έτσι ώστε να εμπεριέχει τις σχέσεις της δομής. Οι λεπτομέρειες της κατασκευής αυτής της επέκτασης παρουσιάζονται στην Ενότητα 3.2.2. Πριν από αυτή την παρουσίαση, δείχνουμε στην Ενότητα 3.2.1 πώς οι αναθέσεις μπορούν να κωδικοποιηθούν ως προς τις δεντροδιασπάσεις.

Τα αυτόματα αποδόμησης αποτελούν ένα είδος αυτομάτων που υπολογίζουν αναθέσεις, όπως δηλαδή κάνουν και τα αυτόματα αναθέσεων του προηγούμενου κεφαλαίου. Πιο συγκεκριμένα, καθώς εκτελούνται σε δεντρικές αναπαραστάσεις των δομών, παράγουν αναθέσεις κωδικοποιημένες ως προς τις δεντροδιασπάσεις. Αποδεικνύουμε, στη Ενότητα 3.2.3, ότι για κάθε MSO τύπο  $\phi$  υπάρχει ένα αυτόματο αποδόμησης που υπολογίζει τις αναθέσεις που ικανοποιούν τον  $\phi$  πάνω σε οποιαδήποτε δομή φραγμένου δεντροπλάτους. Συνεπώς, επιλύοντας το πρόβλημα αποτίμησης για τα αυτόματα αποδόμησης έχουμε άμεσα τη λύση του αρχικού προβλήματος αποτίμησης της MSO.

### 3.2.1 Κωδικοποίηση αναθέσεων: διαίσθηση & τυπικοί ορισμοί

Ας δούμε τώρα πώς τα αυτόματα αναθέσεων του προηγούμενου κεφαλαίου, τα οποία εκτελούνται πάνω σε δέντρα και επιστρέφουν πλειάδες από σύνολα κόμβων των δέντρων αυτών, μπορούν να γενικευθούν στην περίπτωση των δεντροδιασπάσεων. Διαισθητικά θα λέγαμε ότι αρκεί κάθε δεντροδιάσπαση να ειδωθεί ως ένα δέντρο με ιδιαίτερης μορφής κόμβους, δηλαδή με κόμβους που είναι σύνολα στοιχείων. Το βασικό σημείο αυτής της γενίκευσης είναι ο ορισμός μιας κατάλληλης κωδικοποίησης των αναθέσεων με τη μορφή μιας ολικής απεικόνισης που έχει πεδίο ορισμού τους κόμβους της δεντροδιάσπασης.

Προτού δώσουμε τους τυπικούς ορισμούς, ας δούμε διαισθητικά το πρόβλημα ορισμού της κατάλληλης κωδικοποίησης το οποίο αποτελεί τη βάση του ορισμού των αυτομάτων για δεντροδιασπάσεις. Ο αναγνώστης, ειδικά ο εξοικειωμένος με το αντικείμενο, μπορεί να παρακάμψει το διαισθητικό κομμάτι και να περάσει απευθείας στον Ορισμό 3.2.2.

Ας ανακαλέσουμε από τον Ορισμό 2.2.1 του προηγούμενου κεφαλαίου ότι στην περίπτωση των δέντρων η κωδικοποίηση μιας ανάθεσης  $\bar{B} = (B_1, \dots, B_k) \in \mathcal{P}(T)^k$  γίνεται μέσω της ολικής απεικόνισης  $\varepsilon$  από το σύνολο  $T$  των κόμβων του δέντρου στο σύνολο  $\{0, 1\}^k$ , έτσι

ώστε  $\varepsilon(n) = (e_1, \dots, e_k)$  με  $e_i = 1$  αν  $n \in B_i$ ,  $1 \leq i \leq k$ . Δηλαδή το 1 στη θέση  $i$  του  $\varepsilon(n)$  υποδεικνύει τη συμμετοχή του κόμβου  $n$  στο σύνολο  $B_i$  του  $\overline{B}$ . Για παράδειγμα, η κωδικοποίηση της ανάθεσης  $\overline{B} = (\{b, c\}, \{a, b\})$  με τη μορφή απεικόνισης  $T \mapsto \{0, 1\}^2$ ,  $T = \{a, b, c\}$  δίνεται στο Σχήμα 3.2(α'). Σημειώνουμε ότι για λόγους απλοποίησης αναπαριστούμε συχνά ένα διάνυσμα μέσω της αντίστοιχης λέξης.

Ας θεωρήσουμε τώρα ένα ιδιαίτερο δέντρο  $\mathcal{T}^*$  όπου κάθε ένας από τους κόμβους του είναι της μορφής

$$\overline{n} = (a_1, \dots, a_\ell) \in A^\ell, \quad 1 \leq \ell \leq w + 1$$

δηλαδή κάθε κόμβος είναι ένα διατεταγμένο σύνολο από  $\ell$  στοιχεία του σύμπαντος  $A$ . Είναι προφανές ότι αυτός είναι ένας διαφορετικός τρόπος να δούμε μία δεντροδιάσπαση πλάτους  $w$ , δηλαδή ως δέντρο με κόμβους τους διατεταγμένους σάκους της. Δηλαδή το σύνολο κόμβων  $\overline{T}$  αυτού του νέου δέντρου που θεωρούμε είναι υποσύνολο του  $\cup_{\ell=1}^{w+1} A^\ell$ .

Μία άμεση προσαρμογή της ιδέας της κωδικοποίησης που χρησιμοποιήσαμε στα δέντρα του προηγούμενου κεφαλαίου οδηγεί στην κωδικοποίηση μιας ανάθεσης  $\overline{B} = (B_1, \dots, B_k) \in (\mathcal{P}(A))^k$  μέσω μιας ολικής απεικόνισης  $\varepsilon_{\overline{B}}^*$  πάνω στο  $\overline{T}$  που ορίζεται ως εξής: απεικονίζουμε κάθε κόμβο  $\overline{n}$  σε μία  $k$ -πλειάδα  $\varepsilon_{\overline{B}}^*(\overline{n}) = (\epsilon_1, \dots, \epsilon_k)$  αλλά τώρα κάθε  $\epsilon_i$ ,  $1 \leq i \leq k$ , είναι ένα  $\ell$ -διάνυσμα πάνω στο  $\{0, 1\}$ . Οι θέσεις του  $\epsilon_i$  στις οποίες εμφανίζεται το 1 υποδεικνύουν ποια στοιχεία του  $\overline{n}$  περιέχονται στο  $B_i$ . Πιο συγκεκριμένα, το 1 εμφανίζεται στη θέση  $j$  του  $\epsilon_i$  αν  $a_j \in B_i$ . Συνοψίζοντας, κάθε ανάθεση

$$\overline{B} = (B_1, \dots, B_k) \in (\mathcal{P}(A))^k$$

κωδικοποιείται ως απεικόνιση

$$\varepsilon_{\overline{B}}^* : \overline{T} \rightarrow \bigcup_{\ell=1}^{w+1} \{0, 1\}^{\ell \cdot k}$$

τέτοια ώστε για κάθε  $\overline{n} = (a_1, \dots, a_\ell)$  να έχουμε  $\varepsilon_{\overline{B}}^*(\overline{n}) = (\epsilon_1, \dots, \epsilon_k)$ ,  $\epsilon_i = (e_i^1, \dots, e_i^\ell)$ , όπου

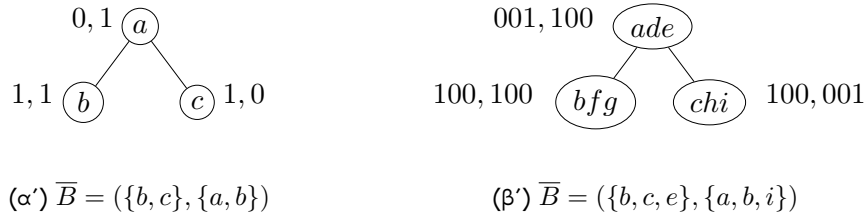
$$e_i^j = 1 \quad \text{αν} \quad a_j \in B_i, \quad 1 \leq i \leq k, \quad 1 \leq j \leq \ell. \quad (*)$$

Το Σχήμα 3.2(β') παρουσιάζει ένα παράδειγμα αυτής της κωδικοποίησης. Είναι προφανές ότι όταν έχουμε  $\ell = 1$  σε κάθε κόμβο, τότε παίρνουμε ακριβώς την κωδικοποίηση που χρησιμοποιούμε στα αυτόματα αναθέσεων.

Ο παραπάνω ορισμός της κωδικοποίησης του  $\overline{B} = (B_1, \dots, B_k)$  μέσω της απεικόνισης  $\varepsilon_{\overline{B}}^*$  έχει το εξής χαρακτηριστικό: αν υποθέσουμε ότι στο δέντρο  $\mathcal{T}^*$ , κάποιο στοιχείο  $a \in A$  εμφανίζεται σε περισσότερους από έναν κόμβους του  $\mathcal{T}^*$  και ότι το  $a \in B_i$  για κάποιο  $i$ . Στην περίπτωση αυτή, το γεγονός ότι το  $a$  ανήκει στο  $B_i$  υποδεικνύεται στο  $\varepsilon_{\overline{B}}^*(\overline{n})$  τόσες



Σχήμα 3.2: Η φυσική επέκταση της κωδικοποίησης αναθέσεων στις δεντροδιασπάσεις.



φορές όσες ο αριθμός των διαφορετικών εμφανίσεων του  $a$  στους κόμβους του δέντρου  $\mathcal{T}^*$ . Για παράδειγμα, αν το  $a$  εμφανίζεται σε δύο κόμβους  $\bar{n}_1$  και  $\bar{n}_2$  τότε θα υπάρχουν δυο εμφανίσεις του 1 στις αντίστοιχες θέσεις των διανυσμάτων  $\varepsilon_{\bar{B}}^*(\bar{n}_1)$  και  $\varepsilon_{\bar{B}}^*(\bar{n}_2)$  οι οποίες θα υποδεικνύουν ακριβώς τον ίδιο γεγονός, δηλαδή ότι « $a \in B_i$ ».

Προκειμένου να αποφύγουμε αυτές τις πολλαπλές ενδείξεις για το ίδιο γεγονός κατά την κωδικοποίηση μιας ανάθεσης  $\bar{B}$  με τη μορφή διανύσματος, επιλέγουμε έναν αντιπροσωπευτικό κόμβο για κάθε  $a \in A$ , ανάμεσα από τους ενδεχομένως πολλούς κόμβους του  $\mathcal{T}^*$  όπου εμφανίζεται το  $a$ , και δηλώνουμε το γεγονός ότι το  $a$  ανήκει σε κάποιο  $B_i$  μόνο μέσω αυτού του κόμβου. Πιο συγκεκριμένα, ορίζουμε τον αντιπροσωπευτικό κόμβο του στοιχείου  $a \in A$  ως προς το δέντρο  $\mathcal{T}^*$ , και τον συμβολίζουμε με  $\text{node}(a)$ , ως τον κόμβο που περιέχει το στοιχείο  $a$  και επιπρόσθετα έχει το μέγιστο ύψος από όλους τους κόμβους που περιέχουν το  $a$ . Σημειώνουμε ότι ένας τέτοιος κόμβος είναι μονοσήμαντα ορισμένος ως προς το  $\mathcal{T}^*$ . Έτσι, το προβληματικό σημείο (\*) του ορισμού της απεικόνισης  $\varepsilon_{\bar{B}}^*$  που κωδικοποιεί την  $\bar{B}$  εξαλείφεται, μέσω της έννοιας του αντιπροσωπευτικού κόμβου, ως εξής: έχουμε  $\varepsilon_{\bar{B}}^*(\bar{n}) = (\epsilon_1, \dots, \epsilon_k)$ ,  $\epsilon_i = (e_i^1, \dots, e_i^\ell)$ , όπου

$$e_i^j = 1 \quad \text{ανν} \quad a_j \in B_i \quad \& \quad \text{node}(a_j) = \bar{n}, \quad 1 \leq i \leq k, \quad 1 \leq j \leq \ell. \quad (**)$$

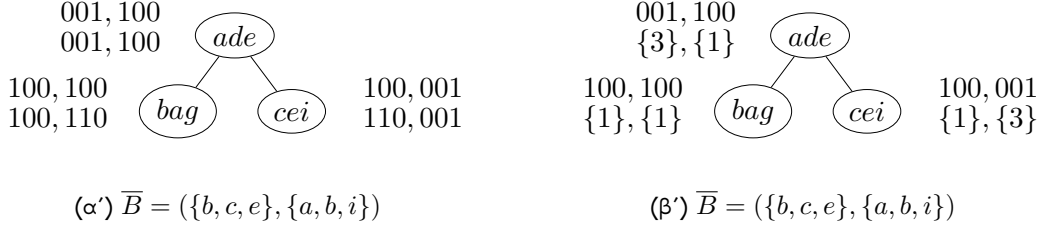
Αυτή είναι η κωδικοποίηση που χρησιμοποιούμε σε κάθε σημείο αυτού του κεφαλαίου με τη μόνη διαφοροποίηση ότι θεωρούμε πιο βολικό να υιοθετήσουμε την αναπαράσταση ενός διανύσματος  $\epsilon = (e^1, \dots, e^\ell)$  πάνω στο  $\{0, 1\}$  με τη μορφή ενός συνόλου

$$I_\epsilon = \{i \mid e^i = 1\} \subseteq \{1, \dots, \ell\}$$

που περιέχει τις θέσεις του  $\epsilon$  όπου εμφανίζεται το 1. Προτού παρουσιάσουμε τον τυπικό ορισμό, δίνουμε ένα παράδειγμα.

**Παράδειγμα 3.2.1.** Το Σχήμα 3.3 παρουσιάζει παραδείγματα των κωδικοποιήσεων που αναφέραμε παραπάνω. Πιο συγκεκριμένα το (α') δείχνει την κωδικοποίηση χρησιμοποιώντας τον αντιπροσωπευτικό κόμβο (πρώτη γραμμή των διανυσμάτων από 0,1) και χρησιμοποιώντας τις πολλαπλές εμφανίσεις ενός στοιχείου (δεύτερη γραμμή). Στο (β') δίνεται η κωδικοποίηση μέσω του αντιπροσωπευτικού κόμβου, αρχικά ως μία απεικόνιση στο  $\bigcup_{\ell=1}^{w+1} \{0, 1\}^{\ell \cdot k}$  και μετά ως μία απεικόνιση στο  $(\mathcal{P}(\{1, \dots, w + 1\}))^k$ .

Σχήμα 3.3: Αντιμετωπίζοντας τις πολλαπλές εμφανίσεις: μία ομοιόμορφη κωδικοποίηση.



**Ορισμός 3.2.1.** Ας υποθέσουμε ότι η  $\mathcal{I} = (\mathcal{T}, (\bar{a}_n)_{n \in T})$  είναι μία ειδική δεντροδιάσπαση μιας πεπερασμένης δομής με σύμπαν το  $A$  και δεντροπλάτος  $w \geq 1$ . Δεδομένης της  $\mathcal{I}$ , κάθε στοιχείο  $a \in A$  αντιστοιχεί μονοσήμαντα σε ένα ζευγάρι

$$\text{pair}_{\mathcal{I}}(a) = (n, j) \in T \times W, \quad W = \{1, \dots, w + 1\}$$

το οποίο ορίζει τη θέση μέγιστου ύψους ως προς την  $\mathcal{I}$  στην οποία εμφανίζεται το  $a$ , και ονομάζεται το (αντιπροσωπευτικό) ζευγάρι του  $a$  ως προς την  $\mathcal{I}$ . Πιο συγκεκριμένα, έχουμε

$$\text{pair}_{\mathcal{I}}(a) = (n, j) \quad \text{ανν} \quad a_n^j = a \ \& \ a_{n'}^i \neq a$$

όπου το  $n'$  είναι ο πατέρας του  $n$  στο  $\mathcal{T}$  και  $1 \leq i \leq |\bar{a}_{n'}|$ .

Ονομάζουμε (αντιπροσωπευτικό) κόμβο του  $a$  ως προς την  $\mathcal{I}$ , και τον συμβολίζουμε με  $\text{node}_{\mathcal{I}}(a)$ , την προβολή πάνω στην πρώτη θέση του  $\text{pair}_{\mathcal{I}}(a)$ , δηλαδή

$$\text{node}_{\mathcal{I}}(a) = n \quad \text{ανν} \quad \text{υπάρχει } j \in W \text{ τ.ω. } \text{pair}_{\mathcal{I}}(a) = (n, j).$$

Επίσης ορίζουμε το σύνολο  $\text{Pos}_{\mathcal{I}}(n)$  από θέσεις στοιχείων στον  $n$  ως προς την  $\mathcal{I}$  ως εξής

$$\text{Pos}_{\mathcal{I}}(n) = \{j \in W \mid \exists a \in A ((n, j) = \text{pair}_{\mathcal{I}}(a))\}$$

δηλαδή ως το σύνολο από τις θέσεις του διατεταγμένου σάκου  $\bar{a}_n$  που εμφανίζονται τα στοιχεία του  $A$  για τα οποία ο  $n$  είναι ο κόμβος με μέγιστο ύψος που τα περιέχει.

**Ορισμός 3.2.2.** Ας θεωρήσουμε ότι τα  $k, w$  είναι θετικοί ακέραιοι και ότι τα  $T, A$  είναι πεπερασμένα σύνολα. Συμβολίζουμε με  $\mathcal{E}_{k,w}(T)$  το σύνολο που περιέχει οποιαδήποτε ολική απεικόνιση  $\varepsilon : T \rightarrow (\mathcal{P}(W))^k$  όπου  $W = \{1, \dots, w + 1\}$ , και με  $\text{Dec}_w(T, A)$  το σύνολο των ειδικών δεντροδιασπάσεων, με πλάτος  $w$  και σύνολο κόμβων  $T$ , δομών με σύμπαν το  $A$ . Ορίζουμε την απεικόνιση κωδικοποίησης

$$\text{enc}_{k,w} : \bigcup_{A,T} ((\mathcal{P}(A))^k \times \text{Dec}_w(T, A)) \rightarrow \bigcup_T (\mathcal{E}_{k,w}(T))$$

ως εξής: για κάθε  $k$ -ανάθεση  $\bar{B} = (B_1, \dots, B_k) \in (\mathcal{P}(A))^k$  και  $\mathcal{I} = (\mathcal{T}, (\bar{a}_n)_{n \in T}) \in \text{Dec}_w(T, A)$ , έχουμε  $\text{enc}_{k,w}(\bar{B}, \mathcal{I}) = \varepsilon_{\bar{B}}$  όπου  $\varepsilon_{\bar{B}} : T \rightarrow (\mathcal{P}(W))^k$  τέτοια ώστε για κάθε  $n \in T$ ,

$$\varepsilon_{\overline{B}}(n) = (I_n^1, \dots, I_n^k) \quad \text{ανν} \quad I_n^i = \{j \in \text{Pos}_{\mathcal{I}}(n) \mid a_n^j \in B_i\} \subseteq W, \quad 1 \leq i \leq k$$

δηλαδή κάθε  $I_n^i$  είναι το σύνολο εκείνων των θέσεων στοιχείων στον  $n$  στις οποίες εμφανίζονται στοιχεία του  $B_i$ . Ονομάζουμε την  $\varepsilon_{\overline{B}}$  απεικόνιση που κωδικοποιεί την ανάθεση  $\overline{B}$  (ή απλά κωδικοποίηση της  $\overline{B}$ ) ως προς τη δεντροδιάσπαση  $\mathcal{I}$ .

Με ανάλογο τρόπο ορίζεται η απεικόνιση αποκωδικοποίησης

$$\text{dec}_{k,w} : \bigcup_{A,T} (\mathcal{E}_{k,w}(T) \times \text{Dec}_w(T, A)) \rightarrow \bigcup_A ((\mathcal{P}(A))^k)$$

ως εξής: για κάθε  $\varepsilon : T \rightarrow (\mathcal{P}(W))^k$  με  $\varepsilon(n) = (I_n^1, \dots, I_n^k)$ , και δεντροδιάσπαση  $\mathcal{I} = (\mathcal{T}, (\overline{a}_n)_{n \in T}) \in \text{Dec}_w(T, A)$ , έχουμε  $\text{dec}_{k,w}(\varepsilon, \mathcal{I}) = \overline{B}_\varepsilon$  όπου  $\overline{B}_\varepsilon = (B_1, \dots, B_k) \in (\mathcal{P}(A))^k$  τέτοιο ώστε

$$B_i = \bigcup_{n \in T} \{a_n^j \mid j \in I_n^i\}, \quad 1 \leq i \leq k.$$

Ας υποθέσουμε ότι  $\mathcal{I} = (\mathcal{T}, (\overline{a}_n)_{n \in T}) \in \text{Dec}_w(T, A)$ . Κάθε απεικόνιση  $\varepsilon : T \rightarrow (\mathcal{P}(W))^k$  τέτοια ώστε  $\varepsilon(n) \in (\mathcal{P}(\text{Pos}_{\mathcal{I}}(n)))^k$  για κάθε  $n \in T$  ονομάζεται απεικόνιση ανάθεσης ως προς το  $\mathcal{I}$ .

**Παρατήρηση 3.2.1.** Ας θεωρήσουμε ότι  $\mathcal{I} = (\mathcal{T}, (\overline{a}_n)_{n \in T}) \in \text{Dec}_w(T, A)$ , και ότι το  $\mathcal{E}_{k,w,\mathcal{I}}$  είναι το σύνολο που περιέχει κάθε απεικόνιση ανάθεσης ως προς την  $\mathcal{I}$ . Από τον ορισμό του, το  $\mathcal{E}_{k,w,\mathcal{I}}$  είναι η εικόνα του περιορισμού της απεικόνισης κωδικοποίησης  $\text{enc}_{k,w}$  στο  $(\mathcal{P}(A))^k \times \mathcal{I}$ . Ο περιορισμός της απεικόνισης αποκωδικοποίησης  $\text{dec}_{k,w}$  στο  $\mathcal{E}_{k,w,\mathcal{I}} \times \mathcal{I}$  αντιστοιχεί στην «αντίστροφη απεικόνιση» του περιορισμού της απεικόνισης κωδικοποίησης  $\text{enc}_{k,w}$  στο  $(\mathcal{P}(A))^k \times \mathcal{I}$  με την έννοια ότι για κάθε  $\varepsilon \in \mathcal{E}_{k,w,\mathcal{I}}$ , έχουμε

$$\text{dec}_{k,w}(\varepsilon, \mathcal{I}) = (B_1, \dots, B_k) \quad \text{ανν} \quad \text{enc}_{k,w}((B_1, \dots, B_k), \mathcal{I}) = \varepsilon.$$

Χρησιμοποιώντας την απεικόνιση κωδικοποίησης που ορίστηκε σε αυτή την ενότητα, θα ορίσουμε στη συνέχεια αυτόματα αναθέσεων που εκτελούνται πάνω σε επεκτεταμένες δεντροδιασπάσεις και παράγουν ως τμήμα των εκτελέσεών τους απεικονίσεις που κωδικοποιούν αναθέσεις για MSO-τύπους. Αυτά τα αυτόματα, που τα ονομάζουμε αυτόματα αποδόμησης, υπολογίζουν τις αναθέσεις που ικανοποιούν MSO-τύπους αποκωδικοποιώντας –μέσω της  $\text{dec}_{k,w}$ – κάθε απεικόνιση ανάθεσης που υπολογίζεται μαζί με τις επιτυχημένες εκτελέσεις τους.

### 3.2.2 Συνδυάζοντας δομές & δεντροδιασπάσεις

Στην ενότητα αυτή ορίζουμε τυπικά τα δέντρα που αποτελούν την είσοδο για τα αυτόματα μας. Όπως έχουμε ήδη αναφέρει, τα αυτόματα αποδόμησης είναι αυτόματα τα οποία εκτελούνται πάνω σε χρωματισμένα δέντρα που κωδικοποιούν δεντροδιασπάσεις και δομές.

Ας υποθέσουμε ότι η  $\mathcal{A}$  είναι μία  $\tau$ -δομή για  $\tau = \{R_1, \dots, R_\ell\}$ , και ότι η  $\mathcal{I} = (\mathcal{T}, (\overline{a}_n)_{n \in T})$  είναι μία διατεταγμένη δεντροδιάσπαση του  $\mathcal{A}$  πλάτους  $w$ . Ένας τετριμμένος τρόπος να

περιγράψουμε το ζευγάρι  $(\mathcal{A}, \mathcal{I})$  με τη μορφή χρωματισμένου δέντρου είναι επεκτείνοντας τους σάκους της  $\mathcal{I}$  με τέτοιο τρόπο ώστε να περιλαμβάνει τις σχέσεις της δομής  $\mathcal{A}$ . Ένα παράδειγμα μιας τέτοιας επέκτασης είναι το χρωματισμένο δέντρο

$$\mathcal{I}; \mathcal{A} = (\mathcal{T}, c_A)$$

με συνάρτηση χρωματισμού  $c_A$  που περιγράφει σε κάθε κόμβο  $n$  i) το αντίστοιχο κομμάτι της  $\mathcal{I}$  και ii) την υποδομή  $\mathcal{A}|_{a_n}$  της  $\mathcal{A}$  που ορίζεται από τον  $a_n$ . Πιο συγκεκριμένα, ας υποθέσουμε ότι

$$\Gamma_{\tau, w, A} = \left( \bigcup_{j=1}^{w+1} A^j \right)^2 \times \mathcal{P}(A^{r_1}) \times \dots \times \mathcal{P}(A^{r_\ell})$$

και ότι το  $\mathcal{I}; \mathcal{A}$  είναι ένα  $\Gamma_{\tau, w, A}$ -δέντρο, τέτοιο ώστε η  $c_A : T \rightarrow \Gamma_{\tau, w, A}$  να αντιστοιχεί σε κάθε κόμβο  $n$  με πατέρα  $n'$  την  $(r_\ell + 2)$ -πλειάδα

$$(\bar{a}_n, \bar{a}_{n'}, R_1^A|_{a_n}, \dots, R_\ell^A|_{a_n})$$

όπου τα  $\bar{a}_n$  και  $\bar{a}_{n'}$  είναι οι διατεταγμένοι σάκοι της δεντροδιάσπασης  $\mathcal{I}$  στους κόμβους  $n$  και  $n'$  αντίστοιχα, και κάθε  $R_i^A|_{a_n} = R_i^A \cap (a_n)^{r_i}$  είναι ο περιορισμός της σχέσης  $R_i^A$  στα στοιχεία του σάκου  $a_n$ ,  $1 \leq i \leq \ell$ . Το χρωματισμένο δέντρο  $\mathcal{I}; \mathcal{A}$ , παρόλο που κωδικοποιεί τα  $\mathcal{A}$  και  $\mathcal{I}$  επιτυχημένα, έχει το εξής μειονέκτημα: ο ορισμός του έχει εξάρτηση από το σύμπαν  $A$  της  $\mathcal{A}$ . Συνεπώς, τα αυτόματα τα οποία θα εκτελούνται πάνω σε τέτοια δέντρα θα είναι αυτόματα που εκτελούνται πάνω σε μία ιδιαίτερα περιορισμένη κλάση δέντρων, δηλαδή σε εκείνα τα δέντρα που κωδικοποιούν δομές με σύμπαν το  $A$ . Κατά συνέπεια, μέσω αυτής της κωδικοποίησης του ζευγαριού  $(\mathcal{A}, \mathcal{I})$  δε μπορούμε να έχουμε μία γενική λύση που να βασίζεται σε ένα αυτόματο.

Θα δούμε τώρα ότι μπορούμε να ορίσουμε ένα χρωματισμένο δέντρο, που κωδικοποιεί τα  $\mathcal{I}$  και  $\mathcal{A}$ , και ταυτόχρονα δεν έχει καμία εξάρτηση από το σύμπαν  $A$  της  $\mathcal{A}$ . Αυτό το νέο δέντρο, που συμβολίζεται με  $\mathcal{T}_{\mathcal{I}; \mathcal{A}}$ , ουσιαστικά κωδικοποιεί τον τύπο ισομορφισμού του  $\mathcal{A}; \mathcal{I}$ , δηλαδή την κλάση όλων των δέντρων που είναι ισόμορφα με το  $\mathcal{A}; \mathcal{I}$ . Όπως και παραπάνω, το υποκείμενο δέντρο του  $\mathcal{T}_{\mathcal{I}; \mathcal{A}}$  είναι το δέντρο  $\mathcal{T}$  της δεντροδιάσπασης  $\mathcal{I}$ . Η νέα συνάρτηση χρωματισμού  $c$  κωδικοποιεί σε κάθε κόμβο  $n$  i) τις ιδιότητες που περιγράφουν τη δομή της  $\mathcal{I}$ , δηλαδή το μέγεθος του σάκου  $a_n$  και την τομή του  $a_n$  με τον σάκο  $a_{n'}$  του πατέρα  $n'$  του  $n$  (αντιστοιχούν στις δύο πρώτες θέσεις του  $c(n)$  παρακάτω), και ii) τον τύπο ισομορφισμού της υποδομής  $\mathcal{A}|_{a_n}$  του  $\mathcal{A}$  που ορίζεται από τον  $a_n$  (αντιστοιχούν στις τελευταίες  $\ell$  θέσεις του  $c(n)$  παρακάτω). Πιο συγκεκριμένα, ας υποθέσουμε ότι

$$\Gamma_{\tau, w} = W \times \mathcal{P}(W^2) \times \mathcal{P}(W^{r_1}) \times \dots \times \mathcal{P}(W^{r_\ell}), \quad W = \{1, \dots, w+1\}.$$

Ορίζουμε το  $\Gamma_{\tau, w}$ -δέντρο  $\mathcal{T}_{\mathcal{I}; \mathcal{A}}$  και το ονομάζουμε το δέντρο της  $\mathcal{A}$  ως προς την  $\mathcal{I}$ , ως εξής

$$\mathcal{T}_{\mathcal{I}; \mathcal{A}} = (\mathcal{T}, c)$$

όπου  $c : T \rightarrow \Gamma_{\tau, w}$  είναι η συνάρτηση χρωματισμού με τιμή σε κάθε  $n \in T$  ίση με  $c(n) = (s, P, S_1, \dots, S_\ell)$  τέτοια ώστε

- $s = |a_n|$ ,
- $P = \begin{cases} \emptyset, & \text{αν το } n \text{ είναι η ρίζα} \\ \{(i, j) \mid a_n^i = a_{n'}^j\}, & \text{αν το } n' \text{ είναι ο πατέρας του } n. \end{cases}$
- $S_i = \{(j_1, \dots, j_{r_i}) \mid (a_n^{j_1}, \dots, a_n^{j_{r_i}}) \in R_i^A\}, \quad 1 \leq i \leq \ell.$

Τα  $\Gamma_{\tau,w}$ -δέντρα προσφέρουν έναν ομοιόμορφο τρόπο για να αναφερόμαστε στα δέντρα που κωδικοποιούν ζευγάρια  $(\mathcal{A}, \mathcal{I})$  όπου η  $\mathcal{A}$  είναι κάποια  $\tau$ -δομή και η  $\mathcal{I}$  είναι μία δεντροδιάσπαση της  $\mathcal{A}$  πλάτους το πολύ  $w$ . Είναι εμφανές ότι κάθε  $\Gamma_{\tau,w}$ -δέντρο δεν κωδικοποιεί απαραίτητα ένα τέτοιο ζευγάρι. Θα ονομάζουμε  $(\tau, w)$ -δέντρα τα στοιχεία της κλάσης  $\mathcal{T}_{\tau,w}$  των  $\Gamma_{\tau,w}$ -δέντρων που κωδικοποιούν τέτοια ζευγάρια. Τα αυτόματα αποδόμησης εκτελούνται πάνω σε  $(\tau, w)$ -δέντρα.

**Παράδειγμα 3.2.2.** Ας θεωρήσουμε ως δομή εισόδου και δεντροδιάσπαση, τον γράφο  $\mathcal{G}$  και την αντίστοιχη δεντροδιάσπαση  $\mathcal{I}$  του Σχήματος 3.1. Οι ορισμοί των συναρτήσεων χρωματισμού  $c_A$ ,  $A = \{a, b, c, d, e, f, g, h\}$  και  $c$  των δέντρων  $\mathcal{I}; \mathcal{A}$  και  $\mathcal{T}_{\mathcal{I}; \mathcal{A}}$  αντίστοιχα δίνονται στον Πίνακα 3.1.

$n$	$c_A(n)$	$c(n)$
$n_1$	$((a, g, f), \emptyset, \{(a, g), (g, f), (g, a), (f, g)\})$	$(3, \emptyset, \{(1, 2), (2, 3), (2, 1), (3, 2)\})$
$n_2$	$((a, f, c), (a, g, f), \{(a, c), (f, c), (c, a), (c, f)\})$	$(3, \{(1, 1), (2, 3)\}, \{(1, 3), (2, 3), (3, 1), (3, 2)\})$
$n_3$	$((g, h), (a, g, f), \{(g, h), (h, g)\})$	$(2, \{(1, 2)\}, \{(1, 2), (2, 1)\})$
$n_4$	$((a, b, c), (a, f, c), \{(a, b), (a, c), (b, c), (b, a), (c, a), (c, b)\})$	$(3, \{(1, 1), (3, 3)\}, \{(1, 2), (1, 3), (2, 3), (2, 1), (3, 1), (3, 2)\})$
$n_5$	$((c, d, e), (a, f, c), \{(c, d), (c, e), (d, e), (d, c), (e, c), (e, d)\})$	$(3, \{(1, 3)\}, \{(1, 2), (1, 3), (2, 3), (2, 1), (3, 1), (3, 2)\})$

Πίνακας 3.1: Οι δύο διαφορετικές αναπαραστάσεις του ζευγαριού  $(\mathcal{A}, \mathcal{I})$ .

**Παρατήρηση 3.2.2.** Ο παραπάνω ορισμός του  $\mathcal{T}_{\mathcal{I}; \mathcal{A}}$  παρουσιάζει βασικές ομοιότητες με τον ορισμό του δέντρου  $\mathcal{T}^*$  του [FFG02], με τη διαφορά ότι το  $\mathcal{T}^*$  ορίζεται ως προς διατεταγμένες δεντροδιασπάσεις των οποίων οι σάκοι  $\bar{a}_n$  έχουν σταθερό μήκος ίσο με  $(w + 1)$ . Καθώς το σταθερό μήκος των σάκων επιτυγχάνεται με πολλαπλές εμφανίσεις στοιχείων του  $a_n$  στο  $\bar{a}_n$ , μία διαφορετική συντεταγμένη, που δηλώνει τις θέσεις του  $\bar{a}_n$  που εμφανίζονται ίδια στοιχεία, αντικαθιστά την πρώτη συντεταγμένη του χρώματος  $c(n)$  του  $\mathcal{T}_{\mathcal{I}; \mathcal{A}}$ . Πιο συγκεκριμένα, το  $\mathcal{T}^*$  είναι ένα δυαδικό  $\Gamma_{\tau,w}^*$ -δέντρο, όπου

$$\Gamma_{\tau,w}^* = \mathcal{P}(W^2) \times \mathcal{P}(W^2) \times \mathcal{P}(W^{r_1}) \times \dots \times \mathcal{P}(W^{r_\ell})$$

με συνάρτηση χρωματισμού  $c^* : T \mapsto \Gamma_{\tau,w}^*$  που αναθέτει σε κάθε κόμβο  $n \in T$  το χρώμα  $\gamma = (I, P, S_1, \dots, S_\ell)$  όπου  $I = \{(i, j) \mid a_n^i = a_n^j\}$  και τα  $P, S_1, \dots, S_\ell$  ορίζονται ακριβώς όπως στο  $c(n)$ . Παρόλο που ο ορισμός της συνάρτησης  $c^*$  είναι σχεδόν ίδιος με αυτόν της  $c$ , το  $\mathcal{T}_{\mathcal{I},\mathcal{A}}$  συνιστά απλοποίηση του  $\mathcal{T}^*$  ως προς μία τυχαία δεντροδιάσπαση  $\mathcal{I}$  με υποκείμενο δέντρο ένα δυαδικό δέντρο, με την έννοια ότι i) δεν χρειάζεται να γεμίζουμε τους σάκους με «άχρηστες» επαναλήψεις στοιχείων προκειμένου να έχουμε το ίδιο μέγεθος σε κάθε διατεταγμένο σάκο  $\bar{a}_n$  και επίσης να κωδικοποιούμε αυτές τις επαναλήψεις στη συνάρτηση χρωματισμού, και ii) δεν έχουμε κατά συνέπεια πολλαπλές αναπαραστάσεις της ίδιας πληροφορίας στα σύνολα  $P$  και  $S_i$  του  $\gamma$ . Για παράδειγμα, όταν  $a_n = (a, b, a, b)$  και  $(a, b) \in R_i^A$  θα έχουμε  $(1, 2), (1, 4), (3, 2), (3, 4) \in S_i$  στο αντίστοιχο χρώμα του  $\mathcal{T}^*$ . Δηλαδή έχουμε μία σημαντική άσκοπη αύξηση του μεγέθους των αντίστοιχων χρωμάτων, η οποία συνεπάγεται τελικά πιο σύνθετους ορισμούς για τα αντίστοιχα αυτόματα.

### 3.2.3 Μία άμεση αναγωγή του προβλήματος αποτίμησης της MSO σε πρόβλημα αυτομάτων: τα αυτόματα αποδόμησης

Στην ενότητα αυτή ορίζουμε τα αυτόματα αποδόμησης για MSO[ $\tau$ ] τύπους. Τα αυτόματα αυτά, που γενικεύουν τα αυτόματα αναθέσεων που ορίστηκαν στον προηγούμενο κεφάλαιο, έχουν σημαντικά πιο σύνθετους ορισμούς τόσο για τις σχέσεις μετάβασης όσο και για τον μηχανισμό υπολογισμού τους. Το γεγονός αυτό οφείλεται στο ότι οι ορισμοί αυτοί δεν εξαρτώνται πλέον μόνο από την αρχική δομή, αλλά και από την αντίστοιχη δεντροδιάσπαση που αποτελεί μέρος της εισόδου του προβλήματος.

**Τα αυτόματα  $w$ -αποδόμησης για MSO τύπους:  $w$ -dec-Assign $_\phi$ .** Θυμίζουμε ότι το Assign $_\phi$  που ορίστηκε στο προηγούμενο κεφάλαιο είναι ένα  $\Gamma$ -αυτόματο  $k$ -αναθέσεων. Με βάση τον Ορισμό 2.2.2, ένα τέτοιο αυτόματο είναι ένα αυτόματο για δέντρα, με σύνολο καταστάσεων  $Q \times E_k$  όπου  $E_k = \{0, 1\}^k$  και απεικόνιση αποκωδικοποίησης την  $\text{dec}_k$ , που υπολογίζει  $k$ -αναθέσεις. Χρησιμοποιώντας τον ίδιο φορμαλισμό, το  $w$ -dec-Assign $_\phi$  μπορεί να οριστεί ως ένα  $\Gamma_{\tau,w}$ -αυτόματο  $(k, w)$ -αναθέσεων, με σύνολο καταστάσεων  $Q \times E_{k,w}$  όπου  $E_{k,w} = (\mathcal{P}(\{1, \dots, w+1\}))^k$  και απεικόνιση αποκωδικοποίησης την  $\text{dec}_{w,k}$ .

**Ορισμός 3.2.3.** Ας υποθέσουμε ότι ο  $\phi$  είναι ένας MSO[ $\tau$ ] τύπος με  $k$  ελεύθερες μεταβλητές. Ένα αυτόματο (αναθέσεων)  $w$ -αποδόμησης για τον  $\phi$ , που συμβολίζεται με  $w$ -dec-Assign $_\phi$ , ορίζεται από το ζευγάρι

$$(\mathcal{A}_{\phi,w}, \text{dec}_{k,w})$$

όπου το

$$\mathcal{A}_{\phi,w} = (\Gamma_{\tau,w}, Q_{k,w}, w\text{-}\Delta_0, w\text{-}\Delta, F_{k,w})$$

είναι ένα είδος μη-ντετερμινιστικού bottom-up αυτομάτου για δέντρα: το  $Q_{k,w} = Q \times (\mathcal{P}(W))^k$ ,  $W = \{1, \dots, w+1\}$ , είναι το σύνολο καταστάσεων, και το  $F_{k,w} = F \times (\mathcal{P}(W))^k$  είναι το σύνολο των τελικών καταστάσεων, όπου το  $Q$  είναι ένα πεπερασμένο σύνολο και ισχύει  $F \subseteq Q$ . Δηλαδή κάθε κατάστασή του είναι μία  $(k+1)$ -πλειάδα  $(q, I_1, \dots, I_k)$  που συχνά συμβολίζουμε

με  $q\bar{I}$ . Η  $w\text{-}\Delta_0 \subseteq \Gamma_{\tau,w} \times Q_{k,w}$  είναι η αρχική σχέση μετάβασης και η  $w\text{-}\Delta \subseteq Q \times Q \times \Gamma_{\tau,w} \times Q_{k,w}$  είναι η σχέση μετάβασης. Η  $\text{dec}_{k,w}$  είναι η απεικόνιση αποκωδικοποίησης

$$\text{dec}_{k,w} : \bigcup_{A,T} (\mathcal{E}_{k,w}(T) \times \text{Dec}_w(T, A)) \rightarrow \bigcup_A ((\mathcal{P}(A))^k)$$

που δίνεται στον Ορισμό 3.2.2 και περιγράφει τον μηχανισμό υπολογισμού του  $w\text{-dec-Assign}_\phi$ , που συνίσταται στην αποκωδικοποίηση των εκτελέσεων σε αναθέσεις με τη βοήθεια μιας δεδομένης δειτροδιάσπασης.

Η εκτέλεση  $\rho$  του  $w\text{-dec-Assign}_\phi$  πάνω στο  $\Gamma_{\tau,w}$ -χρωματισμένο δέντρο  $\mathcal{T}_{\mathcal{I};A}$  είναι μία απεικόνιση που αναθέτει καταστάσεις στους κόμβους του  $\mathcal{T}_{\mathcal{I};A}$  έτσι ώστε

- αν το  $n$  είναι ένα φύλλο χρώματος  $\gamma$ , τότε έχουμε  $\rho(n) = q\bar{I}$  αν υπάρχει μία μετάβαση  $(\gamma, q\bar{I}) \in w\text{-}\Delta_0$
- αν το  $n$  είναι ένας κόμβος χρώματος  $\gamma$ , με παιδιά τα  $n_1, n_2$ , τότε η τιμή  $\rho(n)$  είναι τέτοια ώστε  $\rho(n_1) = q\bar{J}$ ,  $\rho(n_2) = p\bar{L}$  και  $(q, p, \gamma, \rho(n)) \in w\text{-}\Delta$ .

Μία εκτέλεση είναι *επιτυχημένη* αν αντιστοιχεί τη ρίζα σε μία τελική κατάσταση.

Είναι πολλές φορές χρήσιμο να θεωρούμε μία εκτέλεση  $\rho : T \rightarrow Q \times (\mathcal{P}(W))^k$  του  $w\text{-dec-Assign}_\phi$  ως τη συνένωση

$$\varsigma; \varepsilon$$

όπου  $\varsigma : T \rightarrow Q$  και  $\varepsilon : T \rightarrow (\mathcal{P}(W))^k$  τ.ω.  $\rho(n) = (q, \bar{I})$  αν  $\varsigma(n) = q$  και  $\varepsilon(n) = \bar{I}$ . Ονομάζουμε τα  $\varsigma, \varepsilon$  το μέρος των καταστάσεων και το μέρος της ανάθεσης της εκτέλεσης  $\rho$  αντίστοιχα.

Λέμε ότι το  $w\text{-dec-Assign}_\phi$  υπολογίζει την ανάθεση  $\bar{B} = (B_1, \dots, B_k) \in (\mathcal{P}(A))^k$  αν υπάρχει μία *επιτυχημένη εκτέλεση*  $\varsigma; \varepsilon$  του  $w\text{-dec-Assign}_\phi$  πάνω στο  $\mathcal{T}_{\mathcal{I};A}$  τέτοια ώστε

$$\text{dec}_{k,w}(\varepsilon, \mathcal{I}) = (B_1, \dots, B_k).$$

Το σύνολο των αναθέσεων που υπολογίζει το  $w\text{-dec-Assign}_\phi$  πάνω στο  $\mathcal{T}_{\mathcal{I};A}$  συμβολίζεται με  $w\text{-dec-Assign}_\phi(\mathcal{T}_{\mathcal{I};A})$ .

Κάθε απεικόνιση  $\varepsilon$  που αποτελεί το μέρος της ανάθεσης μιας επιτυχημένης εκτέλεσης του  $w\text{-dec-Assign}_\phi$  είναι μία απεικόνιση ανάθεσης (βλέπε τον Ορισμό 3.2.2). Μάλιστα, υπάρχει μία ένα-προς-ένα αντιστοιχία ανάμεσα στις αναθέσεις που υπολογίζει το  $w\text{-dec-Assign}_\phi$  και στις επιτυχημένες εκτελέσεις του.

Σημειώνουμε ότι είναι για λόγους απλοποίησης που ο παραπάνω ορισμός των αυτομάτων αποδόμησης δίνεται ως προς MSO-τύπους με  $k$  δευτεροβάθμιες μεταβλητές, δηλαδή τύπους  $\phi(X_1, \dots, X_k)$ . Ο ορισμός προσαρμόζεται εύκολα στη γενική περίπτωση, δηλαδή για τύπους  $\phi(x_1, \dots, x_u, X_1, \dots, X_v)$ ,  $k = u + v$ , επεκτείνοντας φυσικά τα παραπάνω. Πιο συγκεκριμένα,

αρκεί να αντικαταστήσουμε την απεικόνιση αποκωδικοποίησης  $\text{dec}_{k,w}$  με μία νέα απεικόνιση αποκωδικοποίησης  $\text{dec}_{u,v,w}$  που ορίζεται ως εξής:

$$\text{dec}_{u,v,w}(\varepsilon, \mathcal{I}) = (a_1, \dots, a_u, B_1, \dots, B_v) \text{ ανν } \text{dec}_{k,w}(\varepsilon, \mathcal{I}) = (\{a_1\}, \dots, \{a_u\}, B_1, \dots, B_v)$$

για κάθε  $\varepsilon$  που αποτελεί μέρος ανάθεσης κάποιας επιτυχημένης εκτέλεσης του  $w\text{-dec-Assign}_\phi$  πάνω στο  $\mathcal{T}_{\mathcal{I};\mathcal{A}}$ .

Στη συνέχεια ορίζουμε τα αυτόματα αποδόμησης με μέτρηση τα οποία χρησιμοποιούμε για να επιλύσουμε τα MSO-ορίσιμα προβλήματα βελτιστοποίησης.

**Ορισμός 3.2.4.** Ας υποθέσουμε ότι ο  $\phi(X)$  είναι ένας MSO[ $\tau$ ] τύπος. Ένα αυτόματο  $w$ -αποδόμησης με μέτρηση για τον  $\phi$ , που συμβολίζεται με  $w\text{-dec-Assign}_\phi^{\text{count}}$ , ορίζεται μέσω του  $w\text{-dec-Assign}_\phi$  με μοναδική διαφορά τον ορισμό της εκτέλεσής του. Πιο συγκεκριμένα, η εκτέλεση  $\rho$  του  $w\text{-dec-Assign}_\phi^{\text{count}}$  πάνω στο  $\mathcal{T}_{\mathcal{I};\mathcal{A}}$  είναι μία απεικόνιση

$$T \rightarrow Q \times \mathcal{P}(W) \times \{1, \dots, |A|\}$$

τέτοια ώστε

- αν ο  $n$  είναι φύλλο χρώματος  $\gamma$ , τότε  $\rho(n) = (qI, |I|)$  αν υπάρχει μία μετάβαση  $(\gamma, qI) \in \Delta_0$ ,
- αν ο  $n$  είναι ένας κόμβος χρώματος  $\gamma$ , με παιδιά  $n_1, n_2$  και  $\rho(n_1) = (r, J, j)$ ,  $\rho(n_2) = (p, L, \ell)$  και  $(r, p, \gamma, qI) \in \Delta$ , τότε  $\rho(n) = (qI, j + \ell + |I|)$ .

**Κεντρικό θεώρημα: η σχέση MSO–αυτομάτων μέσω των αυτομάτων αποδόμησης.** Το Θεώρημα 3.2.1, που παρουσιάζεται παρακάτω, δηλώνει ότι για κάθε MSO[ $\tau$ ]-τύπο  $\phi$  υπάρχει ένα αυτόματο αποδόμησης  $w\text{-dec-Assign}_\phi$  που εκτελείται πάνω σε δέντρα  $\tau$ -δομών και υπολογίζει τις αναθέσεις που ικανοποιούν τον  $\phi$  πάνω στις  $\tau$ -δομές. Δηλαδή αποτελεί γενίκευση του αντίστοιχου θεωρήματος του προηγούμενου κεφαλαίου για την περίπτωση των δέντρων.

Η απόδειξη που δίνουμε είναι κατασκευαστική και πραγματοποιείται με επαγωγή στη δομή του MSO τύπου  $\phi$ . Στην πραγματικότητα, αυτό που αποδεικνύουμε είναι η ύπαρξη ενός ντετερμινιστικού bottom-up αυτομάτου δέντρων  $w\text{-dec-A}_\phi$  που αποδέχεται  $(\tau, w)$ -δέντρα που κωδικοποιούν, μέσω της  $\text{enc}_{k,w}$ , τις αναθέσεις που ικανοποιούν τον τύπο  $\phi$ . Η σχέση ανάμεσα στα  $w\text{-dec-A}_\phi$  και  $w\text{-dec-Assign}_\phi$  είναι η εξής: η  $\varsigma; \varepsilon_{\overline{B}}$  είναι μία εκτέλεση του  $w\text{-dec-Assign}_\phi$  πάνω στο  $\mathcal{T}_{\mathcal{I};\mathcal{A}} = (\mathcal{T}, c)$  ανν η  $\varsigma$  είναι μία εκτέλεση του  $w\text{-dec-A}_\phi$  πάνω στο  $(\mathcal{T}_{\mathcal{I};\mathcal{A}}; \overline{B}) = (\mathcal{T}, c; \varepsilon_{\overline{B}})$ . Αναλυτικότερα, ας θεωρήσουμε ότι

$$w\text{-dec-A}_\phi = (\Gamma_{\tau,w} \times (\mathcal{P}(W))^k, Q, w\text{-}\delta_0, w\text{-}\delta, F).$$

Οι σχέσεις μετάβασης  $w\text{-}\Delta_0, w\text{-}\Delta$  του αυτομάτου

$$w\text{-dec-Assign}_\phi = (\Gamma_{\tau,w}, Q \times (\mathcal{P}(W))^k, w\text{-}\Delta_0, w\text{-}\Delta, F \times (\mathcal{P}(W))^k)$$

προκύπτουν εύκολα μέσω των συναρτήσεων μετάβασης  $w\text{-}\delta_0, w\text{-}\delta$  του  $w\text{-dec-A}_\phi$  ως εξής:



(i)  $(\gamma, (q, \bar{I})) \in w\text{-}\Delta_0$  ανν  $w\text{-}\delta_0((\gamma, \bar{I})) = q$

(ii) για κάθε  $\bar{I}_1, \bar{I}_2 \in (\mathcal{P}(W))^k$ ,

$$((q_1, \bar{I}_1), (q_2, \bar{I}_2), \gamma, (q, \bar{I})) \in w\text{-}\Delta \text{ ανν } w\text{-}\delta(q_1, q_2, (\gamma, \bar{I})) = q.$$

Έτσι, συχνά χρησιμοποιούμε το  $w\text{-dec-}\mathcal{A}_\phi$  ως συμπαγή αναπαράσταση του  $w\text{-dec-Assign}_\phi$ . Άλλωστε, οι εκτελέσεις του  $w\text{-dec-Assign}_\phi$  μπορούν να υπολογιστούν άμεσα από τις συναρτήσεις μετάβασης  $w\text{-}\delta_0, w\text{-}\delta$  του  $w\text{-dec-}\mathcal{A}_\phi$ , και αυτός είναι ένας επιπλέον λόγος για αυτή την επιλογή αναπαράστασης του  $w\text{-dec-Assign}_\phi$ . Επισημαίνουμε ότι, λόγω των κωδικοποιήσεων μέσω της  $\text{enc}_{k,w}$ , κάθε ανάθεση που ικανοποιεί τον τύπο παράγεται από μία μοναδική εκτέλεση του  $w\text{-dec-Assign}_\phi$ .

Προτού παρουσιάσουμε το κεντρικό θεώρημα, δίνουμε ένα λήμμα που χρησιμοποιείται στην απόδειξή του. Θυμίζουμε ότι σε ένα δέντρο  $\mathcal{T}$ , ένας κόμβος  $n$  βρίσκεται σε κατώτερο επίπεδο από κάποιον άλλο κόμβο  $n'$ , και το συμβολίζουμε με  $n <_{\mathcal{T}} n'$ , όταν η απόσταση του  $n$  από τη ρίζα –δηλαδή το μήκος του μονοπατιού που τα ενώνει– είναι μεγαλύτερη από την απόσταση του  $n'$  από τη ρίζα. Επιπλέον, θεωρούμε ότι

$$\bar{A}_n = \{a \in A \mid \text{node}_{\mathcal{T}}(a) \in T \setminus T_n\}.$$

**Ορισμός 3.2.5.** Ας υποθέσουμε ότι η  $\mathcal{A}$  είναι μία  $\tau$ -δομή με σύμπαν το  $A$  και ότι το  $R \in \tau$  έχει βαθμό 2. Θα λέμε ότι το  $b \in A$  είναι ένα  $R^A$ -ταίρι του  $a$  αν είτε  $(a, b) \in R^A$  ή  $(b, a) \in R^A$ .

**Λήμμα 3.2.1.** Ας υποθέσουμε ότι η  $\mathcal{A}$  είναι μία  $\tau$ -δομή με σύμπαν το  $A$ , ότι το  $R \in \tau$  έχει βαθμό 2, και ότι η  $\mathcal{I}$  είναι μία δεντροδιάσπαση της  $\mathcal{A}$ . Για κάθε  $a \in A$  τέτοιο ώστε  $\text{node}_{\mathcal{I}}(a) = n$ , ισχύει το εξής: κάθε  $R^A$ -ταίρι του  $a$  που περιέχεται στο  $\bar{A}_n$  περιέχεται επίσης στο σάκο  $a_n$  του  $n$ .

**Απόδειξη.** Το αποτέλεσμα αυτό αποτελεί άμεση συνέπεια των (i) και (ii) του ορισμού της δεντροδιάσπασης. Ας υποθέσουμε ότι  $(b, c) \in R^A$  και ότι  $\text{node}_{\mathcal{I}}(b) = n$  και  $\text{node}_{\mathcal{I}}(c) = n'$ . Αρκεί να αποδείξουμε ότι αν  $n <_{\mathcal{T}} n'$ , τότε  $\{b, c\} \in a_n$ .

Πράγματι, αφού  $(b, c) \in R^A$ , από το (ii) του ορισμού της δεντροδιάσπασης έχουμε ότι υπάρχει ένας κόμβος  $v$  τέτοιος ώστε  $\{b, c\} \in a_v$ . Όμως  $\text{node}_{\mathcal{I}}(b) = n$ , δηλαδή το  $b$  δεν εμφανίζεται σε κάποιον σάκο κόμβου που βρίσκεται σε ύψος μεγαλύτερο από αυτό του  $n$ , συνεπώς είτε  $v = n$  ή  $v <_{\mathcal{T}} n$ . Ας υποθέσουμε ότι  $v <_{\mathcal{T}} n$ . Επειδή  $c \in a_v$ , και  $c \in a_{n'}$ , και  $n <_{\mathcal{T}} n'$ , από το (i) του ορισμού της δεντροδιάσπασης θα έχουμε επίσης ότι  $c \in a_n$ .  $\square$

**Θεώρημα 3.2.1.** Για κάθε MSO[ $\tau$ ]-τύπο  $\phi$  με  $k$  ελεύθερες μεταβλητές και για κάθε ακέραιο  $w \geq 1$ , υπάρχει ένα αυτόματο αποδόμησης  $w\text{-dec-Assign}_\phi$  τέτοιο ώστε για κάθε  $\Gamma_{\tau,w}$ -χρωματισμένο δέντρο  $\mathcal{T}_{\mathcal{I},\mathcal{A}}$ , που κωδικοποιεί ένα ζευγάρι  $(\mathcal{A}, \mathcal{I})$ , να υπολογίζει τις αναθέσεις που ικανοποιούν τον  $\phi$  στην  $\mathcal{A}$ , δηλαδή τέτοιο ώστε

$$\phi(\mathcal{A}) = w\text{-dec-Assign}_\phi(\mathcal{T}_{\mathcal{I},\mathcal{A}}).$$

**Απόδειξη.** Η απόδειξη πραγματοποιείται με επαγωγή στη δομή του MSO τύπου  $\phi$ . Όπως έχουμε ήδη αναφέρει, αντί να κατασκευάσουμε το ίδιο το αυτόματο αποδόμησης  $w$ -dec-Assign $_{\phi}$  κατασκευάζουμε το ντετερμινιστικό αυτόματο  $w$ -dec-A $_{\phi}$ . Θυμίζουμε ότι το τελευταίο αποτελεί μία συμπαγή αναπαράσταση του πρώτου. Ένας ακόμη λόγος για αυτή την επιλογή στην απόδειξη αυτή, είναι το γεγονός ότι οι ορισμοί που απαιτούνται και οι αποδείξεις των επιμέρους προτάσεων απλοποιούνται σημαντικά λόγω του ότι έχουμε ένα ντετερμινιστικό αυτόματο.

Το ντετερμινιστικό αυτόματο  $w$ -dec-A $_{\phi}$  που κατασκευάζουμε, αποδέχεται την κλάση των  $(\Gamma_{\tau,w} \times (\mathcal{P}(W))^k)$ -δέντρων που κωδικοποιούν τις αναθέσεις που ικανοποιούν τον  $\phi$  πάνω σε  $\tau$ -δομές με δεντροπλάτος το πολύ  $w$ . Στη συνέχεια της απόδειξής μας, θα αναφερόμαστε σε αυτά τα δέντρα και ως τα δέντρα που επεκτείνονται με αναθέσεις ή με απεικονίσεις αναθέσεων. Η ονομασία αυτή δεν είναι τυχαία, καθώς πράγματι τα δέντρα αυτά προκύπτουν από  $\Gamma_{\tau,w}$ -δέντρα επεκτείνοντας τις αντίστοιχες συναρτήσεις χρωματισμού με απεικονίσεις αναθέσεων που κωδικοποιούν τις αναθέσεις που ικανοποιούν τον  $\phi$ . Σημειώνουμε ότι οι μεταβάσεις του  $w$ -dec-A $_{\phi}$  έχουν τη μορφή  $q_1, q_2, (\gamma, \bar{I}) \mapsto q$ .

Θεωρούμε αρχικά την περίπτωση των ατομικών τύπων: τα αυτόματα για τους  $\text{In}(x, X)$  και  $R(x, y)$  (δηλαδή για  $R_i = R$  με  $r_i = 2$ ) δίνονται στον Πίνακα 3.4 και στον Πίνακα 3.5 αντίστοιχα. Οι συνθήκες που χρησιμοποιούνται κατά τους ορισμούς τους δίνονται στον Πίνακα 3.2 ενώ η ερμηνεία των καταστάσεων για την περίπτωση του αυτομάτου για τον τύπο  $R(x, y)$  δίνεται στον Πίνακα 3.3.

Σημειώνουμε ότι οι πίνακες που παραθέτουν τις μεταβάσεις, δηλαδή ο Πίνακας 3.4 και ο Πίνακας 3.5, δίνουν μία συνοπτική αναπαράσταση των αντίστοιχων μεταβάσεων. Πιο συγκεκριμένα, κάθε γραμμή, διαφορετική από την τελευταία, που περιέχει μία μετάβαση της μορφής  $q_1, q_2, (\gamma, \bar{I}) \mapsto q$  με συνθήκη  $c(\bar{I})$  αντιστοιχεί στα εξής δυο σύνολα μεταβάσεων: i) στο σύνολο  $\{q_1, q_2, (g, \bar{J}) \mapsto q \mid \text{το } \bar{J} \text{ έχει την ίδια μορφή με το } \bar{I} \text{ \& το } g \in \Gamma_{\tau,w} \text{ ικανοποιεί την } c(\bar{J})\}$  των μεταβάσεων με χρώματα που ικανοποιούν αυτή την συνθήκη και ii) στο σύνολο μεταβάσεων  $\{q_1, q_2, (g, \bar{J}) \mapsto q_f \mid \text{το } \bar{J} \text{ έχει την ίδια μορφή με το } \bar{I} \text{ \& το } g \in \Gamma_{\tau,w} \text{ δεν ικανοποιεί τη } c(\bar{J})\}$  που αποτελείται από μεταβάσεις με χρώματα που δεν ικανοποιούν την αντίστοιχη συνθήκη και ουσιαστικά προκύπτουν μέσω της τελευταίας γραμμής του κάθε πίνακα. Η τελευταία γραμμή κάθε πίνακα περιγράφει επίσης όλες τις περιπτώσεις μεταβάσεων για τις οποίες κάποια από τα  $q_1, q_2$  ή  $\bar{I}$  δεν έχουν αναφερθεί στις προηγούμενες γραμμές του πίνακα. Σημειώνουμε ότι δυο  $k$ -μελή διανύσματα  $\bar{J}$  και  $\bar{I}$  θεωρούνται τις ίδιες μορφής όταν οι εμφανίσεις του κενού συνόλου καθώς και μονοσυνόλων συμβαίνουν ακριβώς στις ίδιες θέσεις των δυο αυτών διανυσμάτων. Οι αρχικές μεταβάσεις προκύπτουν από τις μεταβάσεις της μορφής  $q_0, q_0, (\gamma, \bar{I}) \mapsto q$ , δηλαδή για κάθε τέτοια μετάβαση υπάρχει μία αρχική μετάβαση της μορφής  $(\gamma, \bar{I}) \mapsto q$ .

Δεν είναι δύσκολο να επαληθεύσουμε την ορθότητα αυτών των ορισμών για τα αυτόματα των ατομικών τύπων. Δίνουμε την απόδειξη για τον  $\phi = R(x, y)$  όπου το  $R$  ερμηνεύεται ως μία συμμετρική σχέση (η απόδειξη για τον  $\phi = \text{In}(x, X)$  πραγματοποιείται ανάλογα και είναι πιο απλή). Πιο συγκεκριμένα, αποδεικνύουμε αρχικά ότι κάθε ανάθεση που ικανοποιεί τον  $R(x, y)$  υπολογίζεται από μία επιτυχημένη εκτέλεση του  $w$ -dec-Assign $_R$  (\*). Στη συνέχεια αποδεικνύουμε ότι αν το  $\varepsilon$  είναι το μέρος ανάθεσης μιας επιτυχημένης εκτέλεσης του  $w$ -dec-Assign $_R$  τότε το  $\varepsilon$  κωδικοποιεί μία ανάθεση που ικανοποιεί τον  $R(x, y)$  (\*\*).

Συνθήκες/Σύνολα	ως προς $\gamma \in \Gamma_{\tau,w}$	ως προς $\mathcal{I}, \mathcal{A}$ στο $n$
$element(j)$	ανν $j \leq \gamma_1$ & $\forall i \in W ((j, i) \notin \gamma_2)$	ανν $j \leq  \bar{a}_n $ & $\forall i \in W (a_n^j \neq a_{n'}^i)$
$set(I)$	ανν $\forall i \in I (element(i))$	
$SameInPar(J)$	$:= \{i \mid \exists j \in J ((j, i) \in \gamma_2)\}$	$:= \{i \mid \exists j \in J (a_n^j = a_{n'}^i)\}$
$R(j)$	$:= \{i \mid (j, i) \in \gamma_3\}$	$:= \{i \mid (a_n^j, a_n^i) \in R^{\mathcal{A}}\}$
$RInPar(j)$	$:= SameInPar(R(j))$	
$R(j, \ell)$	ανν $element(j)$ & $element(\ell)$ & $(j, \ell) \in \gamma_3$	ανν $element(j)$ & $element(\ell)$ & $(a_n^j, a_n^\ell) \in R^{\mathcal{A}}$

Πίνακας 3.2: Συνθήκες/Σύνολα που εμφανίζονται στους ορισμούς των μεταβάσεων.

Πίνακας 3.3: Η ερμηνεία των καταστάσεων του  $w\text{-dec-Assign}_{R(x,y)}$ .

Κατάσταση	Ερμηνεία
$q_0$	κανένας κόμβος δεν έχει επιλεγεί ακόμα
$q_1^I$ (αντίστοιχα $q_2^I$ )	ένα στοιχείο $a$ (αντίστοιχα $b$ ) έχει ήδη επιλεγεί & το σύνολο $\{a_p^i \mid i \in I\}$ περιέχει κάθε κόμβο $b$ (αντίστοιχα $a$ ) στο $\bar{A}_n$ τ.ω. η $(a, b)$ να είναι μία ανάθεση που ικανοποιεί τον $R(x, y)$
$q_a$	τελική κατάσταση: μία ανάθεση που ικανοποιεί τον $R(x, y)$ έχει επιλεγεί
$q_f$	αποτυχία αποδοχής: μία ανάθεση που δεν ικανοποιεί τον $R(x, y)$ έχει επιλεγεί

Για την απόδειξή του (\*), ας υποθέσουμε ότι η  $(b, c)$  είναι μία ανάθεση που ικανοποιεί τον  $R(x, y)$  πάνω σε μία δομή  $\mathcal{A}$  με δεντροδιάσπαση  $\mathcal{I}$ . Ας υποθέσουμε επίσης ότι  $n = \text{node}_{\mathcal{I}}(b)$  και  $n' = \text{node}_{\mathcal{I}}(c)$ . Διακρίνουμε τις εξής δύο περιπτώσεις:

1. Θεωρούμε αρχικά ότι  $n = n'$ . Στην περίπτωση αυτή, η  $(b, c) \in R^{\mathcal{A}}$  υπολογίζεται από το  $w\text{-dec-Assign}_R$  μέσω της μοναδικής επιτυχημένης εκτέλεσης  $\rho$  η οποία ορίζεται ως εξής:

- $\rho(n) = (q_a, (\{i\}, \{j\}))$  όπου τα  $i, j$  είναι οι θέσεις που εμφανίζονται τα  $b, c$  στο  $\bar{a}_n$  αντίστοιχα. Δηλαδή η συνθήκη  $R(i, j)$  ικανοποιείται στον κόμβο  $n$ .
- $\rho(v) = (q_a, (\emptyset, \emptyset))$  για κάθε κόμβο  $v > n$ .
- $\rho(v) = (q_0, (\emptyset, \emptyset))$  για κάθε κόμβο  $v < n$ .

Πίνακας 3.4: Οι μεταβάσεις του  $w\text{-dec-Assign}_{\text{In}(x,X)}$ .

$q_1, q_2, (\gamma, \bar{I}) \mapsto q, i \in W, I \subseteq W, Q = \{q_0, q_a, q_f\}, F = \{q_a\}$	
Μετάβαση	Συνθήκες που ικανοποιούνται από το $\gamma \in \Gamma_{\tau,w}$
$q_0, q_0, \gamma, (\emptyset, \emptyset) \mapsto q_0$	
$q_0, q_0, \gamma, (\emptyset, I) \mapsto q_0$	$set(I)$
$q_0, q_0, \gamma(\{i\}, I) \mapsto q_a$	$set(I) \ \& \ i \in I$
$q_0, q_a, \gamma, (\emptyset, \emptyset) \mapsto q_a$	
$q_a, q_0, \gamma, (\emptyset, \emptyset) \mapsto q_a$	
$q_0, q_a, \gamma, (\emptyset, I) \mapsto q_a$	$set(I)$
$q_a, q_0, \gamma, (\emptyset, I) \mapsto q_a$	$set(I)$
$q_1, q_2, \gamma, \bar{I} \mapsto q_f$	όλες οι άλλες περιπτώσεις

2. Ας υποθέσουμε τώρα ότι  $n < n'$  (η περίπτωση όπου  $n > n'$  είναι απόλυτα συμμετρική). Στην περίπτωση αυτή, το  $w\text{-dec-Assign}_R$ , διατρέχοντας το  $\mathcal{T}$  από τα φύλλα προς τη ρίζα, υπολογίζει την ανάθεση  $(b, c) \in R^A$  μέσω της επιτυχημένης εκτέλεσης  $\rho$  για την οποία έχουμε:

- $\rho(v) = (q_0, (\emptyset, \emptyset))$  για κάθε κόμβο  $v < n$ .
- $\rho(n) = (q_1^I, (\{i\}, \emptyset))$  όπου το  $i$  είναι η θέση εμφάνισης του  $b$  στο σάκο  $\bar{a}_n$ , και το  $I$  περιέχει κάθε θέση του σάκου  $\bar{a}_p$  του πατέρα του  $n$  στην οποία ένα  $R$ -ταίρι του  $b$  εμφανίζεται. Κατά συνέπεια, η συνθήκη  $element(i) \ \& \ RInPar(i) = I \neq \emptyset$  ικανοποιείται στον κόμβο  $n$ .

Επισημαίνουμε ότι λόγω του Λήμματος 3.2.1, ισχύει ότι

$$\{a_p^i \mid i \in I\} = \{d \in \bar{A}_n \mid (b, d) \in R^A\}$$

και άρα  $c \in \{a_p^i \mid i \in I\}$ .

- $\rho(v) = (q_1^J, (\emptyset, \emptyset))$  για κάθε κόμβο  $v$  που βρίσκεται στο μονοπάτι που ενώνει τα  $n$  και  $n'$ , όπου  $J \subseteq W$  τέτοιο ώστε, όταν ο  $a_p$  συμβολίζει τον σάκο του πατέρα του  $v$  να έχουμε

$$\{a_p^i \mid i \in J\} = \{d \in \bar{A}_v \mid (b, d) \in R^A\}$$

και  $c \in \{a_p^i \mid i \in J\}$ .

Η συνθήκη  $SameInPar(I) = J \neq \emptyset$  ικανοποιείται στο κόμβο  $v$  όταν η  $(q_1^I, (\emptyset, \emptyset))$  είναι η κατάσταση που αντιστοιχεί η  $\rho$  στο παιδί του  $v$  που ανήκει στο μονοπάτι που ενώνει τα  $n$  και  $n'$ .

- $\rho(n') = (q_a, (\emptyset, \{j\}))$  όπου το  $j$  είναι η θέση εμφάνισης του  $c$  στο σάκο  $\bar{a}_{n'}$ . Η συνθήκη  $element(j) \ \& \ j \in I$  ικανοποιείται στον  $n'$  όταν η  $(q_1^I, (\emptyset, \emptyset))$  είναι η κατάσταση που αντιστοιχεί η  $\rho$  στο παιδί του  $n'$  που ανήκει στο μονοπάτι που ενώνει τα  $n$  και  $n'$ .

Πίνακας 3.5: Οι μεταβάσεις του  $w\text{-dec-Assign}_{R(x,y)}$ .

$q_1, q_2, (\gamma, \bar{I}) \mapsto q, j, \ell \in W, I, J \subseteq W$	
Μετάβαση	Συνθήκες που ικανοποιούνται
$q_0, q_0, \gamma, (\emptyset, \emptyset) \mapsto q_0$	
$q_0, q_0, \gamma, (\{j\}, \emptyset) \mapsto q_1^I$	$element(j) \ \& \ RInPar(j) = I \neq \emptyset$
$q_0, q_0, \gamma, (\emptyset, \{j\}) \mapsto q_2^I$	
$q_0, q_0, \gamma, (\{j\}, \{\ell\}) \mapsto q_a$	$R(j, \ell)$
$q_0, q_1^J, \gamma, (\emptyset, \emptyset) \mapsto q_1^I$	$SameInPar(J) = I \neq \emptyset$
$q_1^J, q_0, \gamma, (\emptyset, \emptyset) \mapsto q_1^I$	
$q_0, q_2^J, \gamma, (\emptyset, \emptyset) \mapsto q_2^I$	
$q_0, q_1^I, \gamma, (\emptyset, \{j\}) \mapsto q_a$	$element(j) \ \& \ j \in I$
$q_1^I, q_0, \gamma, (\emptyset, \{j\}) \mapsto q_a$	
$q_0, q_2^I, \gamma, (\{j\}, \emptyset) \mapsto q_a$	
$q_2^I, q_0, \gamma, (\{j\}, \emptyset) \mapsto q_a$	
$q_0, q_a, \gamma, (\emptyset, \emptyset) \mapsto q_a$	
$q_a, q_0, \gamma, (\emptyset, \emptyset) \mapsto q_a$	
$q_1, q_2, \gamma, \bar{I} \mapsto q_f$	όλες οι άλλες περιπτώσεις

- $\rho(v) = (q_a, (\emptyset, \emptyset))$  για κάθε  $v > n'$ .
- $\rho(v) = (q_0, (\emptyset, \emptyset))$  για κάθε  $v \in T_u$  όπου το  $u$  είναι το παιδί του  $n'$  το οποίο δεν ανήκει στο μονοπάτι που ενώνει τα  $n$  και  $n'$ .

Επομένως, έχουμε αποδείξει ότι κάθε ανάθεση που ικανοποιεί τον  $R(x, y)$  υπολογίζεται από μία μοναδική επιτυχημένη εκτέλεση του  $w\text{-dec-Assign}_R$ .

Για το (\*\*), θα πρέπει να αποδείξουμε ότι οι επιτυχημένες εκτελέσεις του  $w\text{-dec-Assign}_R$  αντιστοιχούν πλήρως στα στοιχεία του  $R^A$ . Αρκεί κανείς να παρατηρήσει ότι άμεση συνέπεια του ορισμού του  $w\text{-dec-Assign}_R$  είναι ότι οι επιτυχημένες εκτελέσεις του έχουν ακριβώς τη μορφή των εκτελέσεων που ορίζονται στις περιπτώσεις (1) και (1) της απόδειξης του (\*) παραπάνω. Δηλαδή πράγματι κωδικοποιούν μόνο επιτυχημένες αναθέσεις.

Η απόδειξη δίνεται για την περίπτωση μιας δυαδικής συμμετρικής σχέσης για λόγους απλοποίησης, αλλά η κατασκευή γενικεύεται ιδιαίτερα εύκολα. Για παράδειγμα, ας υποθέσουμε ότι η  $R$  δεν αντιστοιχεί σε μία συμμετρική σχέση. Το αντίστοιχο αυτόματο προκύπτει μέσω μιας μικρής τροποποίησης του ορισμού του  $w\text{-dec-Assign}_R$ . Πιο συγκεκριμένα, ας υποθέσουμε ότι

$$\begin{aligned}
 R_1(j) &= \{i \mid (i, j) \in \gamma_3\} \\
 R_2(j) &= \{i \mid (j, i) \in \gamma_3\} \\
 R_i InPar(j) &= SameInPar(R_i(j)) \text{ για } i = 1, 2.
 \end{aligned}$$

Αρκεί να αντικαταστήσουμε το δεύτερο σύνολο μεταβάσεων του Πίνακα 3.5 με τις μεταβάσεις

$$\begin{aligned} q_0, q_0, \gamma, (\{j\}, \emptyset) &\mapsto q_1^I \quad \text{με συνθήκη } element(j) \ \& \ R_2 InPar(j) = I \neq \emptyset, \text{ και} \\ q_0, q_0, \gamma, (\emptyset, \{j\}) &\mapsto q_2^I \quad \text{με συνθήκη } element(j) \ \& \ R_1 InPar(j) = I \neq \emptyset. \end{aligned}$$

Ανάλογα, τα αυτόματα αποδόμησης για έναν  $k$ -μελή ατομικό τύπο  $P(x_1, \dots, x_k)$  αποτελούν γενίκευση του αυτομάτου αποδόμησης για τον δυαδικό ατομικό τύπο  $R(x, y)$ . Πιο συγκεκριμένα, οι καταστάσεις του  $w$ -dec-Assign $_P$  θα έχουν τώρα τη μορφή  $q_0, q_a, q_f$  και  $q^{I_1, \dots, I_k}$  όπου  $(I_1, \dots, I_k) \in (\mathcal{P}(W))^k$ .

Ας υποθέσουμε ότι ο  $\phi$  είναι ένας μη ατομικός τύπος. Οι ορισμοί των αυτομάτων  $w$ -dec- $A_\phi$  για τις περιπτώσεις όπου  $\phi = \phi_1 \wedge \phi_2$  και  $\phi = \exists X \psi$  ακολουθούν τους αντίστοιχους κλασικούς επαγωγικούς ορισμούς των ντετερμινιστικών αυτομάτων για δέντρα. Πιο συγκεκριμένα, στη πρώτη περίπτωση το αυτόματο αντιστοιχεί στην τομή των αυτομάτων  $w$ -dec- $A_{\phi_1}$  και  $w$ -dec- $A_{\phi_2}$ . Στη δεύτερη περίπτωση το αυτόματο προκύπτει μετά από μία κατάλληλη προβολή πάνω στο  $w$ -dec- $A_\phi$  και έναν περαιτέρω ντετερμινισμό του αυτομάτου που προκύπτει από την προβολή. Στο Παράρτημα Α' υπάρχουν αναλυτικοί ορισμοί για αυτές τις κατασκευές.

Η μοναδική περίπτωση που απαιτεί ειδική αντιμετώπιση είναι αυτή τύπων της μορφής

$$\phi = \neg\psi.$$

Η συνήθης κατασκευή ορίζει το  $A_\phi$  ως το συμπλήρωμα  $A_\psi^{\text{compl}}$  του  $A_\psi$ . Ωστόσο, το γεγονός ότι τα αυτόματά μας αποδέχονται αναθέσεις που είναι κωδικοποιημένες με συγκεκριμένο τρόπο, έχει ως συνέπεια το συμπλήρωμα να μην αποτελεί ορθή κατασκευή στην περίπτωση μας. Πιο συγκεκριμένα, το  $w$ -dec- $A_\psi$  δέχεται δέντρα που κωδικοποιούν αναθέσεις με τη μορφή των απεικονίσεων αναθέσεων, σύμφωνα με τον Ορισμό 3.2.2 που βασίζεται στην έννοια του αντιπροσωπευτικού ζευγαριού ενός στοιχείου. Δηλαδή το  $w$ -dec- $A_\psi$  απορρίπτει εκείνα να δέντρα που κωδικοποιούν αναθέσεις με απεικονίσεις οι οποίες δεν είναι απεικονίσεις αναθέσεων. Υπάρχουν όμως απεικονίσεις που δεν έχουν τη μορφή απεικονίσεων αναθέσεων οι οποίες αντιστοιχούνται μέσω της  $\text{dec}_{k,w}$  σε αναθέσεις που ικανοποιούν τον  $\phi$ . Ωστόσο τα δέντρα που κωδικοποιούν αυτές τις αναθέσεις, καθώς απορρίπτονται από το  $w$ -dec- $A_\psi$ , γίνονται δεκτά από το  $w$ -dec- $A_\psi^{\text{compl}}$ . Έτσι, ορίζοντας το  $w$ -dec- $A_\phi$ ,  $\phi = \neg\psi$ , να είναι ίδιο με το  $w$ -dec- $A_\psi^{\text{compl}}$  θα έχουμε ένα αυτόματο που δέχεται και αναθέσεις που ικανοποιούν τον  $\psi$ .

Το προβληματικό αυτό σημείο μπορεί να διορθωθεί, αρκεί να ορίσουμε το  $w$ -dec- $A_\phi$  έτσι ώστε να δέχεται μόνο εκείνες από τις αναθέσεις που δέχεται το  $w$ -dec- $A_\psi^{\text{compl}}$  οι οποίες είναι αναθέσεις απεικόνισης. Πιο συγκεκριμένα, για τον  $\psi(X_1, \dots, X_k)$  ορίζουμε το  $w$ -dec- $A_\phi$  ως την τομή των αυτομάτων

$$w\text{-dec-}A_\psi^{\text{compl}} \quad \& \quad w\text{-dec-}A_{\text{Set}^k}$$

όπου το  $w$ -dec- $A_{\text{Set}^k}$  είναι ένα αυτόματο που δέχεται μόνο δέντρα που επεκτείνονται με κωδικοποιήσεις αναθέσεων  $(B_1, \dots, B_k)$  μέσω της  $\text{enc}_{k,w}$  του Ορισμού 3.2.2.

Ανάλογα, όταν ο  $\psi$  έχει τη μορφή  $\psi(x_1, \dots, x_u, X_1, \dots, X_v)$ , το  $w$ -dec- $A_\phi$  είναι η τομή των αυτομάτων

$$w\text{-dec-}A_{\psi}^{\text{compl}}, \quad w\text{-dec-}A_{\text{Element}^u} \quad \& \quad w\text{-dec-}A_{\text{Set}^v}$$

όπου το  $w\text{-dec-}A_{\text{Element}^u}$  είναι το αυτόματο που δέχεται μόνο δέντρα που επεκτείνονται με κωδικοποιήσεις, μέσω της  $\text{enc}_{k,w}$ , αναθέσεων της μορφής  $(\{b_1\}, \dots, \{b_u\})$ . Οι ορισμοί αυτών των αυτομάτων δίνονται στο Παράρτημα Γ'.  $\square$

Στην επόμενη ενότητα παρουσιάζονται βελτιστοποιημένοι αλγόριθμοι για την αποτίμηση των αυτομάτων αποδόμησης πάνω σε δέντρα δομών. Οι ορισμοί που δίνονται παρακάτω αποτελούν διαφορετικές αναπαραστάσεις του  $w\text{-dec-Assign}_{\phi}$ , και βασίζονται ουσιαστικά στις συναρτήσεις μετάβασης  $w\text{-}\delta_0$ ,  $w\text{-}\delta$  του ντετερμινιστικού αυτομάτου  $w\text{-dec-}A_{\phi}$ . Οι αναπαραστάσεις αυτές εμφανίζονται στα datalog προγράμματα των επόμενων ενοτήτων, με τρόπο ακριβώς ανάλογο που οι αντίστοιχες αναπαραστάσεις του Ορισμού 2.2.3 χρησιμοποιούνται στα datalog προγράμματα του προηγούμενου κεφαλαίου. Ο υπολογισμός του μεγέθους των συνόλων μεταβάσεων που ορίζουμε βασίζεται στη σχέση των τελευταίων με τις συναρτήσεις μετάβασης  $w\text{-}\delta_0$ ,  $w\text{-}\delta$  του  $w\text{-dec-}A_{\phi}$ .

**Ορισμός 3.2.6.** Ας υποθέσουμε ότι ο  $\phi$  είναι ένας  $\text{MSO}[\tau]$  τύπος με  $k$  ελεύθερες μεταβλητές και ότι

$$w\text{-dec-Assign}_{\phi} = (\Gamma_{\tau,w}, Q \times (\mathcal{P}(W))^k, w\text{-}\Delta_0, w\text{-}\Delta, F \times (\mathcal{P}(W))^k).$$

Ονομάζουμε  $\phi$ -καταστάσεις τα στοιχεία του συνόλου  $Q$  και  $\phi$ -μεταβάσεις  $w\text{-δεντροδιασπάσεων}$  τα στοιχεία των συνόλων  $w\text{-}D_0$ ,  $w\text{-}D'_0$ ,  $w\text{-}D$  και  $w\text{-}D'$  που ορίζονται ως εξής:

- $w\text{-}D_0 = \{\gamma q \bar{I} \mid (\gamma, (q, \bar{I})) \in w\text{-}\Delta_0\}$  με πληθάρημο  $|\Gamma_{\tau,w}| \cdot 2^{k \cdot (w+1)}$ ,
- $w\text{-}D'_0 = \{\gamma q \mid \text{υπάρχει } \bar{I} \text{ τ.ω. } \gamma q \bar{I} \in w\text{-}D_0\}$  με πληθάρημο το πολύ  $|\Gamma_{\tau,w}| \cdot 2^{k \cdot (w+1)}$ ,
- $w\text{-}D = \{\gamma q q_1 q_2 \bar{I} \mid \text{υπάρχουν } \bar{I}_1, \bar{I}_2 \text{ τ.ω. } ((q_1, \bar{I}_1), (q_2, \bar{I}_2), \gamma, (q, \bar{I})) \in w\text{-}\Delta\}$   
με πληθάρημο  $|\Gamma_{\tau,w}| \cdot |Q|^2 \cdot 2^{k \cdot (w+1)}$  και
- $w\text{-}D' = \{\gamma q q_1 q_2 \mid \text{υπάρχει } \bar{I} \text{ τ.ω. } \gamma q q_1 q_2 \bar{I} \in w\text{-}D\}$   
με πληθάρημο το πολύ  $|\Gamma_{\tau,w}| \cdot |Q|^2 \cdot 2^{k \cdot (w+1)}$ .

Για  $k = 1$ , ορίζουμε επίσης τα

- $w\text{-}D_0^c = \{\gamma q | I \mid \gamma q I \in w\text{-}D_0\}$  και
- $w\text{-}D^c = \{\gamma q q_1 q_2 | I \mid \gamma q q_1 q_2 I \in w\text{-}D\}$ .

### 3.3 Η datalog λύση στα προβλήματα MSO αποτίμησης

Οι βασικές έννοιες της datalog δίνονται στη Ενότητα 2.4.1 του προηγούμενου κεφαλαίου.

Ο ορισμός του datalog ορίσμου ερωτήματος προσαρμόζεται στις δομές φραγμένους δεντροπλάτους ως εξής: μια απεικόνιση (αντίστοιχα boolean απεικόνιση)  $f$  πάνω σε μία κλάση  $\mathcal{C}_{\tau}$

από πεπερασμένες  $\tau$ -δομές ονομάζεται *a-datalog ορίσιμη* αν υπάρχει ένα datalog ερώτημα  $Q_f$  βαθμού  $a$  τέτοιο ώστε, για κάθε  $\tau$ -δομή  $\mathcal{A} \in \mathcal{C}_\tau$ ,

$$f(\mathcal{A}) = Q_f(\mathcal{D}_{\mathcal{A}})$$

(αντίστοιχα  $f(\mathcal{A}) = 1$  αν  $Q_f(\mathcal{D}_{\mathcal{A}}) \neq \emptyset$ ), όπου η βάση δεδομένων  $\mathcal{D}_{\mathcal{A}}$  αντιστοιχεί στη δομή  $\mathcal{A}$ . Καθώς εκτός από τη δομή  $\mathcal{A}$  έχουμε πάντα ως μέρος της εισόδου μία ειδική δεντροδιάσπαση  $\mathcal{I}$  της  $\mathcal{A}$ , θεωρούμε ότι το  $\mathcal{D}_{\mathcal{A}}$  στον παραπάνω ορισμό περιέχει επίσης γεγονότα που κωδικοποιούν την  $\mathcal{I}$ .

Σημειώνουμε επίσης ότι ένα datalog<sup>-</sup> πρόγραμμα με ανισότητα και άρνηση που εμφανίζεται μόνο στα EDB κατηγορήματα ονομάζεται *ημιθετικό* (semipositive) datalog πρόγραμμα.

### 3.3.1 Από την MSO στην datalog: η κεντρική ιδέα της αναγωγής

Στην ενότητα αυτή ανάγουμε το πρόβλημα αποτίμησης της MSO σε δομές φραγμένου δεντροπλάτους σε ένα πρόβλημα αποτίμησης της datalog, γενικεύοντας έτσι την datalog λύση, που παρουσιάστηκε στην Ενότητα 2.4 του προηγούμενου κεφαλαίου, για το πρόβλημα αποτίμησης της MSO σε δέντρα. Η αναγωγή αυτή βασίζεται στην αναγωγή του αρχικού προβλήματος αποτίμησης της MSO σε ένα πρόβλημα αποτίμησης των αυτομάτων αποδόμησης. Πιο συγκεκριμένα, όπως κάναμε και στη περίπτωση των δέντρων, τα datalog προγράμματα που παρουσιάζουμε επιλύουν σε κάθε περίπτωση το αντίστοιχο πρόβλημα αυτομάτων.

Καθώς και στις δύο περιπτώσεις, δηλαδή τόσο στα δέντρα όσο και στις δομές φραγμένου δεντροπλάτους, τα datalog προγράμματα αποτιμούν αυτόματα δέντρων, ο βασικός πυρήνας των αντίστοιχων datalog λύσεων είναι στην ουσία ο ίδιος. Εντούτοις, η γενίκευση αυτή δεν είναι σε καμία περίπτωση τετριμμένη, με την έννοια ότι χρειάστηκε να αναπτύξουμε τη θεωρία των αυτομάτων αποδόμησης έτσι ώστε να μπορούμε να έχουμε ανάλογα datalog προγράμματα που αποτιμούν τα αυτόματα αποδόμησης. Πιο συγκεκριμένα, ορίσαμε τα αυτόματα αποδόμησης με αυτόν τον τρόπο έχοντας εκ των προτέρων στο νου μας την δυνατότητα εφαρμογής μιας ενιαίας ομοιόμορφης datalog λύσης. Επισημαίνουμε επίσης ότι χάρη στα αυτόματα αποδόμησης καταλήγουμε σε ενδιαφέροντα αποτελέσματα εκφρασιμότητας για την datalog.

Παρόλα αυτά, υπάρχουν σημαντικές διαφορές στη λύση που αφορά στις δομές φραγμένου δεντροπλάτους η οποία είναι πολύ πιο σύνθετη από αυτή που αφορά στα δέντρα. Πιο συγκεκριμένα, τα datalog προγράμματα που ορίζονται σε αυτό το κεφάλαιο έχουν ως είσοδο δύο διαφορετικές βάσεις δεδομένων, μία που αποτελείται από γεγονότα που κωδικοποιούν την  $\tau$ -δομή  $\mathcal{A}$  και μία δεύτερη που αποτελείται από γεγονότα που κωδικοποιούν την ειδική δεντροδιάσπαση  $\mathcal{I}$  του  $\mathcal{A}$  με πλάτος  $w$ . Επιπρόσθετα, αυτά τα datalog προγράμματα, υλοποιώντας την πιο σύνθετη απεικόνιση αποκωδικοποίησης  $\text{dec}_{k,w}$ , υπολογίζουν τις αναθέσεις που ικανοποιούν έναν δεδομένο MSO τύπο  $\phi$  με έναν πιο σύνθετο τρόπο που απαιτεί και τη συμμετοχή της βάσης δεδομένων που αντιστοιχεί στη δεντροδιάσπαση.

Τα βασικά σημεία της datalog λύσης συνοψίζονται ως εξής:

- Χρησιμοποιώντας ως είσοδο μία βάση δεδομένων με γεγονότα που αντιστοιχούν σε μία δεδομένη δομή  $\mathcal{A}$  και μία δεδομένη δεντροδιάσπαση  $\mathcal{I}$  σύμφωνα με τον Ορισμό 3.3.1,



γράφουμε κανόνες που παράγουν γεγονότα που περιγράφουν τη συνάρτηση χρωματισμού του δέντρου  $\mathcal{T}_{\mathcal{I};\mathcal{A}}$  που κωδικοποιεί τα  $\mathcal{A}$  και  $\mathcal{I}$ . Οι κανόνες αυτοί αποτελούν το πρόγραμμα  $\Pi_{\tau,w}$  του Λήμματος 3.3.1.

Σημειώνουμε ότι το βήμα αυτό δεν υπάρχει στην περίπτωση της datalog λύσης για τα δέντρα, όπου τα χρώματα του δεδομένου δέντρου αποτελούν μέρος της βάσης δεδομένων που είναι η είσοδος του αντίστοιχου προγράμματος.

- Στη συνέχεια, χρησιμοποιώντας τις μεταβάσεις του  $w$ -dec-Assign $_{\phi}$  γράφουμε κανόνες που προσομοιώνουν τις εκτελέσεις του  $w$ -dec-Assign $_{\phi}$  πάνω στο  $\mathcal{T}_{\mathcal{I};\mathcal{A}}$ .
- Τελικά, μέσω κανόνων που συνδυάζουν κατάλληλα την πληροφορία που αφορά στις εκτελέσεις του  $w$ -dec-Assign $_{\phi}$  πάνω στο  $\mathcal{T}_{\mathcal{I};\mathcal{A}}$  και τα γεγονότα που περιγράφουν την δεντροδιάσπαση  $\mathcal{I}$ , παράγεται το σύνολο  $\phi(\mathcal{A})$ .

### 3.3.2 Η κατασκευή των δέντρων δομών

Ορίζουμε τις βάσεις δεδομένων που περιγράφουν μία δεδομένη δομή  $\mathcal{A}$  και μία δεδομένη δεντροδιάσπαση  $\mathcal{I}$ . Στη συνέχεια κατασκευάζουμε ένα datalog πρόγραμμα που με είσοδο αυτές τις βάσεις δεδομένων παράγει το δέντρο  $\mathcal{T}_{\mathcal{I};\mathcal{A}}$  της δομής  $\mathcal{A}$  που κωδικοποιεί τα  $\mathcal{A}$  και  $\mathcal{I}$ .

**Ορισμός 3.3.1.** Ορίζουμε τη βάση δεδομένων  $\mathcal{D}_{\mathcal{A}}$  μιας δομής  $\mathcal{A}$ , με λεξικό  $\tau = \{R_1, \dots, R_{\ell}\}$  όπου κάθε  $R_i$  έχει βαθμό  $r_i$ , κατά τον φυσικό τρόπο:

- $r_i(a_1, \dots, a_{r_i}) \in \mathcal{D}_{\mathcal{A}}$  αν  $\mathcal{A} \models R_i(a_1, \dots, a_{r_i})$ ,  $1 \leq i \leq \ell$ .

Δηλαδή η  $\mathcal{D}_{\mathcal{A}}$  είναι μία βάση δεδομένων πάνω στο σύμπαν  $A$  της  $\mathcal{A}$ .

Ας υποθέσουμε ότι η  $\mathcal{I} = (\mathcal{T}, (\bar{a}_n)_{n \in \mathcal{T}})$  είναι μία ειδική δεντροδιάσπαση πλάτους  $w$  της δομής  $\mathcal{A}$ . Η βάση δεδομένων  $\mathcal{D}_{\mathcal{I}}$  της  $\mathcal{I}$  ορίζεται ως εξής:

- $\text{leaf}(n) \in \mathcal{D}_{\mathcal{I}}$  αν το  $n$  είναι φύλλο του  $\mathcal{T}$
- $\text{root}(n) \in \mathcal{D}_{\mathcal{I}}$  αν το  $n$  είναι η ρίζα του  $\mathcal{T}$
- $\text{succ}_0(n, n_1), \text{succ}_1(n, n_2) \in \mathcal{D}_{\mathcal{I}}$  αν τα  $n_1, n_2$  είναι αντίστοιχα το αριστερό, δεξί παιδί του  $n$  στο  $\mathcal{T}$
- $\text{decomp}^s(n, a_n^1, \dots, a_n^s) \in \mathcal{D}_{\mathcal{I}}$  αν  $\bar{a}_n = (a_n^1, \dots, a_n^s)$
- $\text{decomp-}i(n, a_n^i) \in \mathcal{D}_{\mathcal{I}}$  αν  $\bar{a}_n = (a_n^1, \dots, a_n^s)$ ,  $1 \leq i \leq s \leq w + 1$
- $\text{decomp-}I(n, \{a_n^i \mid i \in I\}) \in \mathcal{D}_{\mathcal{I}}$  αν  $\bar{a}_n = (a_n^1, \dots, a_n^s)$ ,  $1 \leq s \leq w + 1$  και  $I \subseteq \{1, \dots, s\}$ .

**Παράδειγμα 3.3.1.** Ας υποθέσουμε ότι τα  $\mathcal{G}$  και  $\mathcal{I}$  είναι όπως στο Παράδειγμα 3.1.1 της σελίδας 88. Η βάση δεδομένων  $\mathcal{D}_{\mathcal{G}}$  αποτελείται από τα παρακάτω γεγονότα:

- $\text{edge}(a, b), \text{edge}(a, c), \text{edge}(a, g), \text{edge}(b, c), \text{edge}(c, d), \text{edge}(c, e), \text{edge}(c, f), \text{edge}(d, e), \text{edge}(f, g), \dots, \text{edge}(h, g)$ .

Η βάση δεδομένων  $\mathcal{D}_{\mathcal{I}}$  αποτελείται από τα εξής γεγονότα:

- $\text{leaf}(n_3), \text{leaf}(n_4), \text{leaf}(n_5), \text{root}(n_1)$
- $\text{succ}_0(n_1, n_2), \text{succ}_1(n_1, n_3), \text{succ}_0(n_2, n_4), \text{succ}_1(n_2, n_5)$
- $\text{decomp-1}(n_1, a), \text{decomp-2}(n_1, g), \text{decomp-3}(n_1, f) \dots$
- $\text{decomp-}\{1\}(n_1, \{a\}), \text{decomp-}\{2\}(n_1, \{g\}), \text{decomp-}\{3\}(n_1, \{f\}),$   
 $\text{decomp-}\{1, 2\}(n_1, \{a, g\}), \text{decomp-}\{1, 3\}(n_1, \{a, f\}), \text{decomp-}\{2, 3\}(n_1, \{g, f\}),$   
 $\text{decomp-}\{1, 2, 3\}(n_1, \{a, g, f\}) \dots$

Το λήμμα που ακολουθεί δηλώνει ότι υπάρχει ένα datalog πρόγραμμα το οποίο με είσοδο τη βάση δεδομένων  $\mathcal{D}_{\mathcal{A}} \cup \mathcal{D}_{\mathcal{I}}$  παράγει τη συνάρτηση χρωματισμού του δέντρου  $\mathcal{T}_{\mathcal{I};\mathcal{A}}$ .

**Λήμμα 3.3.1.** Ας υποθέσουμε ότι  $\tau = \{R_1, \dots, R_\ell\}$  και ότι το  $w$  είναι ένας θετικός ακέραιος. Για κάθε  $\tau$ -δομή  $\mathcal{A}$  με δεντροπλάτος το πολύ  $w$  και μία ειδική δεντροδιάσπαση  $\mathcal{I}$  του  $\mathcal{A}$ , υπάρχει ένα μοναδιαίο ημιθετικό datalog πρόγραμμα

$$\Pi_{\tau, w}$$

τέτοιο ώστε με είσοδο τη βάση δεδομένων  $\mathcal{D}_{\mathcal{A}} \cup \mathcal{D}_{\mathcal{I}}$  να παράγει τα χρώματα του  $\mathcal{T}_{\mathcal{I};\mathcal{A}} = (\mathcal{T}, c)$  με τη μορφή γεγονότων ως εξής:

$$(\Pi_{\tau, w}, \mathcal{D}_{\mathcal{A}} \cup \mathcal{D}_{\mathcal{I}}) \models \text{color1-}\gamma_1(n), \dots, \text{color}(\ell + 2)\text{-}\gamma_{\ell+2}(n) \quad \text{ανν} \quad c(n) = (\gamma_1, \dots, \gamma_{\ell+2})$$

Η πολυπλοκότητα του προγράμματος είναι  $(\ell + 2) \cdot |T|$ .

**Απόδειξη.** Είναι εύκολο να επιβεβαιώσουμε ότι το παρακάτω πρόγραμμα ικανοποιεί τις επιθυμητές συνθήκες.

$$\begin{aligned} \Pi_{\tau, w}. \quad c_{1,s} : & \quad \text{color1-}s(x) \leftarrow \text{decomp}^s(x, x_1, \dots, x_s) \\ c_{2,u,v,P}^k : & \quad \text{color2-}P(x) \leftarrow \text{decomp}^u(x, x_1, \dots, x_u), \text{decomp}^v(y, y_1, \dots, y_v), \\ & \quad \text{succ}_k(y, x), \{x_i = y_j \mid (i, j) \in P\}, \\ & \quad \{\neg(x_i = y_j) \mid (i, j) \notin P\} \quad \text{για } k = 0, 1 \\ c_2 : & \quad \text{color2-}\emptyset(x) \leftarrow \text{root}(x) \\ c_{2+i,s,S_i} : & \quad \text{color}(i + 2)\text{-}S_i(x) \leftarrow \text{decomp}^s(x, x_1, \dots, x_s), \\ & \quad \{r_i(x_{j_1}, \dots, x_{j_{r_i}}) \mid (j_1, \dots, j_{r_i}) \in S_i\}, \\ & \quad \{\neg r_i(x_{j_1}, \dots, x_{j_{r_i}}) \mid (j_1, \dots, j_{r_i}) \notin S_i\} \quad \text{για } 1 \leq i \leq \ell \end{aligned}$$

όπου  $s, u, v \in W$ ,  $P \in \mathcal{P}(W_u \times W_v)$ ,  $S_1 \in \mathcal{P}(W_s^{r_1}), \dots, S_\ell \in \mathcal{P}(W_s^{r_\ell})$  για  $W = \{1, \dots, w + 1\}$  και  $W_i = \{1, \dots, i\}$ .

*Πολυπλοκότητα.* Το πρόγραμμα  $\Pi_{\tau, w}$  πραγματοποιεί ακριβώς  $(\ell + 2) \cdot |T|$  στιγμιοτυποποιήσεις κανόνων.  $\square$

### 3.3.3 Αποτίμηση μονομελών MSO-ορίσιμων ερωτημάτων μέσω της μοναδιαίας datalog

Τα μονομελή MSO-ορίσιμα ερωτήματα πάνω σε δομές φραγμένου δεντροπλάτους είναι ερωτήματα που ορίζονται από  $\text{MSO}[\tau]$  τύπους της μορφής  $\phi(x)$ , δηλαδή τύπους που έχουν μία πρωτοβάθμια ελεύθερη μεταβλητή. Από το Θεώρημα 3.2.1 έχουμε ότι για κάθε τέτοιο τύπο  $\phi(x)$ , υπάρχει ένα αυτόματο αποδόμησης  $w\text{-dec-Assign}_\phi$  τέτοιο ώστε για κάθε  $\tau$ -δομή  $\mathcal{A}$  με δεντροπλάτος το πολύ  $w$  και ειδική δεντροδιάσπαση  $\mathcal{I}$  της  $\mathcal{A}$  να ισχύει

$$a \in \phi(\mathcal{A}) \text{ ανν } a \in w\text{-dec-Assign}_\phi(\mathcal{T}_{\mathcal{I};\mathcal{A}}).$$

Δηλαδή  $a \in \phi(\mathcal{A})$  ανν υπάρχει μία επιτυχημένη εκτέλεση του  $w\text{-dec-Assign}_\phi$  στο  $\mathcal{T}_{\mathcal{I};\mathcal{A}}$  με μέρος ανάθεσης

$$\varepsilon_a : T \rightarrow \mathcal{P}(W)$$

όπου  $\varepsilon_a(n) = \emptyset$  για κάθε  $n \neq \text{node}_{\mathcal{I}}(a)$ , και  $\varepsilon_a(n) = \{i\}$  ανν  $\text{pair}_{\mathcal{I}}(a) = (n, i)$ .

Η παραπάνω απεικόνιση περιγράφει τη μορφή κάθε μέρους ανάθεσης που αντιστοιχεί σε μία επιτυχημένη εκτέλεση του  $w\text{-dec-Assign}_\phi$ . Πιο συγκεκριμένα, για κάθε  $a \in \mathcal{A}$  τέτοιο ώστε  $\text{pair}_{\mathcal{I}}(a) = (n, i)$  έχουμε ότι  $a \in \phi(\mathcal{A})$  ανν υπάρχει μία επιτυχημένη εκτέλεση του  $w\text{-dec-Assign}_\phi$  στο  $\mathcal{T}_{\mathcal{I};\mathcal{A}}$  που αναθέτει στον κόμβο  $n$  την κατάσταση  $q\{i\}$  για κάποιο  $q \in Q$ . Όμως από τον ορισμό του ζευγαριού ενός στοιχείου έχουμε ότι αν  $\text{pair}_{\mathcal{I}}(a) = (n, i)$  τότε  $a = a_n^i$ , όπου το  $a_n^i$  είναι το  $i$ -οστό στοιχείο του σάκου  $\bar{a}_n$  του κόμβου  $n$  μιας δεντροδιάσπασης. Κατά συνέπεια ισχύει το εξής:

το  $a_n^i$  είναι μία ανάθεση που ικανοποιεί τον  $\phi(x)$  στην  $\mathcal{A}$  ανν στον κόμβο  $n$  υπάρχει μία επιτυχημένη κατάσταση του  $w\text{-dec-Assign}_\phi$  στο  $\mathcal{T}_{\mathcal{I};\mathcal{A}}$  της μορφής  $q\{i\}$ .

Έτσι, όπως και στην περίπτωση των δέντρων, αρκεί να έχουμε datalog κανόνες που υπολογίζουν επιτυχημένες καταστάσεις και συλλέγουν κάθε στοιχείο  $a_n^i$  για το οποίο υπάρχει μία επιτυχημένη κατάσταση της μορφής  $q\{i\}$  στον κόμβο  $n$ . Οι κανόνες αυτοί συνθέτουν ένα βελτιστοποιημένο μοναδιαίο datalog πρόγραμμα που αποτιμά τα μονομελή MSO-ορίσιμα ερωτήματα.

Θυμίζουμε ότι για την αποτίμηση των μονομελών ερωτημάτων στα δέντρα –βλέπε την Ενότητα 2.4.5– ισχύει το εξής: το  $\{n\}$  είναι μία ανάθεση που ικανοποιεί τον  $\phi$  ανν ο κόμβος  $n$  έχει μία επιτυχημένη κατάσταση της μορφής  $(q, 1)$ . Έτσι, και σε εκείνη την περίπτωση ήταν επαρκές να υπολογίσουμε τις επιτυχημένες καταστάσεις του αυτομάτου σε κάθε κόμβο. Τα προγράμματα που παράγουν επιτυχημένες καταστάσεις αυτομάτων για δέντρα, δηλαδή το  $P' \cup S'$  στην περίπτωση των δέντρων και το  $P^* \cup S^*$  εδώ, είναι τα ίδια και στις δύο περιπτώσεις.

**Πρόταση 3.3.1.** Ας υποθέσουμε ότι  $\tau = \{R_1, \dots, R_n\}$ . Για κάθε  $\text{MSO}[\tau]$  τύπο  $\phi(x)$  και ακέραιο  $w \geq 1$ , υπάρχει ένα μοναδιαίο ημιθετικό datalog<sup>+</sup> ερώτημα

$$Q_\phi^w = (\Pi_\phi^w, \text{assign})$$

τέτοιο ώστε για κάθε  $\tau$ -δομή  $\mathcal{A}$  με δεντροπλάτος το πολύ  $w$  και ειδική δεντροδιάσπαση  $\mathcal{I} = (\mathcal{T}, (\bar{a}_n)_{n \in \mathcal{T}})$  να ισχύει

$$a \in \phi(\mathcal{A}) \text{ ανν } a \in Q_\phi^w(\mathcal{D}_\mathcal{A} \cup \mathcal{D}_\mathcal{I}).$$

Η πολυπλοκότητα του υπολογισμού είναι το πολύ  $(\ell + 2 + 2^{w+2} \cdot s^2) \cdot |\mathcal{T}| + a$  όπου  $s$  είναι το μέγεθος του συνόλου  $Q$  των  $\phi$ -καταστάσεων,  $m$  είναι ο αριθμός κόμβων του  $\mathcal{T}$  και  $a = |\phi(\mathcal{T})|$ .

**Απόδειξη.** Ορίζουμε το  $\Pi_\phi^w = \Pi_{\tau,w} \cup P^* \cup S^* \cup_{i,q} r_{q,i}$ . Το πρόγραμμα  $\Pi_{\tau,w}$  παράγει τη συνάρτηση χρωματισμού του  $\mathcal{T}_{\mathcal{I},\mathcal{A}}$ . Αυτή χρησιμοποιείται στη συνέχεια ως είσοδος, μαζί με τη δεντρική δομή  $\mathcal{T}$  του  $\mathcal{T}_{\mathcal{I},\mathcal{A}}$ , στο πρόγραμμα  $P^* \cup S^*$  που υπολογίζει τις επιτυχημένες καταστάσεις του  $w$ -dec-Assign $_\phi$  πάνω στο  $\mathcal{T}_{\mathcal{I},\mathcal{A}}$ . Τελικά, οι κανόνες  $r_{q,i}$  αποκωδικοποιούν τις επιτυχημένες καταστάσεις, με τη βοήθεια της  $\mathcal{D}_\mathcal{I}$ , στα στοιχεία του  $\mathcal{A}$  που αποτελούν το  $\phi(\mathcal{A})$ .

Αναλυτικότερα, το πρόγραμμα  $\Pi_{\tau,w}$ , που ορίστηκε στο Λήμμα 3.3.1, παράγει τον χρωματισμό του δέντρου  $\mathcal{T}_{\mathcal{I},\mathcal{A}}$  που κωδικοποιεί την  $\tau$ -δομή  $\mathcal{A}$  και την ειδική δεντροδιάσπαση  $\mathcal{I}$ . Πιο συγκεκριμένα, το πρόγραμμα  $\Pi_{\tau,w}$  συμπεραίνει τα

$$\text{color}_{1-\gamma_1}(n), \dots, \text{color}_{(\ell+2)-\gamma_{\ell+2}}(n)$$

ανν το  $\gamma = (\gamma_1, \dots, \gamma_{\ell+2})$  είναι το χρώμα του  $n$  στο  $\mathcal{T}_{\mathcal{I},\mathcal{A}}$ . Αυτό είναι το μοναδικό σημείο του  $\Pi_\phi^w$  που εμφανίζεται άρνηση.

Το πρόγραμμα  $P^*$ , πραγματοποιώντας μία bottom-up διάσχιση του δέντρου  $\mathcal{T}_{\mathcal{I},\mathcal{A}}$ , υπολογίζει τις πιθανές καταστάσεις του  $w$ -dec-Assign $_\phi$  σε κάθε κόμβο του δέντρου. Στη συνέχεια, το πρόγραμμα  $S^*$ , πραγματοποιώντας μία top-down διάσχιση του δέντρου  $\mathcal{T}_{\mathcal{I},\mathcal{A}}$  και χρησιμοποιώντας τις πιθανές καταστάσεις, υπολογίζει τις επιτυχημένες καταστάσεις. Πιο συγκεκριμένα, το γεγονός  $\text{succ-}qI(n)$  συμπεραίνεται από το  $S^*$  ανν υπάρχει μία επιτυχημένη εκτέλεση του  $w$ -dec-Assign $_\phi$  πάνω στο  $\mathcal{T}_{\mathcal{I},\mathcal{A}}$  που αναθέτει την κατάσταση  $qI$  στον κόμβο  $n$ , όπου  $I = \emptyset$  ή  $I = \{i\}$  για κάποιο  $i$ .

Ξέρουμε όμως ότι αν κάποια επιτυχημένη εκτέλεση του  $w$ -dec-Assign $_\phi$  πάνω στο  $\mathcal{T}_{\mathcal{I},\mathcal{A}}$  αντιστοιχεί την κατάσταση  $q\{i\}$  στον κόμβο  $n$ , τότε το  $a_n^i$  είναι μία ανάθεση που ικανοποιεί τον τύπο  $\phi$ . Έτσι, με τους κανόνες  $r_{q,i}$  επιλέγουμε κάθε επιτυχημένη κατάσταση  $q\{i\}$  σε κάποιον  $n$  και τη μετατρέπουμε στην αντίστοιχη ανάθεση  $a_n^i$ , υπολογίζοντας τελικά το  $\phi(\mathcal{A})$ .

$$P^*. \quad p_{\gamma q I}^* : \quad \text{pot-}qI(x) \leftarrow \text{leaf}(x), \{ \text{color}_{j-\gamma_j}(x) \mid 1 \leq j \leq \ell + 2 \}$$

$$p_{\gamma q q_1 q_2 I}^* : \quad \text{pot-}qI(x) \leftarrow \text{pot-}q_1 I_1(x_1), \text{pot-}q_2 I_2(x_2), \text{succ}_0(x, x_1), \text{succ}_1(x, x_2), \\ \{ \text{color}_{j-\gamma_j}(x) \mid 1 \leq j \leq \ell + 2 \}$$

$$S^*. \quad s_{q I}^* : \quad \text{succ-}qI(x) \leftarrow \text{pot-}qI(x), \text{root}(x)$$

$$s_{\gamma q q_1 q_2 I, j}^* : \quad \text{succ-}q_j I_j(x_j) \leftarrow \text{succ-}qI(x), \text{pot-}q_1 I_1(x_1), \text{pot-}q_2 I_2(x_2), \text{succ}_0(x, x_1), \text{succ}_1(x, x_2),$$

$$r_{q,i} : \text{assign}(x_i) \leftarrow \text{suc-}q\{i\}(x), \text{decomp-}i(x, x_i)$$

όπου  $q \in F$  στους κανόνες  $s_{qI}^*$  και  $r_{q,i}$ ,  $\gamma qI \in w\text{-}D_0$ ,  $\gamma q q_1 q_2 I \in w\text{-}D$ ,  $\gamma = (\gamma_1, \dots, \gamma_{\ell+2})$ ,  $I, I_1, I_2 \subseteq W$  και  $i \in W$ .

**Πολυπλοκότητα.** Το πρόγραμμα  $\Pi_{\tau,w}$  πραγματοποιεί ακριβώς  $(\ell + 2) \cdot |T|$  στιγμιοτυποποιήσεις κανόνων. Στη συνέχεια, κατά την αποτίμηση του  $P^*$ , έχουμε για κάθε φύλλο ακριβώς  $2^{w+1}$  στιγμιοτυποποιήσεις κανόνων, δηλαδή ίσες με τον αριθμός των διαφορετικών συνόλων  $I \in \mathcal{P}(W)$ , και για κάθε εσωτερικό κόμβο το πολύ  $2^{w+1} \cdot s^2$  στιγμιοτυποποιήσεις κανόνων, δηλαδή ίσες με το  $|\mathcal{P}(W)| \times |Q|^2$ . Ανάλογος είναι και ο αριθμός των στιγμιοτυποποιήσεων κανόνων του  $S^*$ . Επίσης, έχουμε  $a = |\phi(\mathcal{A})|$  στιγμιοτυποποιήσεις κανόνων της οικογένειας  $r_{q,i}$ . Έτσι, συνολικά η πολυπλοκότητα αποτίμησης του προγράμματος  $\Pi_{\phi}^w$  είναι το πολύ  $(\ell + 2 + 2^{w+2} \cdot s^2) \cdot |T| + a$ .  $\square$

Άμεση συνέπεια της Πρότασης 3.3.1 είναι το παρακάτω αποτέλεσμα ορισιμότητας.

**Θεώρημα 3.3.1.** Η κλάση των μονομελών MSO-ορίσιμων ερωτημάτων σε δομές φραγμένου δεντροπλάτους είναι ορίσιμη στη μοναδιαία datalog.

### 3.3.4 Αποτίμηση $k$ -μελών MSO ερωτημάτων μέσω της $(k + 1)$ -datalog

Οι datalog λύσεις που παρουσιάζουμε σε αυτή την ενότητα, όπως και γενικά στη διατριβή, βασίζονται στην απευθείας αναγωγή του αρχικού προβλήματος αποτίμησης της MSO σε ένα πρόβλημα αποτίμησης αυτομάτων για δέντρα που υπολογίζουν άμεσα αναθέσεις. Πιο συγκεκριμένα, τα datalog προγράμματα αυτής της ενότητας υλοποιούν την εξής διαδικασία: συνθέτουν σταδιακά –διασχίζοντας το δέντρο με κατεύθυνση από τα φύλλα προς τη ρίζα– τις επιτυχημένες εκτελέσεις των αυτομάτων αποδόμησης, και ταυτόχρονα αποκωδικοποιούν τα μέρη αναθέσεων αυτών των εκτελέσεων. Το  $\phi(\mathcal{A})$ , που αποτελείται από αυτά τα αποκωδικοποιημένα μέρη αναθέσεων, επιλέγεται τελικά μέσω των κατηγορημάτων assign.

**Αλγόριθμος αυτομάτων.** Ας υποθέσουμε ότι ο  $\phi$  είναι ένας  $\text{MSO}[\tau]$  τύπος με  $k$  ελεύθερες μεταβλητές, τον οποίο θέλουμε να αποτιμήσουμε πάνω σε μία  $\tau$ -δομή  $\mathcal{A}$  με σύμπαν  $A$ . Από το Θεώρημα 3.2.1 έχουμε ότι υπάρχει ένα αυτόματο αποδόμησης  $w\text{-dec-Assign}_{\phi}$  τέτοιο ώστε για κάθε  $\tau$ -δομή  $\mathcal{A}$  και ειδική δεντροδιάσπαση  $\mathcal{I}$  της  $\mathcal{A}$  πλάτους  $w$  να ισχύει

$$\phi(\mathcal{A}) = w\text{-dec-Assign}_{\phi}(\mathcal{T}_{\mathcal{I},\mathcal{A}}).$$

Θεωρούμε ότι το  $a\text{-suc-runs}_{\phi,\mathcal{I},\mathcal{A}}$  είναι το σύνολο που περιέχει κάθε μέρος ανάθεσης που αποτελεί τμήμα μιας επιτυχημένης εκτέλεσης του  $w\text{-dec-Assign}_{\phi}$  πάνω στο  $\mathcal{T}_{\mathcal{I},\mathcal{A}}$ , δηλαδή

$$a\text{-suc-run}_{\phi,\mathcal{I},\mathcal{A}} = \{\varepsilon \mid \text{υπάρχει επιτυχ. εκτέλεση } \rho \text{ του } w\text{-dec-Assign}_{\phi} \text{ στο } \mathcal{T}_{\mathcal{I},\mathcal{A}} \text{ τ.ω. } \rho = \varsigma; \varepsilon\}$$

και ότι το  $\varepsilon_n^i \subseteq W$  συμβολίζει το  $i$ -οστό στοιχείο του  $\varepsilon(n) = (\varepsilon_n^1, \dots, \varepsilon_n^k)$ . Επίσης θυμίζουμε ότι το  $a_n^j$  συμβολίζει το  $j$ -οστό στοιχείο του σάκου  $\bar{a}_n$  της δεντροδιάσπασης  $\mathcal{I}$  και ότι το  $T$  είναι το σύνολο κόμβων του  $\mathcal{T}_{\mathcal{I};A}$ . Συμβολίζουμε με  $B_{\varepsilon,n}^i$  το σύνολο στοιχείων του  $A$  που αντιστοιχούν στην προβολή  $\pi_{\varepsilon_n^i} \bar{a}_n$ , δηλαδή

$$B_{\varepsilon,n}^i = \{a_n^j \mid j \in \varepsilon_n^i\} \text{ για } 1 \leq i \leq k.$$

Τότε, από του Ορισμούς 3.2.2 & 3.2.3, έχουμε ότι

$$\phi(A) = \left\{ \bigcup_{n \in T} (B_{\varepsilon,n}^1, \dots, B_{\varepsilon,n}^k) \mid \varepsilon \in \text{a-suc-runs}_{\phi, \mathcal{I};A} \right\}$$

όπου η ένωση είναι κατά-συνιστώσες ένωση. Η παραπάνω ισότητα αποτελεί τον πυρήνα ενός απλού αλγορίθμου αυτομάτων για τον υπολογισμό του  $\phi(A)$  που περιγράφεται από τα παρακάτω δυο βήματα:

- Υπολόγισε, διασχίζοντας το δέντρο  $\mathcal{T}_{\mathcal{I};A}$  με κατεύθυνση από τα φύλλα προς τη ρίζα, τις αναθέσεις που κωδικοποιούνται στα μέρη αναθέσεων όλων των πιθανών εκτελέσεων του  $w\text{-dec-Assign}_{\phi}$  στο  $\mathcal{T}_{\mathcal{I};A}$ . Πιο συγκεκριμένα, για κάθε τέτοιο μέρος ανάθεσης  $\varepsilon$ , υπολόγισε σε κάθε κόμβο  $n$  το σύνολο

$$\bar{B}_{\varepsilon, T_n} = \bigcup_{n \in T_n} \bar{B}_{\varepsilon, n}$$

όπου  $\bar{B}_{\varepsilon, n} = (B_{\varepsilon, n}^1, \dots, B_{\varepsilon, n}^k)$  και το  $T_n \subseteq T$  είναι το σύνολο κόμβων του υποδέντρου του  $\mathcal{T}_{\mathcal{I};A}$  που έχει ρίζα τον κόμβο  $n$ . Όταν το  $n$  είναι φύλλο, τότε προφανώς  $\bar{B}_{\varepsilon, T_n} = \bar{B}_{\varepsilon, n}$ . Όταν ο  $n$  είναι εσωτερικός κόμβος με παιδιά  $n_1, n_2$ , τότε ο υπολογισμός πραγματοποιείται με τις κατά-συνιστώσες ενώσεις

$$\bar{B}_{\varepsilon, T_n} = \bar{B}_{\varepsilon, n} \cup \bar{B}_{\varepsilon, T_{n_1}} \cup \bar{B}_{\varepsilon, T_{n_2}}. \quad (*)$$

Ας θεωρήσουμε ότι το  $B_n \subseteq A$  αποτελεί την ένωση των σάκων  $a_n$  για κάθε  $n \in T_n$ , δηλαδή  $B_n = \bigcup_{n \in T_n} a_n$ . Το σύνολο  $\bar{B}_{\varepsilon, T_n}$  που υπολογίζεται αντιστοιχεί στον περιορισμό της ανάθεσης  $\bar{B}_{\varepsilon}$  στο  $B_n$ , δηλαδή έχουμε  $\bar{B}_{\varepsilon, T_n} = \bar{B}_{\varepsilon} \cap (B_n)^k$  όπου η τομή πραγματοποιείται κατά συνιστώσες.

- *Επέλεξε μόνο τις αναθέσεις που αντιστοιχούν σε μέρη αναθέσεων εκτελέσεων που είναι επιτυχημένες.* Πιο συγκεκριμένα, η διάκριση αυτή, ανάμεσα στις αναθέσεις που αντιστοιχούν σε επιτυχημένες εκτελέσεις και εκείνες που αντιστοιχούν σε μη επιτυχημένες, πραγματοποιείται μέσω της τιμής του μέρους καταστάσεων στη ρίζα. Είναι προφανές ότι αυτή είναι η μοναδική τιμή του μέρους καταστάσεων  $\varsigma$  μιας εκτέλεσης  $\varsigma; \varepsilon$  που χρειαζόμαστε προκειμένου να αποφανθούμε για το εάν η  $\bar{B}_{\varepsilon}$  είναι μία ανάθεση που ικανοποιεί τον δεδομένο τύπο, καθώς ισχύει ότι η  $\bar{B}_{\varepsilon}$  είναι μία ανάθεση που ικανοποιεί τον δεδομένο τύπο αν  $\varsigma(r) \in F$ .

Στην datalog, το πρώτο από τα παραπάνω δύο βήματα εκφράζεται με τους κανόνες  $r_{\gamma q \bar{I}}$  και  $r_{\gamma q_1 q_2 \bar{I}}$  του προγράμματος  $A$  παρακάτω. Πιο συγκεκριμένα, το γεγονός

$$\text{assign-}q(n, \bar{B})$$

υπολογίζεται από το  $A$  αν υπάρχει μία εκτέλεση  $\varsigma; \varepsilon$  του  $w\text{-dec-Assign}_\phi$  πάνω στο  $\mathcal{T}_{\mathcal{L};A}$  τ.ω.

$$\varsigma(n) = q \ \& \ \overline{B}_\varepsilon|_{B_n} = \overline{B}.$$

Είναι προφανές ότι οι κανόνες  $s_q$  του  $A$  εκφράζουν το δεύτερο βήμα του αλγορίθμου.

$$\begin{aligned} A. \quad r_{\gamma q \bar{I}} &: \text{assign-}q(x, \bar{x}) \leftarrow \text{"color-}\gamma(x)\text{"}, \text{leaf}(x), \text{"decomp-}\bar{I}(x, \bar{x})\text{"} \\ r_{\gamma q q_1 q_2 \bar{I}} &: \text{assign-}q(x, \bar{x}) \leftarrow \text{"color-}\gamma(x)\text{"}, \text{succ}_0(x, u_1), \text{assign-}q_1(u_1, \bar{y}), \text{succ}_1(x, u_2), \\ &\quad \text{assign-}q_2(u_2, \bar{z}), \text{decomp-}\bar{I}(x, \bar{v}), \text{cw-union}(\bar{x}, \bar{v}, \bar{y}, \bar{z}) \\ s_q &: \text{assign}(\bar{x}) \leftarrow \text{assign-}q(x, \bar{x}), \text{root}(x) \end{aligned}$$

όπου  $\gamma q \bar{I} \in w\text{-}D_0$ ,  $\gamma q q_1 q_2 \bar{I} \in w\text{-}D$  και  $q \in F$  στους κανόνες  $s_q$ . Επίσης  $\bar{x} = x_1, \dots, x_k$ ,  $\bar{y} = y_1, \dots, y_k$ ,  $\bar{z} = z_1, \dots, z_k$  και  $\bar{I} = (I_1, \dots, I_k)$ . Το "color- $\gamma(x)$ " αποτελεί συντομογραφία για την ακολουθία γεγονότων

$$\text{color}_{1-\gamma_1}(x), \dots, \text{color}_{(\ell+2)-\gamma_{\ell+2}}(x)$$

και το "decomp- $\bar{I}(x, \bar{x})$ " για την

$$\text{decomp-}I_1(x, x_1), \dots, \text{decomp-}I_k(x, x_k).$$

**Βελτιστοποιημένη datalog λύση.** Η datalog λύση μας για το πρόβλημα αποτίμησης MSO τύπων πάνω σε δομές φραγμένου αποτελεί μία βελτιστοποιημένη εκδοχή του datalog προγράμματος  $A$  που παρουσιάσαμε παραπάνω και είναι το αποτέλεσμα αναγραφής μέσω της μεθόδου φιλτραρίσματος που βασίζεται στην επίλυση, γνωστής στη βιβλιογραφία ως «magic-sets»: οι επιτυχημένες καταστάσεις του αυτομάτου λειτουργούν ως φίλτρα που εισάγονται στο σώμα των κανόνων του  $A$ . Πιο συγκεκριμένα, οι επιτυχημένες καταστάσεις του  $w\text{-dec-Assign}_\phi$  στο  $\mathcal{T}_{\mathcal{L};A}$  φιλτράρουν όλες τις αναθέσεις που δεν ικανοποιούν τον δεδομένο τύπο ήδη από την αρχή των υπολογισμών, γεγονός που δε μπορούσε να πραγματοποιήσει το  $A$  παρά μόνο στον τελικό υπολογισμό στη ρίζα. Ο αναγνώστης μπορεί να ανακαλέσει από το προηγούμενο κεφάλαιο μια καλύτερη ανάλυση της μεθόδου.

Αναλυτικότερα, το φιλτραρισμένο στρώμα  $A_S^*$  του προγράμματος  $\Pi_{\phi, k}^w$ , της πρότασης που ακολουθεί, υπολογίζει –με τη μορφή γεγονότων πάνω στο κατηγορημα στόχο assign– τις αναθέσεις που ικανοποιούν τον  $\phi$ . Ο υπολογισμός γίνεται με την πραγματοποίηση ενώσεων κατά-συνιστώσες  $k$ -πλειάδων από υποσύνολα του  $A$ , διασχίζοντας τους κόμβους του δέντρου με κατεύθυνση από τα φύλλα προς τη ρίζα και υλοποιώντας την παραπάνω ισότητα (\*). Πιο συγκεκριμένα, ως θεωρήσουμε ότι

$$A_n = \{a \in A \mid \text{node}(a) \in T_n\}.$$

Το πρόγραμμα  $A_S^*$  συμπεραίνει το γεγονός

$$\text{assign-}q(n, \overline{B})$$

ανν υπάρχει μία επιτυχημένη εκτέλεση  $\varsigma; \varepsilon$  του  $w\text{-dec-Assign}_\phi$  πάνω στο  $\mathcal{T}_{\mathcal{I}; \mathcal{A}}$  τ.ω.

$$\varsigma(n) = q \ \& \ \overline{B}_\varepsilon|_{A_n} = \overline{B}$$

δηλαδή οι πλειάδες  $\overline{B}$  που υπολογίζονται από το  $A_S^*$  αποτελούν περιορισμούς των αναθέσεων που ικανοποιούν τον  $\phi$  πάνω στο σύνολο  $A_n \subseteq A$ .

Σημειώνουμε ότι το πρόγραμμα που δίνει τη λύση έχει ως είσοδο μία βάση δεδομένων εμπλουτισμένη με κατηγορήματα που εκφράζουν ενώσεις συνόλων. Πιο συγκεκριμένα, η αρχική μας βάση δεδομένων επεκτείνεται με μία βάση δεδομένων πάνω στο δυναμοσύνολο του σύμπαντος  $A$  της δεδομένης δομής  $\mathcal{A}$ . Αυτή η βάση δεδομένων συμβολίζεται με

$$U_A$$

και αποτελείται από γεγονότα που ορίζουν ενώσεις υποσυνόλων του  $A$  που έχουν την παρακάτω μορφή

$$\text{union}(A \cup A_1 \cup A_2, A, A_1, A_2)$$

όπου  $A \subseteq a_n$  και  $A_1 \subseteq B_{n_1}, A_2 \subseteq B_{n_2}, n \in T$ . Τελικά, ορίζουμε τη νέα μας βάση ως εξής

$$\mathcal{D}_{\mathcal{A}, \mathcal{I}}^U = \mathcal{D}_{\mathcal{A}} \cup \mathcal{D}_{\mathcal{I}} \cup U_A.$$

**Πρόταση 3.3.2.** Ας θεωρήσουμε ότι  $\tau = \{R_1, \dots, R_\ell\}$ . Για κάθε MSO $[\tau]$  τύπο  $\phi(X_1, \dots, X_k)$  και θετικό ακέραιο  $w$ , υπάρχει ένα ημιθετικό datalog ερώτημα

$$Q_{\phi, k}^w = (\Pi_{\phi, k}^w, \text{assign})$$

βαθμού  $k + 1$  τέτοιο ώστε για κάθε  $\tau$ -δομή  $\mathcal{A}$  με δεντροπλάτος το πολύ  $w$  και ειδική δεντροδιάσπαση  $\mathcal{I} = (T, (\overline{a}_n)_{n \in T})$  να ισχύει

$$(B_1, \dots, B_k) \in \phi(\mathcal{A}) \ \text{ανν} \ (B_1, \dots, B_k) \in Q_{\phi}^w(\mathcal{D}_{\mathcal{A}, \mathcal{I}}^U).$$

Η πολυπλοκότητα του υπολογισμού είναι το πολύ

$$(\ell + 2 + 2^{k \cdot (w+1)+1} \cdot s^2 + a) \cdot |T| + a$$

όπου  $s$  είναι το μέγεθος του συνόλου  $Q$  των  $\phi$ -καταστάσεων,  $m$  είναι μέγεθος του  $T$  και  $a = |\phi(T)|$ .

**Απόδειξη.** Ορίζουμε το  $\Pi_{\phi, k}^w$  ίσο με την ένωση  $\Pi_{\tau, w} \cup P \cup S \cup A_S^*$ . Το πρόγραμμα  $\Pi_{\tau, w}$  είναι το πρόγραμμα που παράγει τον χρωματισμό του δέντρου  $\mathcal{T}_{\mathcal{I}; \mathcal{A}}$ , που χρησιμοποιείται στη συνέχεια ως είσοδος στο πρόγραμμα  $P \cup S$  που υπολογίζει τις επιτυχημένες καταστάσεις



του  $w$ -dec-Assign $_{\phi}$  πάνω στο  $\mathcal{T}_{\mathcal{I};\mathcal{A}}$ . Οι επιτυχημένες καταστάσεις φιλτράρουν στην συνέχεια τον υπολογισμό των αναθέσεων στο πρόγραμμα  $A_S^*$  με την έννοια της ανάλυσης που δόθηκε παραπάνω.

Στο πρόγραμμα που ορίζεται στη συνέχεια έχουμε ότι  $\gamma = (\gamma_1, \dots, \gamma_{\ell+2})$  και  $\bar{I} = (I_1, \dots, I_k) \in (\mathcal{P}(W))^k$ .

$$\begin{aligned}
 P. \quad p_{\gamma q} &: \quad \text{pot-}q(x) \leftarrow \text{leaf}(x), \{\text{color}i\text{-}\gamma_i(x) \mid 1 \leq i \leq \ell + 2\} \\
 p_{\gamma q q_1 q_2} &: \quad \text{pot-}q(x) \leftarrow \text{pot-}q_1(x_1), \text{pot-}q_2(x_2), \text{succ}_0(x, x_1), \text{succ}_1(x, x_2), \\
 &\quad \{\text{color}i\text{-}\gamma_i(x) \mid 1 \leq i \leq \ell + 2\} \\
 S. \quad s_q &: \quad \text{suc-}q(x) \leftarrow \text{pot-}q(x), \text{root}(x) \\
 s_{\gamma q q_1 q_2}^i &: \quad \text{suc-}q_i(x_i) \leftarrow \text{suc-}q(x), \text{pot-}q_1(x_1), \text{pot-}q_2(x_2), \text{succ}_0(x, x_1), \text{succ}_1(x, x_2), \\
 &\quad \{\text{color}i\text{-}\gamma_i(x) \mid 1 \leq i \leq \ell + 2\} \quad i = 1, 2
 \end{aligned}$$

όπου  $q \in F$  στους κανόνες  $s_q$ ,  $\gamma q \in w\text{-}D'_0$  και  $\gamma q q_1 q_2 \in w\text{-}D'$ .

$$\begin{aligned}
 A_S^*. \quad g_{\gamma q \bar{I}} &: \quad \text{assign-}q(x, x_1, \dots, x_k) \leftarrow \text{suc-}q(x), \{\text{color}i\text{-}\gamma_i(x) \mid 1 \leq i \leq \ell + 2\}, \text{leaf}(x), \\
 &\quad \text{decomp-}I_1(x, x_1), \dots, \text{decomp-}I_k(x, x_k) \\
 g_{\gamma q q_1 q_2 \bar{I}} &: \quad \text{assign-}q(x, x_1, \dots, x_k) \leftarrow \text{suc-}q(x), \{\text{color}i\text{-}\gamma_i(x) \mid 1 \leq i \leq \ell + 2\}, \\
 &\quad \text{succ}_0(x, u_1), \text{assign-}q_1(u_1, y_1, \dots, y_k), \\
 &\quad \text{succ}_1(x, u_2), \text{assign-}q_2(u_2, z_1, \dots, z_k), \\
 &\quad \text{decomp-}I_1(x, v_1), \dots, \text{decomp-}I_k(x, v_k) \\
 &\quad \text{union}(x_1, v_1, y_1, z_1), \dots, \text{union}(x_k, v_k, y_k, z_k) \\
 a_q &: \quad \text{assign}(x_1, \dots, x_k) \leftarrow \text{assign-}q(x, x_1, \dots, x_k), \text{root}(x)
 \end{aligned}$$

όπου  $\gamma q \bar{I} \in w\text{-}D_0$ ,  $\gamma q q_1 q_2 \bar{I} \in w\text{-}D$ ,  $q \in Q$ .

*Πολυπλοκότητα.* Αρχικά, το πρόγραμμα  $\Pi_{\tau, w}$  πραγματοποιεί ακριβώς  $(\ell + 2) \cdot |T|$  στιγμιοτυποποιήσεις κανόνων. Στη συνέχεια, κάθε ένα από τα  $P$  και  $S$ , πραγματοποιεί σε κάθε κόμβο  $n$  το πολύ  $2^{k \cdot (w+1)} \cdot s^2$  στιγμιοτυποποιήσεις κανόνων, δηλαδή τόσες όσο το  $(|\mathcal{P}(W)|)^k \times |Q|^2$ . Στο τελευταίο βήμα, κατά την αποτίμηση του  $A_S^*$  έχουμε  $|\phi(\mathcal{A})|$  στιγμιοτυποποιήσεις κανόνων σε κάθε κόμβο, ενώ κατά την αποτίμηση των κανόνων  $a_q$  πραγματοποιούνται επιπλέον  $|\phi(\mathcal{A})|$  στιγμιοτυποποιήσεις στη ρίζα. Έτσι, συνολικά η πολυπλοκότητα αποτίμησης του προγράμματος  $\Pi_{\phi, k}^w$  είναι το πολύ  $(\ell + 2 + 2^{k \cdot (w+1)} \cdot s^2 + a) \cdot |T| + a$ .  $\square$

Άμεση συνέπεια της Πρότασης 3.3.2 είναι το αποτέλεσμα ορισιμότητας που δίνεται στη συνέχεια και γενικεύει το αντίστοιχο που δίνεται στο Θεώρημα 2.4.1 για την περίπτωση των δέντρων.

**Θεώρημα 3.3.2.** Η κλάση των  $k$ -μελών MSO-ορίσιμων ερωτημάτων πάνω σε σχεσιακές δομές φραγμένου δεντροπλάτους είναι ορίσιμη στην  $(k + 1)$ -datalog.

### 3.4 MSO-ορίσιμα προβλήματα βελτιστοποίησης & οι datalog λύσεις τους

Στην ενότητα αυτή παρουσιάζουμε datalog λύσεις για τα MSO-ορίσιμα προβλήματα βελτιστοποίησης.

#### 3.4.1 Το $k$ -card-MSO πρόβλημα ελέγχου μοντέλου

Το  $k$ -card-MSO πρόβλημα ελέγχου μοντέλου, για έναν MSO τύπο  $\phi(X)$  και μία δομή  $\mathcal{A}$ , συνίσταται στο να αποφανθούμε εάν υπάρχει ένα σύνολο  $X$  μεγέθους το πολύ  $k$  που να ικανοποιεί τον  $\phi$  στην  $\mathcal{A}$ . Δηλαδή μπορεί να οριστεί ως το πρόβλημα ελέγχου μοντέλου για έναν « $k$ -card-MSO-τύπο  $\psi^{\exists \leq k} = \exists^{\leq k} X \phi(X)$ » ο οποίος αληθεύει σε μία δομή αν και μόνο αν η απάντηση στο  $k$ -card-MSO πρόβλημα ελέγχου μοντέλου για τον  $\phi(X)$  στην ίδια δομή είναι «ναι». Επισημαίνουμε ότι ο παραπάνω φραγμαλισμός του προβλήματος ως πρόβλημα ελέγχου μοντέλου για κάποιο «τύπο» αποτελεί κατάχρηση του συμβολισμού, ενώ το ίδιο ισχύει και για τις αντίστοιχες τυποποιήσεις στις υπόλοιπες περιπτώσεις MSO-ορίσιμων προβλημάτων.

Στην ενότητα αυτή ανάγουμε το  $k$ -card-MSO πρόβλημα ελέγχου μοντέλου σε ένα πρόβλημα αποτίμησης για τη μοναδιαία datalog. Το datalog πρόγραμμα που επιλύει αυτό το πρόβλημα αποτίμησης πραγματοποιεί μία μοναδική διάσχιση πάνω στη δεδομένη επεκτεταμένη δεντροδιάσπαση, υπολογίζοντας τις πιθανές καταστάσεις του αυτομάτου με μέτρηση  $w$ -dec-Assign $_{\phi}^{\text{count}}$  για τον  $\phi(X)$ . Ένα τέτοιο πρόγραμμα, που παρουσιάζει βασικές ομοιότητες με τα προγράμματα των προηγούμενων ενοτήτων που υπολογίζουν τις πιθανές καταστάσεις του  $w$ -dec-Assign $_{\phi}$ , έχει την παρακάτω μορφή:

$$P^c. \quad \begin{aligned} p_{\gamma qi}^c &: \text{pot-}qi(x) \leftarrow \text{leaf}(x), \text{"color-}\gamma(x)\text{"} \\ p_{\gamma q_1 q_2 i_1 i_2}^c &: \text{pot-}qj(x) \leftarrow \text{pot-}q_1 i_1(x_1), \text{pot-}q_2 i_2(x_2), \text{succ}_0(x, x_1), \text{succ}_1(x, x_2), \\ &\quad \text{"color-}\gamma(x)\text{"} \quad \text{για } j = i + i_1 + i_2 \end{aligned}$$

όπου  $\gamma qi \in w-D_0^c$ ,  $\gamma q_1 q_2 i \in w-D^c$  και  $i_1, i_2$  θετικοί ακέραιοι.

Η μοναδική διαφορά εδώ είναι ότι παράλληλα με τον υπολογισμό των πιθανών καταστάσεων  $q$ , υπολογίζουμε το μέγεθος  $i$  των αναθέσεων που αντιστοιχούν στις εκτελέσεις με αυτές τις καταστάσεις. Πιο συγκεκριμένα, το datalog πρόγραμμα  $P^c$ , που ορίστηκε παραπάνω, παράγει το ζευγάρι  $(q, i)$  σε έναν κόμβο  $n$  ανν υπάρχει μία εκτέλεση  $\varsigma; \varepsilon; c$  του  $w$ -dec-Assign $_{\phi}^{\text{count}}$  στο  $\mathcal{T}_{\mathcal{I}; \mathcal{A}}$  τέτοια ώστε  $\varsigma(n) = q$  και  $c(n) = i$ . Θυμίζουμε ότι το τελευταίο ισχύει ανν υπάρχει μία εκτέλεση  $\varsigma; \varepsilon$  του  $w$ -dec-Assign $_{\phi}$  στο  $\mathcal{T}_{\mathcal{I}; \mathcal{A}}$  τέτοια ώστε

$$\varsigma(n) = q \quad \& \quad i = \sum_{n \in T_n} |\varepsilon(n)|.$$

Γνωρίζουμε όμως ότι όταν η  $\varsigma; \varepsilon$  είναι μία επιτυχημένη εκτέλεση, τότε το  $q$  είναι μία επιτυχημένη κατάσταση και το  $i$  είναι το μέγεθος του περιορισμού στο  $A_n$  της ανάθεσης  $B_{\varepsilon}$  που ικανοποιεί

τον δεδομένο τύπο. Συνεπώς, ο υπολογισμός ενός ζευγαριού  $(q, i)$ ,  $q \in F$  στη ρίζα  $r$ , δηλαδή ο συμπερασμός του γεγονότος  $\text{rot}^k\text{-}qi(r)$ , αποδεικνύει την ύπαρξη μιας ανάθεσης που ικανοποιεί τον  $\phi(X)$  και έχει μέγεθος  $i$ . Έτσι, ο τύπος  $\psi^{\exists \leq k}$  αληθεύει ανν έχουμε την παραγωγή ενός ζευγαριού  $(q, i)$  στη ρίζα για το οποίο ισχύουν  $q \in F$  και  $i \leq k$ . Στη λύση που δίνουμε στην πρόταση που ακολουθεί, χρησιμοποιούμε αυτή την τελευταία απαίτηση  $i \leq k$  ώστε να εφαρμόσουμε μια μορφή φιλτραρίσματος που βασίζεται στην επίλυση και που έχει ως αποτέλεσμα την απαλοιφή στιγμιοτυποποιήσεων που παράγουν «άχρηστα» γεγονότα. Παρέχουμε έτσι μία βελτιστοποιημένη datalog λύση.

**Πρόταση 3.4.1.** Ας υποθέσουμε ότι  $\tau = \{R_1, \dots, R_\ell\}$ , ο  $\phi(X)$  είναι ένας  $\text{MSO}[\tau]$  τύπος και το  $k$  ένας θετικός ακέραιος. Για κάθε τύπο  $\psi^{\exists \leq k} = \exists^{\leq k} X \phi(X)$  και θετικό ακέραιο  $w$ , υπάρχει ένα μοναδιαίο ημιθετικό datalog ερώτημα

$$Q_\psi^{\exists \leq k, w} = (\Pi_\psi^{\exists \leq k, w}, \psi\text{-true})$$

που αποτιμά τον  $\psi^{\exists \leq k}$  σε κάθε  $\tau$ -δομή με δεντροπλάτος το πολύ  $w$ . Πιο συγκεκριμένα, για κάθε  $\tau$ -δομή  $\mathcal{A}$  με ειδική δεντροδιάσπαση  $\mathcal{I} = (\mathcal{T}, (\bar{a}_n)_{n \in T})$  πλάτους το πολύ  $w$  έχουμε

$$\mathcal{A} \models \psi^{\exists \leq k} \text{ ανν } \Pi_\psi^{\exists \leq k, w}(\mathcal{D}_\mathcal{A} \cup \mathcal{D}_\mathcal{I}) \models \psi\text{-true}.$$

Η πολυπλοκότητα του υπολογισμού είναι το πολύ  $(\ell + 2 + 2^{w+1} \cdot s^2) \cdot |T|$ .

**Απόδειξη.** Ορίζουμε το πρόγραμμα  $\Pi_\phi^{\exists \leq k, w}$  μέσω της ένωσης  $\Pi_{\tau, w} \cup P^{\exists \leq k}$ . Το πρώτο πρόγραμμα  $\Pi_{\tau, w}$  είναι το ημιθετικό datalog πρόγραμμα που ορίστηκε στο Λήμμα 3.3.1 και παράγει το χρωματισμό του δέντρου  $\mathcal{T}_{\mathcal{I}; \mathcal{A}}$  που κωδικοποιεί την  $\tau$ -δομή  $\mathcal{A}$  και την ειδική δεντροδιάσπαση  $\mathcal{I}$ . Το δεύτερο datalog πρόγραμμα  $P^{\exists \leq k}$  ορίζεται παρακάτω.

$$\begin{aligned} P^{\exists \leq k}. \quad & p_{\gamma qi}^k : \text{rot}^k\text{-}qi(x) \leftarrow \text{leaf}(x), \text{"color-}\gamma(x)\text{"} \\ & p_{\gamma q q_1 q_2 i, i_1 i_2}^k : \text{rot}^k\text{-}qj(x) \leftarrow \text{rot}^k\text{-}q_1 i_1(x_1), \text{rot}^k\text{-}q_2 i_2(x_2), \text{succ}_0(x, x_1), \text{succ}_1(x, x_2), \\ & \quad \quad \quad \text{"color-}\gamma(x)\text{"} \quad \quad \quad \text{για } j = i + i_1 + i_2 \\ & a_{qi}^k : \quad \quad \psi\text{-true} \leftarrow \text{rot}^k\text{-}qi(x), \text{root}(x) \end{aligned}$$

όπου οι κανόνες  $p_{\gamma qi}^k$  ορίζονται για κάθε  $\gamma qi$  τ.ω.  $\gamma qi \in w\text{-}D_0^c$  &  $i \leq k$  και οι κανόνες  $p_{\gamma q q_1 q_2 i, i_1 i_2}^k$  ορίζονται για  $\gamma q q_1 q_2 i, i_1 i_2$  τ.ω.  $\gamma q q_1 q_2 i \in w\text{-}\Delta^c$  &  $i + i_1 + i_2 \leq k$  όπου τα  $i_1, i_2$  είναι θετικοί ακέραιοι. Οι κανόνες  $a_{qi}^k$  ορίζονται για κάθε  $q \in F$  και  $i \leq k$ . Το "color- $\gamma(x)$ " αποτελεί συντομογραφία για την ακολουθία των γεγονότων  $\{\text{color}j\text{-}\gamma_j(x) \mid 1 \leq j \leq \ell + 2\}$ .

Η αποτίμηση του  $P^{\exists \leq k}$  ξεκινάει από τα φύλλα του δέντρου της δεδομένη δομής και συνεχίζει διασχίζοντας όλους τους κόμβους μέχρι τη ρίζα. Πιο συγκεκριμένα, οι κανόνες  $p_{\gamma qi}^k$  και  $p_{\gamma q q_1 q_2 i, i_1 i_2}^k$  παράγουν το γεγονός  $\text{rot}^k\text{-}qi(n)$  ανν υπάρχει μία εκτέλεση  $\varsigma; \varepsilon$  του  $w\text{-dec-Assign}_\phi$  στο  $\mathcal{T}_{\mathcal{I}; \mathcal{A}}$  τέτοια ώστε

$$\varsigma(n) = q \quad \& \quad i = \sum_{n \in T_n} |\varepsilon(n)| \leq k.$$

Μέσω του κανόνα  $a_{qi}^k$  ελέγχουμε αν το γεγονός  $\text{rot}^k\text{-}qi(r)$ ,  $q \in F$ , συμπεραίνεται από το πρόγραμμα στη ρίζα  $r$  του δέντρου της δομής. Καθώς όλα τα γεγονότα που συμπεραίνονται από το πρόγραμμα ικανοποιούν τη συνθήκη  $i \leq k$ , ο συμπερασμός ενός τέτοιου γεγονότος αποδεικνύει την ύπαρξη μιας επιτυχημένης εκτέλεσης του  $w\text{-dec-Assign}_\phi$  η οποία υπολογίζει μία ανάθεση που ικανοποιεί τον  $\phi(X)$  μεγέθους το πολύ  $k$ . Συνεπώς αποδεικνύει ότι ο  $\psi^{\exists \leq k} = \exists^{\leq k} X \phi(X)$  αληθεύει στη δομή  $\mathcal{A}$ .

Επισημαίνουμε ότι το πρόγραμμα  $P^{\exists \leq k}$  αντιστοιχεί σε έναν βελτιστοποιημένο υπολογισμό καθώς ορίζει τα κατηγορήματα  $\text{rot}^k\text{-}qi$  μόνο για εκείνα τα  $i$  για τα οποία ικανοποιείται η συνθήκη  $i \leq k$ . Η μη-βελτιστοποιημένη εκδοχή του είναι το αρχικό πρόγραμμα  $P^c$  που ορίζει κατηγορήματα χωρίς περιορισμό στα  $i$  ως προς το  $k$ . Πιο συγκεκριμένα, το πρόγραμμα  $P^{\exists \leq k}$  αντιστοιχεί στο αποτέλεσμα την αναγραφής του μη-βελτιστοποιημένου προγράμματος μέσω της εισαγωγής του φίλτρου « $j \leq k$ » στο σώμα των κανόνων του. Δηλαδή τα δύο αυτά προγράμματα παράγουν τα ίδια γεγονότα πάνω στο κατηγορήματα στόχος με είσοδο την ίδια βάση δεδομένων. Πράγματι, το φίλτρο αυτό είναι ένα ορθό φίλτρο καθώς τα ζευγάρια  $(p, j)$  που συμμετέχουν στην παραγωγή του ζευγαριού  $(q, i)$ ,  $i \leq k$  στη ρίζα πρέπει και τα ίδια να ικανοποιούν τη συνθήκη  $j \leq k$ . Η απαίτηση αυτή είναι προφανής καθώς κάθε τέτοιο  $i$  είναι το άθροισμα των αντίστοιχων  $j$  που υπολογίζονται κατά την διάσχιση του δέντρου πριν καταλήξουμε στη ρίζα. Συνεπώς η παραγωγή των ζευγαριών  $(p, j)$  με  $j > k$  είναι παντελώς περιττή για την απάντηση στο δεδομένο ερώτημα.

*Πολυπλοκότητα.* Το πρόγραμμα  $\Pi_{\tau, w}$  πραγματοποιεί ακριβώς  $(\ell + 2) \cdot |T|$  στιγμιοτυποποιήσεις κανόνων. Στη συνέχεια, κατά την αποτίμηση του  $P^{\exists \leq k}$ , έχουμε σε κάθε φύλλο το πολύ  $2^{w+1}$  στιγμιοτυποποιήσεις, δηλαδή όσες και τα διαφορετικά σύνολα  $I \in \mathcal{P}(W)$ , και για κάθε εσωτερικό κόμβο το πολύ  $2^{w+1} \cdot s^2$  στιγμιοτυποποιήσεις, δηλαδή ίσες με  $|\mathcal{P}(W)| \cdot |Q|^2$ . Συνεπώς, η πολυπλοκότητα υπολογισμού του προγράμματος  $\Pi_{\phi}^{\exists k, w}$  είναι το πολύ  $(\ell + 2 + 2^{w+1} \cdot s^2) \cdot |T|$ .  $\square$

Αξίζει να σημειωθεί ότι ένα βασικό πλεονέκτημα της βελτιστοποιημένης λύσης που δόθηκε στην παραπάνω πρόταση είναι η γενικότητά της, δηλαδή το γεγονός ότι επιλύει το πρόβλημα για οποιαδήποτε δομή ανεξάρτητα από το μέγεθος της. Η μη-βελτιστοποιημένη εκδοχή της λύσης έχει κανόνες που ορίζουν κατηγορήματα  $\text{rot}\text{-}qi$  για κάθε  $i \leq m$  όπου το  $m$  αποτελεί φράγμα που καθορίζεται από το μέγεθος  $|A|$  του σύμπαντος της δεδομένης δομής  $\mathcal{A}$ . Στην περίπτωση αυτή, η datalog λύση, έχοντας εξάρτηση από κάποιο θετικό ακέραιο  $m$ , αφορά μόνο σε δομές μεγέθους το πολύ  $m$ .

### 3.4.2 Το min-card-MSO (min-card-MSO) πρόβλημα αποτίμησης

Δεδομένου ενός MSO-τύπου  $\phi(X)$ , το min-card-MSO πρόβλημα αποτίμησης για έναν «min-card-MSO-τύπο»  $\text{card}\text{-}\phi^{\min}$  (αντίστοιχα «max-card-MSO-τύπο»  $\text{card}\text{-}\phi^{\max}$ ) συνίσταται στον υπολογισμό του πληθάριθμου του συνόλου των αναθέσεων που ικανοποιούν τον  $\phi(X)$  και έχουν ελάχιστο (αντίστοιχα μέγιστο) μέγεθος. Η επίλυση αυτού του προβλήματος αποτίμησης είναι ακριβώς ανάλογη με αυτή που δόθηκε στην προηγούμενη ενότητα και αφορούσε τον  $\exists^{\leq k} X \phi(X)$ . Δηλαδή, βασίζεται στον υπολογισμό των πιθανών καταστάσεων του αυτομάτου

με μέτρηση για τον  $\phi(X)$  πάνω στο δέντρο της δεδομένης δομής και στην κατάλληλη επεξεργασία των τιμών που αντιστοιχούν στη ρίζα του δέντρου αυτού. Πιο συγκεκριμένα, μία προφανής datalog λύση που αποτιμά τον  $\text{card-}\phi^{\min}$  πάνω σε δομές μεγέθους το πολύ  $m$  προκύπτει άμεσα από την επέκταση του προγράμματος  $P^c$ , που αντιστοιχεί στη μη-βελτιστοποιημένη λύση της προηγούμενης ενότητας, με το διαστρωματωμένο  $\text{datalog}^\neg$  πρόγραμμα  $A_m^{c,\min}$  που ορίζεται παρακάτω.

$$A_m^{c,\min}. \quad \begin{array}{ll} s_{qi} : & \text{suc-}i \leftarrow \text{pot-}qi(x), \text{root}(x) \\ a_i : & \text{min-}\phi\text{-}i \leftarrow \text{suc-}i, \{\neg\text{suc-}j \mid j \in \{1, \dots, i-1\}\} \end{array}$$

όπου  $q \in F$  και  $i$  θετικός ακέραιος τ.ω.  $i \leq m$ .

Μετά από τον υπολογισμό των πιθανών καταστάσεων μέσω του προγράμματος  $P^c$ , οι κανόνες  $s_{qi}$  συλλέγουν στα κατηγορήματα  $\text{suc-}i$  τα μεγέθη  $i$  των συνόλων που αποτελούν αναθέσεις που ικανοποιούν τον  $\phi$  στη δεδομένη δομή. Είναι προφανές ότι αυτά αντιστοιχούν πλήρως στα ζευγάρια  $(q, i)$  που είναι μέρος των επιτυχημένων καταστάσεων του αυτομάτου με μέτρηση στη ρίζα. Στο τελευταίο βήμα, οι κανόνες  $a_i$  υπολογίζουν το ελάχιστο στοιχείο του συνόλου που αποτελείται από τα μεγέθη των αναθέσεων που ικανοποιούν τον  $\phi$ , παρέχοντας έτσι τη λύση στο αρχικό πρόβλημα αποτίμησης. Η περίπτωση του τύπου  $\text{card-}\phi^{\max}$  είναι πλήρως συμμετρική.

Όπως κάναμε και στην προηγούμενη ενότητα, εφαρμόζουμε στην παραπάνω μη-βελτιστοποιημένη datalog λύση τις ιδέες του φιλτραρίσματος που βασίζεται στην επίλυση, και δίνουμε μία βελτιστοποιημένη datalog λύση για το αρχικό πρόβλημα αποτίμησης. Ο ορισμός του φίλτρου που χρησιμοποιούμε βασίζεται στο λήμμα που ακολουθεί, το οποίο αποτελεί και απόδειξη της ορθότητάς του.

Για έναν τύπο  $\phi(X)$ , συμβολίζουμε με  $S_q(n)$  το σύνολο των επιτυχημένων καταστάσεων ενός αυτομάτου με μέτρηση, για τον  $\phi$  στον κόμβο  $n$ , που έχουν τη μορφή  $(q, I, i)$  για κάποια  $I, i$ . Για  $S_q(n) \neq \emptyset$ , ορίζουμε το σύνολο

$$\min_q(n) = \min\{i \mid \exists I((q, I, i) \in S_q(n))\}.$$

**Λήμμα 3.4.1.** Μία επιτυχημένη κατάσταση  $(q, I, i) \in S_q(n)$  τέτοια ώστε  $i > \min_q(n)$  δεν ανήκει στο σύνολο των επιτυχημένων καταστάσεων στον κόμβο  $n$  που συμμετέχουν στον υπολογισμό μιας ανάθεσης ελάχιστου μεγέθους που ικανοποιεί τον τύπο  $\phi$ .

**Απόδειξη.** Η απόδειξη βασίζεται στο γεγονός ότι, οποιεσδήποτε δύο εκτελέσεις ενός αυτομάτου στο υποδέντρο  $\mathcal{T}_n$  οι οποίες έχουν κοινό μέρος καταστάσεων στον κόμβο  $n$ , επεκτείνονται κατά τον ίδιο ακριβώς τρόπο σε ολόκληρο το δέντρο  $\mathcal{T}$ .

Πιο συγκεκριμένα, ας υποθέσουμε ότι έχουμε μια κατάσταση  $(q, I, i) \in S_q(n)$  για την οποία ισχύει ότι  $i > \min_q(n)$ . Αρκεί να αποδείξουμε ότι για κάθε επιτυχημένη εκτέλεση  $\rho = \varsigma; \varepsilon; c$  με  $\rho(n) = (q, I, i)$ , υπάρχει μία επιτυχημένη εκτέλεση  $\rho' = \varsigma'; \varepsilon'; c'$  τέτοια ώστε

$$|B_{\varepsilon'}| < |B_\varepsilon| \quad (*)$$

Πράγματι, μία τέτοια εκτέλεση  $\rho'$  μπορεί να κατασκευαστεί εύκολα. Ας θεωρήσουμε μία κατάσταση  $(q, J, j) \in S_q(n)$  με  $j = \min_q(n)$ . Από τον ορισμό του  $S_q(n)$ , υπάρχει μία επιτυχημένη εκτέλεση  $\rho'' = \zeta''; \varepsilon''; c''$  τέτοια ώστε  $\rho''(n) = (q, J, j)$  και  $|B_{\varepsilon''} \cap A_n| = j$ . Πράγματι, η εκτέλεση  $\rho' = \zeta'; \varepsilon'; c'$  που ορίζεται ως εξής

$$\zeta'; \varepsilon'|_{T_n} = \zeta''; \varepsilon''|_{T_n} \quad \& \quad \zeta'; \varepsilon'|_{\bar{T}_n} = \zeta; \varepsilon|_{\bar{T}_n}$$

είναι μία επιτυχημένη εκτέλεση για την οποία ισχύει

$$\begin{aligned} |B_{\varepsilon'}| &= |B_{\varepsilon''} \cap A_n| + |B_{\varepsilon} \cap \bar{A}_n| = j + |B_{\varepsilon} \cap \bar{A}_n| \\ &< i + |B_{\varepsilon} \cap \bar{A}_n| = |B_{\varepsilon}| \end{aligned}$$

δηλαδή ικανοποιεί την (\*). □

α. Βελτιστοποιημένη επίλυση μέσω της μοναδιαίας  $\text{datalog}^\neg$  για δομές μεγέθους το πολύ  $m$ . Παρουσιάζουμε αρχικά μία λύση για το  $\text{min-card-MSO}$  πρόβλημα αποτίμησης που έχει εξάρτηση από το μέγεθος της δομής εισόδου.

**Πρόταση 3.4.2.** Ας υποθέσουμε ότι  $\tau = \{R_1, \dots, R_\ell\}$ . Για κάθε τύπο  $\text{card-}\phi^{\min}$  και θετικούς ακέραιους  $w, m$  υπάρχει ένα πρόγραμμα  $\Pi_{c \min \phi}^{w, m}$  της μοναδιαίας  $\text{datalog}^\neg$  τέτοιο ώστε για κάθε  $\tau$ -δομή  $\mathcal{A}$  μεγέθους το πολύ  $m$  με ειδική δεντροδιάσπαση  $\mathcal{I} = (\mathcal{T}, (\bar{a}_n)_{n \in T})$  πλάτους  $w$  να ισχύει

$$\text{το } i \text{ ικανοποιεί τον } \text{card-}\phi^{\min} \text{ στην } \mathcal{A} \quad \text{ανν} \quad \Pi_{c \min \phi}^{w, m}(\mathcal{D}_{\mathcal{A}} \cup \mathcal{D}_{\mathcal{I}}) \models \phi\text{-min-}i.$$

Η πολυπλοκότητα του υπολογισμού είναι το πολύ  $(\ell + 2 + 2^{w+1} \cdot s^2 + 2 \cdot s) \cdot |T|$ .

**Απόδειξη.** Ορίζουμε το  $\Pi_{c \min \phi}^{w, m}$  ως την ένωση  $\Pi_{\tau, w} \cup P_m^{c, \min}$ . Το ημιθετικό  $\text{datalog}$  πρόγραμμα  $\Pi_{\tau, w}$  είναι αυτό που ορίστηκε στο Λήμμα 3.3.1, και παράγει τη συνάρτηση χρωματισμού του δέντρου  $\mathcal{T}_{\mathcal{I}; \mathcal{A}}$  που κωδικοποιεί την  $\tau$ -δομή  $\mathcal{A}$  και την ειδική δεντροδιάσπαση  $\mathcal{I}$ . Το  $\text{datalog}^\neg$  πρόγραμμα  $P_m^{c, \min}$  ορίζεται ως εξής

$$\begin{aligned} P_m^{c, \min}. \quad p_{\gamma qi} &: \quad \text{pot-}qi(x) \leftarrow \text{leaf}(x), \text{ "color-}\gamma(x)\text{"} \\ m_{qi, w} &: \quad \text{min-pot-}qi(x) \leftarrow \text{leaf}(x), \text{pot-}qi(x), \{\neg \text{pot-}qj(x) \mid j \in \{1, \dots, i-1\}\} \\ p_{\gamma q_1 q_2 i, i_1 i_2} &: \quad \text{pot-}qj(x) \leftarrow \text{min-pot-}q_1 i_1(x_1), \text{min-pot-}q_2 i_2(x_2), \text{succ}_0(x, x_1), \\ &\quad \text{succ}_1(x, x_2), \text{ "color-}\gamma(x)\text{"} \quad \text{για } j = i + i_1 + i_2 \\ m_{qi} &: \quad \text{min-pot-}qi(x) \leftarrow \text{succ}_0(x, y), \text{pot-}qi(x), \{\neg \text{pot-}qj(x) \mid j \in \{1, \dots, i-1\}\} \\ s_{qi} &: \quad \text{min-suc-}i \leftarrow \text{min-pot-}qi(x), \text{root}(x) \\ a_i &: \quad \text{min-}\phi\text{-}i \leftarrow \text{min-suc-}i, \{\neg \text{min-suc-}j \mid j \in \{1, \dots, i-1\}\} \end{aligned}$$

όπου οι κανόνες  $p_{\gamma qi}$  ορίζονται για κάθε  $\gamma qi \in w-D_0^c$ , οι κανόνες  $p_{\gamma q_1 q_2 i, i_1 i_2}$  ορίζονται για  $\gamma q_1 q_2 i, i_1 i_2$  τ.ω.  $\gamma q_1 q_2 i \in w-D^c$  και  $i_1, i_2$  θετικούς ακέραιους, οι κανόνες  $m_{qi, w}$  ορίζονται για κάθε  $q \in Q$  και  $i \leq w + 1$ , οι κανόνες  $m_{qi, w}$  ορίζονται για κάθε  $q \in Q$  και  $i \leq m$ , και οι κανόνες  $s_{qi}$  ορίζονται για κάθε  $q \in F$  και  $i \leq m$ , ενώ οι  $a_i$  για  $i \leq m$ . Το "color- $\gamma(x)$ " αποτελεί συντομογραφία για την ακολουθία γεγονότων που ανήκουν στο  $\{\text{color-}\gamma_j(x) \mid 1 \leq j \leq \ell + 2\}$ .  $\square$

**β. Βελτιστοποιημένη datalog<sup>min</sup> λύση για κάθε δομή ανεξάρτητα από το μέγεθός της.** Μπορούμε να έχουμε μία γενική datalog λύση, δηλαδή για όλες τις δομές φραγμένου δεντροπλάτους, επεκτείνοντας την  $\text{datalog}^\square$  με τη συναθροιστική συνάρτηση min. Ονομάζουμε  $\text{datalog}^{\text{min}}$  αυτή την επέκταση. Επιπρόσθετα, επεκτείνουμε τη βάση δεδομένων που αποτελεί είσοδο στο αντίστοιχο πρόγραμμα με άτομα που περιγράφουν αριθμητικούς τελεστές πάνω σε θετικούς ακέραιους και πιο συγκεκριμένα με άτομα που ορίζουν το άθροισμα τριών θετικών ακεραίων αριθμών. Για δεδομένο θετικό ακέραιο  $m$ , ορίζουμε την βάση δεδομένων  $S_m$  που ορίζει τέτοια αθροίσματα ανάμεσα σε θετικούς ακέραιους μικρότερους ή ίσους του  $m$  ως εξής:

$$\text{sum}(k, l, i, j) \in S_m \text{ ανν } k = l + i + j.$$

Για κάθε δομή  $\mathcal{A}$  με σύμπαν  $A$  μεγέθους  $m = |A|$ , ορίζουμε τη βάση δεδομένων

$$\mathcal{D}_{\mathcal{A}}^{\text{sum}} = \mathcal{D}_{\mathcal{A}} \cup S_m.$$

**Πρόταση 3.4.3.** Ας υποθέσουμε ότι  $\tau = \{R_1, \dots, R_\ell\}$  και ότι το  $w$  είναι ένας θετικός ακέραιος. Για κάθε τύπο  $\text{card-}\phi^{\text{min}}$ , υπάρχει ένα  $\text{datalog}^{\text{min}}$  ερώτημα  $Q_{\text{cmin}\phi}^w = (\Pi_{\text{cmin}\phi}^w, \text{min-}\phi)$  βαθμού 2 τέτοιο ώστε για κάθε  $\tau$ -δομή  $\mathcal{A}$  με ειδική δεντροδιάσπαση  $\mathcal{I} = (\mathcal{T}, (\bar{a}_n)_{n \in T})$  πλάτους  $w$  να ισχύει

$$\text{το } i \text{ ικανοποιεί τον } \text{card-}\phi^{\text{min}} \text{ στην } \mathcal{A} \text{ ανν } i \in Q_{\text{cmin}\phi}^w(\mathcal{D}_{\mathcal{A}}^{\text{sum}} \cup \mathcal{D}_{\mathcal{I}}).$$

Η πολυπλοκότητα του υπολογισμού είναι το πολύ  $(\ell + 2 + 2^{w+1} \cdot s^2 + 2 \cdot s) \cdot |T|$ .

**Απόδειξη.** Ορίζουμε το  $\Pi_{\text{cmin}\phi}^w$  μέσω της ένωσης προγραμμάτων  $\Pi_{\tau, w} \cup P^{c, \text{min}}$ . Θυμίζουμε ότι το ημιθετικό datalog πρόγραμμα  $\Pi_{\tau, w}$ , που ορίστηκε στο Λήμμα 3.3.1, παράγει τη συνάρτηση χρωματισμού του δέντρου  $\mathcal{T}_{\mathcal{I}; \mathcal{A}}$  που αποτελεί κωδικοποίηση της  $\tau$ -δομής  $\mathcal{A}$  και της ειδικής δεντροδιάσπασης  $\mathcal{I}$ . Το  $\text{datalog}^{\text{min}}$  πρόγραμμα  $P^{c, \text{min}}$  ορίζεται παρακάτω.

$$\begin{array}{ll} P^{c, \text{min}}. & p_{\gamma qi} : \quad \text{pot-}q(x, i) \leftarrow \text{leaf}(x), \text{"color-}\gamma(x)\text{"} \\ & m_q : \quad \text{min-pot-}q(x, \text{min}(i)) \leftarrow \text{pot-}q(x, i) \\ & p_{\gamma q_1 q_2 i} : \quad \text{pot-}q(x, j) \leftarrow \text{min-pot-}q_1(x_1, i_1), \text{min-pot-}q_2(x_2, i_2), \text{succ}_0(x, x_1), \\ & \quad \quad \quad \text{succ}_1(x, x_2), \text{sum}(j, i, i_1, i_2), \text{"color-}\gamma(x)\text{"} \\ & s_q : \quad \text{min-suc}(i) \leftarrow \text{min-pot-}q(x, i), \text{root}(x) \\ & a : \quad \text{min-}\phi(\text{min}(i)) \leftarrow \text{min-suc}(i) \end{array}$$

όπου οι κανόνες  $p_{\gamma qi}$  ορίζονται για κάθε  $\gamma qi \in w\text{-}\Delta_0^c$ , οι κανόνες  $p_{\gamma q_1 q_2 i}$  ορίζονται για κάθε  $\gamma q_1 q_2 i \in w\text{-}\Delta^c$ , οι κανόνες  $m_q$  ορίζονται για κάθε  $q \in Q$ , και οι κανόνες  $s_q$  ορίζονται για κάθε  $q \in F$ . Το "color- $\gamma(x)$ " αποτελεί συντομογραφία για την ακολουθία γεγονότων του  $\{\text{color}j\text{-}\gamma_j(x) \mid 1 \leq j \leq \ell + 2\}$ .  $\square$

### 3.4.3 Το min-MSO (max-MSO) πρόβλημα αποτίμησης

Το min-MSO (αντίστοιχα max-MSO) πρόβλημα αποτίμησης ορίζεται ως το πρόβλημα αποτίμησης ενός min-MSO-τύπου  $\phi^{\min}(X)$  (αντίστοιχα max-MSO-τύπου  $\phi^{\max}(X)$ ) ο οποίος ικανοποιείται μόνο από εκείνες τις αναθέσεις που ικανοποιούν τον MSO τύπο  $\phi(X)$  που έχουν επιπρόσθετα ελάχιστο (αντίστοιχα μέγιστο) μέγεθος.

Η λύση προϋποθέτει μία επιπλέον επέκταση της βάσης δεδομένων που αποτελεί είσοδο σε αυτά τα datalog προγράμματα. Πιο συγκεκριμένα, εμπλουτίζουμε την αρχική βάση δεδομένων με κατηγορήματα που υπολογίζουν τον πληθάρημο των υποσυνόλων του σύμπαντος της δεδομένης δομής. Για κάθε πεπερασμένο σύνολο  $A$  ορίζουμε τη βάση δεδομένων  $C_A$  ως εξής:

$$\text{card}(B, |B|) \in C_A \text{ ανν } B \subseteq A.$$

Ας υποθέσουμε ότι το  $A$  είναι το σύμπαν της  $\mathcal{A}$  και ότι  $m = |A|$ . Ορίζουμε τη βάση δεδομένων

$$\mathcal{D}_{\mathcal{A};\mathcal{I}}^{\text{ext}} = \mathcal{D}_{\mathcal{A};\mathcal{I}}^U \cup S_m \cup C_A.$$

**Πρόταση 3.4.4.** Ας υποθέσουμε ότι  $\tau = \{R_1, \dots, R_\ell\}$  και ότι το  $w$  είναι ένα θετικός ακέραιος. Για κάθε τύπο  $\phi^{\min}(X)$ , υπάρχει ένα datalog<sup>min</sup> ερώτημα  $Q_{\min \phi}^w = (\Pi_{\min \phi}^w, \text{min-assign-}\phi)$ , βαθμού 2, τέτοιο ώστε για κάθε  $\tau$ -δομή  $\mathcal{A}$  με ειδική δεντροδιάσπαση  $\mathcal{I} = (\mathcal{T}, (\bar{a}_n)_{n \in T})$  μεγέθους  $w$  να ισχύει

$$\text{το } B \text{ ικανοποιεί τον } \phi^{\min}(X) \text{ στην } \mathcal{A} \text{ ανν } B \in Q_{\min \phi}^w(\mathcal{D}_{\mathcal{A};\mathcal{I}}^{\text{ext}}).$$

Η πολυπλοκότητα του υπολογισμού είναι το πολύ  $(\ell + 2 + 2^{w+2} \cdot s^2 + 2 \cdot s + a) \cdot |T| + a$ .

**Απόδειξη.** Ορίζουμε το datalog πρόγραμμα  $\Pi_{\min \phi}^w$  μέσω της παρακάτω ένωσης προγραμμάτων

$$\Pi_{\min \phi}^w = \Pi_{\tau, w} \cup P^{c, \min} \cup S^{\min} \cup A_S^{\min}.$$

Θυμίζουμε ότι το ημιθετικό datalog πρόγραμμα  $\Pi_{\tau, w}$ , που ορίστηκε στο Λήμμα 3.3.1, παράγει τη συνάρτηση χρωματισμού του δέντρου  $\mathcal{T}_{\mathcal{I};\mathcal{A}}$  που αποτελεί κωδικοποίηση της  $\tau$ -δομής  $\mathcal{A}$  και της ειδικής δεντροδιάσπασης  $\mathcal{I}$ . Το datalog<sup>min</sup> πρόγραμμα  $P^{c, \min}$ , που ορίστηκε στην προηγούμενη ενότητα, παράγει το γεγονός  $\text{min-}\phi(i)$  ανν το  $i$  είναι το μέγεθος των αναθέσεων που ικανοποιούν τον  $\phi(X)$  και έχουν ελάχιστο μέγεθος. Το datalog πρόγραμμα  $S^{\min} \cup A_S^{\min}$  ορίζεται παρακάτω.

$$S^{\min}. \quad s_q : \quad \text{min-suc-}q(x, i) \leftarrow \text{min-}\phi(i), \text{min-pot-}q(x, i), \text{root}(x)$$



$$\begin{aligned}
 s_{\gamma q q_1 q_2 i} &: \text{min-suc-}q_\ell(x_\ell, i_\ell) \leftarrow \text{min-suc-}q(x, j), \text{min-pot-}q_1(x_1, i_1), \text{min-pot-}q_2(x_2, i_2), \\
 &\quad \text{succ}_0(x, x_1), \text{succ}_1(x, x_2), \text{sum}(j, i, i_1, i_2), \\
 &\quad \text{"color-}\gamma(x)\text{"} \quad \text{για } \ell = 1, 2 \\
 A_S^{\text{min}}. \quad a_{\gamma q I} &: \text{assign-}q(x, y) \leftarrow \text{min-suc-}q(x, i), \text{leaf}(x), \text{"color-}\gamma(x)\text{"} \\
 &\quad \text{decomp-}I(x, y), \text{card}(y, i) \\
 a_{\gamma q q_1 q_2 I} &: \text{assign-}q(x, y) \leftarrow \text{min-suc-}q(x, i), \text{"color-}\gamma(x)\text{"}, \text{succ}_0(x, u_1), \text{succ}_1(x, u_2), \\
 &\quad \text{assign-}q_1(u_1, w), \text{assign-}q_2(u_2, z), \\
 &\quad \text{decomp-}I(x, v), \text{union}(y, v, w, z), \text{card}(y, i) \\
 a_q &: \text{min-assign-}\phi(y) \leftarrow \text{assign-}q(x, y), \text{root}(x)
 \end{aligned}$$

όπου οι κανόνες  $s_{\gamma q}$  και  $a_q$  ορίζονται για κάθε  $q \in F$ , οι κανόνες  $s_{\gamma q q_1 q_2 i}$  ορίζονται για κάθε  $\gamma q q_1 q_2 i \in w\text{-}\Delta^c$ , και οι κανόνες  $a_{\gamma q I}$ ,  $a_{\gamma q q_1 q_2 I}$  ορίζονται για κάθε  $\gamma q I \in w\text{-}D_0$ ,  $\gamma q q_1 q_2 I \in w\text{-}D$  αντίστοιχα. Το "color- $\gamma(x)$ " αποτελεί συντομογραφία για την ακολουθία γεγονότων του  $\{\text{color}j\text{-}\gamma_j(x) \mid 1 \leq j \leq \ell + 2\}$ .  $\square$

**Πόρισμα 3.4.1.** Τα min-MSO-ορίσματα ερωτήματα πάνω σε δομές φραγμένου δεντροπλάτους είναι ορίσματα στη  $2\text{-data}\log^{\text{min}}$ .

Παραθέτουμε παραδείγματα επίλυσης μέσω αυτής της προσέγγισης στις ενότητες που ακολουθούν.

### 3.5 Μία περίπτωση μελέτης: υπολογισμός καλύμματος κορυφών

Ας θεωρήσουμε ένα γράφο  $\mathcal{G}$  με σύνολο κορυφών  $V$ . Ένα σύνολο  $A \subseteq V$  είναι *κάλυμμα κορυφών* όταν κάθε ακμή του  $\mathcal{G}$  έχει τουλάχιστον ένα άκρο στο  $A$ . Ένα ελάχιστο κάλυμμα κορυφών είναι ένα κάλυμμα κορυφών με ελάχιστο αριθμό κορυφών, δηλαδή με το μικρότερο δυνατό μέγεθος. Το πρόβλημα υπολογισμού ενός ελάχιστου καλύμματος κορυφών είναι ένα κλασσικό NP-hard πρόβλημα βελτιστοποίησης της επιστήμης υπολογιστών, ενώ το αντίστοιχο πρόβλημα απόφασης είναι επίσης πολύ γνωστό NP-πλήρες πρόβλημα της θεωρίας υπολογιστικής πολυπλοκότητας.

Στην ενότητα αυτή εφαρμόζουμε τις μεθόδους επίλυσης που αναπτύχθηκαν στις προηγούμενες ενότητες, τόσο αυτές που βασίζονται στα αυτόματα όσο και τις αντίστοιχες datalog λύσεις, για να αποτιμήσουμε τον  $\text{MSO}[\tau_{\mathcal{G}}]$  τύπο

$$\phi(Z) = \forall x \forall y (E(x, y) \rightarrow \text{In}(x, Z) \vee \text{In}(y, Z))$$

που εκφράζει ότι το σύνολο  $Z$  είναι ένα κάλυμμα κορυφών του γράφου  $\mathcal{G}$ .

### 3.5.1 Η κατασκευή του αυτομάτου αποδόμησης που υπολογίζει καλύμματα κορυφών

Αρχικά εφαρμόζουμε τις κατασκευές που δόθηκαν στην απόδειξη του κεντρικού θεωρήματος, Θεώρημα 3.2.1, για την παρακάτω ισοδύναμη μορφή του τύπου που περιγράφει το κάλυμμα κορυφών

$$\phi(Z) \equiv \neg \exists x \exists y (E(x, y) \wedge \neg \ln(x, Z) \wedge \neg \ln(y, Z)).$$

Τα βασικά σημεία αυτής της επαγωγικής διαδικασίας συνοψίζονται στον Πίνακα 3.7 και στον Πίνακα 3.8. Ουσιαστικά, τα αυτόματα που παραθέτουμε αποτελούν αποτέλεσμα της εφαρμογής των ιδεών του επαγωγικού ορισμού αυτομάτων, σε συμπαγείς αναπαραστάσεις αυτομάτων και όχι ως προς την αναλυτική τους μορφή που προϋποθέτει την πλήρη παράθεση των αντίστοιχων μεταβάσεων. Επιπρόσθετα, παράλληλα με τις αντίστοιχες κατασκευές στα διάφορα στάδια, πραγματοποιούμε μία *άτυπη ελαχιστοποίηση αυτομάτων* που βασίζεται στην ταύτιση καταστάσεων με κριτήριο τη σημασία τους. Οι περιπτώσεις αυτές επεξηγούνται αναλυτικότερα στις παρακάτω περιγραφές των *βασικών βημάτων κατασκευής του τελικού αυτομάτου για τον  $\phi(Z)$* .

- Βήμα 1. Άμεση κατασκευή αυτομάτου για τον υποτύπο

$$S(x, y, Z) \equiv E(x, y) \wedge \neg \ln(x, Z) \wedge \neg \ln(y, Z)$$

από το αυτόματο του ατομικού τύπου  $E(x, y)$ . Το αυτόματο για τον  $S(x, y, Z)$  προκύπτει ως αποτέλεσμα μιας κατάλληλης επέκτασης του αυτομάτου του ατομικού τύπου  $E(x, y)$  που περιγράφει τη σχέση ακμών, και δόθηκε στον Πίνακα 3.5 της σελίδας 103. Περιγράφουμε στη συνέχεια τις λεπτομέρειες αυτής της κατασκευής. Και τα δύο αυτόματα, δηλαδή τόσο το αρχικό του  $E(x, y)$  όσο και το νέο που αντιστοιχεί στον  $S(x, y, Z)$ , έχουν το ίδιο σύνολο καταστάσεων και το ίδιο σύνολο τελικών καταστάσεων. Αναλυτικότερα, οι μεταβάσεις του αυτομάτου  $\mathcal{A}_S$  του  $S(x, y, Z)$  προκύπτουν από τις μεταβάσεις του αυτομάτου  $\mathcal{A}_E$  του  $E(x, y)$  σύμφωνα με τις παρακάτω βασικές ιδέες.

- Κάθε μετάβαση  $q_1, q_2, \gamma, I_x I_y \mapsto q$  του  $\mathcal{A}_E$ , που δεν συνεπάγεται αποτυχία αποδοχής, επεκτείνεται i) σε μία ίδιου τύπου μετάβαση  $q_1, q_2, \gamma, I_x I_y I_Z \mapsto q$  του  $\mathcal{A}_S$ , για κάθε υποσύνολο  $I_Z \subseteq W$ , με στοιχεία θέσεις στοιχείων ως προς το  $\gamma$ , για το οποίο ισχύει ότι όταν  $I_x = \{i\}$  (αντίστοιχα  $I_y = \{j\}$ ) τότε έχουμε  $i \notin I_Z$  (αντίστοιχα  $j \notin I_Z$ ), και ii) σε μία μετάβαση μη-αποδοχής  $q_1, q_2, \gamma, I_x I_y I_Z \mapsto q_f$ , για κάθε σύνολο  $I_Z \subseteq W$  το οποίο δεν ικανοποιεί την αντίστοιχη συνθήκη του (i) παραπάνω.
- Κάθε μετάβαση μη-αποδοχής  $q_1, q_2, \gamma, I_x I_y \mapsto q_f$  του  $\mathcal{A}_E$ , επεκτείνεται σε μία μετάβαση μη-αποδοχής  $q_1, q_2, \gamma, I_x I_y I_Z \mapsto q$  του  $\mathcal{A}_S$  για κάθε σύνολο  $I_Z \subseteq W$ .

Η εφαρμογή αυτών των βασικών κατευθύνσεων μας δίνει το αυτόματο του Πίνακα 3.7 (α'). Οι συνθήκες  $vertex(i)$ ,  $AdgInPar(i, j)$ ,  $Edge(i, j)$  που χρησιμοποιούνται στους αντίστοιχους

Πίνακας 3.6: Συνθήκες/Σύνολα που εμφανίζονται στον ορισμό των μεταβάσεων.

Συνθήκες/Σύνολα	ως προς $\gamma \in \Gamma_{\tau,w}$	ως προς τα $\mathcal{I}, \mathcal{G}$
$vertex(j)$	ανν $j \leq \gamma_1$ & $\forall i \in W ((j, i) \notin \gamma_2)$	ανν $j \in Pos_{\mathcal{I}}(n)$
$V_\gamma$	:= $\{i \mid vertex(i)\}$	:= $Pos_{\mathcal{I}}(n)$
$set(I)$	ανν $I \subseteq V_\gamma$	ανν $I \subseteq Pos_{\mathcal{I}}(n)$
$SameInPar(J)$	:= $\{i \mid \exists j \in J ((j, i) \in \gamma_2)\}$	:= $\{i \mid \exists j \in J (a_n^j = a_p^i)\}$
$SameInPar(J) = I$	ανν $I = \{i \mid \exists j \in J ((j, i) \in \gamma_2)\}$	ανν $\{a_n^j \mid j \in J\} \cap a_p = \{a_p^i \mid i \in I\}$
$Adj(j)$	:= $\{i \mid (j, i) \in \gamma_3\}$	:= $\{i \mid (a_n^j, a_n^i) \in E^{\mathcal{G}}\}$
$AdjInPar(j)$	:= $SameInPar(R(j))$	
$Edge(j, \ell)$	ανν $vertex(j)$ & $vertex(\ell)$ & $(j, \ell) \in \gamma_3$ & $(\ell, j) \in \gamma_3$	ανν $j \in Pos_{\mathcal{I}}(n)$ & $\ell \in Pos_{\mathcal{I}}(n)$ & $(a_n^j, a_n^\ell) \in E^{\mathcal{G}}$
$AdjInPar_{\gamma \setminus J}$	:= $\bigcup_{j \in V_\gamma \setminus J} AdjInPar(j)$	
$Edge_{\gamma \setminus J}$	ανν $\exists j, \ell \notin J (Edge(j, \ell))$	
$NotEdge_{\gamma \setminus J}$	ανν $\forall j, \ell \in V_\gamma \setminus J \neg (Edge(j, \ell))$	

ορισμούς μεταβάσεων, αντιστοιχούν ακριβώς στις συνθήκες  $element(i)$ ,  $RInPar(i, j)$ ,  $R(i, j)$  που ορίστηκαν στον Πίνακα 3.2 της σελίδας 101. Ο λόγος που εδώ τις μετονομάζουμε είναι έτσι ώστε η νέα ονομασία να είναι πλήρως συμβατή με την ερμηνεία τους στο πλαίσιο των γράφων. Οι αναλυτικοί ορισμοί των συνθηκών που χρησιμοποιούνται δίνονται στον Πίνακα 3.6.

Αξίζει να σημειωθεί στο σημείο αυτό ότι η παραπάνω ιδέα μιας άμεσης κατασκευής αυτομάτων μπορεί να επεκταθεί φυσικά και σε άλλες ανάλογες περιπτώσεις τύπων. Πιο συγκεκριμένα, μία ανάλογη άμεση κατασκευή αυτομάτου μπορεί να πραγματοποιηθεί κάθε φορά που έχουμε συζεύξεις ενός ατομικού τύπου  $R(x_1, \dots, x_n)$  με άτομα της μορφής  $\ln(x_i, Y)$  και  $\neg \ln(x_j, Z)$ , δηλαδή στην περίπτωση που θέλουμε να κατασκευάσουμε το αυτόματο ενός τύπου της μορφής

$$\psi(x_1, \dots, x_n, Y_1, \dots, Y_m) = R(x_1, \dots, x_n) \wedge \bigwedge_{i=1}^m \bigwedge_{j=1}^n * \ln(x_j, Y_i),$$

όπου το  $* \ln$  είναι είτε  $\ln$  ή  $\neg \ln$ , δεδομένου του αυτομάτου  $\mathcal{A}_\phi$  του τύπου  $R(x_1, \dots, x_n)$ .

- Βήμα 2. Προβολή & Ντετερμινισμός. Η κατασκευή του ντετερμινιστικού αυτομάτου  $w\text{-dec-}\mathcal{A}_\psi$  για τον τύπο

$$\psi(Z) \equiv \exists x \exists y S(x, y, Z)$$

πραγματοποιείται σε δύο φάσεις. Ακολουθώντας τον επαγωγικό ορισμό αυτομάτων, προβάλλουμε αρχικά πάνω στο  $w\text{-dec-}A_S$  και παίρνουμε το μη-ντετερμινιστικό αυτόματο  $w\text{-dec-}A_\psi^{nd}$ , που δίνεται στον Πίνακα 3.7 (β'). Επισημαίνουμε ότι ταυτόχρονα εφαρμόζουμε ένα είδος «ελαχιστοποίησης μέσω νοήματος». Πιο συγκεκριμένα, ενοποιούμε τις καταστάσεις  $q_1^I$  και  $q_2^I$  σε μία κατάσταση  $q^I$  καθώς η ερμηνεία τους ως προς το  $w\text{-dec-}A_\psi^{nd}$  είναι ισοδύναμη. Αρκεί να παρατηρήσουμε ότι η  $q_1^I$  (αντίστοιχα η  $q_2^I$ ) δηλώνει ότι μία κορυφή έχει επιλεγεί στην πρώτη (αντίστοιχα δεύτερη) θέση και το  $I$  είναι οι θέσεις στο σάκο του πατέρα όπου εμφανίζονται οι κορυφές με τις οποίες η επιλεγείσα κορυφή ενώνεται με ακμή. Δηλαδή, η πληροφορία για τη θέση της επιλεγείσας κορυφής, παρόλο που είναι απόλυτα αναγκαία για τον τύπο  $S$ , είναι αδιάφορη για την περίπτωση του  $\psi$ .

Κατά τη δεύτερη φάση, εφαρμόζουμε την κατασκευή δυναμοσυνόλου στο αυτόματο  $w\text{-dec-}A_\psi^{nd}$  και παράγουμε το ντετερμινιστικό αυτόματο  $w\text{-dec-}A_\psi$  που δίνεται στον Πίνακα 3.8. Όπως και προηγουμένως, εφαρμόζουμε παράλληλα ένα είδος «ελαχιστοποίησης μέσω νοήματος». Παρατηρούμε αρχικά ότι οι καταστάσεις του αυτομάτου που προκύπτει άμεσα από την εφαρμογή της κατασκευής δυναμοσυνόλου θα έπρεπε να έχουν τη μορφή υποσυνόλων του  $\{q^I \mid I \subseteq W\}$ . Ας θεωρήσουμε τώρα ότι  $\Sigma \subseteq \mathcal{P}(W)$ . Στο αυτόματο  $w\text{-dec-}A_\psi$  που κατασκευάζουμε, κάθε κατάσταση της μορφής

$$Q^\Sigma = \{q^I \mid I \in \Sigma\}$$

έχει αντικατασταθεί από μία νέα κατάσταση

$$Q^{\cup\Sigma}, \quad \text{όπου } \cup\Sigma = \cup_{I \in \Sigma} I.$$

Επισημαίνουμε ότι με αυτό τον τρόπο, έχουμε ενοποιήσει όλες τις καταστάσεις  $Q^{\Sigma_i}$  και  $Q^{\Sigma_j}$  για τις οποίες ισχύει  $\cup\Sigma_i = \cup\Sigma_j$ .

- Βήμα 3. Άρνηση. Στο τελευταίο βήμα, παράγουμε το αυτόματο για τον

$$\phi_{VC}(Z) \equiv \neg\psi(Z).$$

Στην απόδειξη του Θεωρήματος 3.2.1 είδαμε ότι το συμπλήρωμα του  $w\text{-dec-}A_\psi$  δεν αποτελεί ορθή κατασκευή για το αυτόματο του  $\neg\psi(Z)$ , αλλά απαιτείται επιπρόσθετα η τομή του πρώτου με το αυτόματο  $w\text{-dec-}A_{Set^1}$  ώστε να προκύψει ένα ορθό αυτόματο για τον  $\neg\psi(Z)$ . Εφαρμόζουμε εδώ μία ισοδύναμη διαδικασία η οποία βασίζεται σε βασικές παρατηρήσεις που αφορούν στη μορφή του συμπληρώματος του αρχικού αυτομάτου. Παρατηρούμε λοιπόν ότι οι «προβληματικές» μεταβάσεις του συμπληρώματος του  $w\text{-dec-}A_\psi$  αποτελούν τμήμα των  $\psi$ -μεταβάσεων που συνιστούν αποτυχία αποδοχής και οι οποίες μετατρέπονται σε  $\phi$ -μεταβάσεις που οδηγούν σε τελική κατάσταση. Πιο συγκεκριμένα, οι προβληματικές αυτές  $\psi$ -μεταβάσεις αποτυχίας αποδοχής είναι οι μεταβάσεις της μορφής  $q_1, q_2, \gamma, I \mapsto q_f$  για τις οποίες το  $\gamma$  δεν ικανοποιεί τη συνθήκη  $Set(I)$ . Έτσι, αρκεί να μετατρέψουμε αυτές τις  $\psi$ -μεταβάσεις αποτυχίας αποδοχής που δεν ικανοποιούν τη συνθήκη  $Set(I)$ , σε  $\phi$ -μεταβάσεις αποτυχίας αποδοχής. Για τη συγκεκριμένη περίπτωση που μελετάμε, αυτές οι μεταβάσεις είναι όλες οι μεταβάσεις  $q_1, q_2, \gamma, I \mapsto Q_f$ . Επιπρόσθετα, οι  $\phi$ -μεταβάσεις αποτυχίας αποδοχής είναι όλες οι

$\psi$ -μεταβάσεις  $q_1, q_2, \gamma, I \mapsto Q_a$ . Το αυτόματο που προκύπτει με αυτή τη διαδικασία δίνεται στον Πίνακα 3.10.

Η ερμηνεία των καταστάσεων του  $w$ -dec-Assign $_{\psi_{VC}(Z)}$  δίνεται στον Πίνακα 3.9, όπου θεωρούμε ότι οι καταστάσεις αποδίδονται στον κόμβο  $n$  και ότι ο  $p$  είναι ο πατέρας του  $n$ . Επίσης θεωρούμε ότι

$$V_n = \{a \in V \mid \text{node}(a) \in T_n\}$$

και ορίζουμε το  $Z_n \subseteq V_n$  ως το σύνολο των κορυφών που έχουν επιλεγεί από το αυτόματο στο  $T_n$ .

**Παράδειγμα 3.5.1.** Παραθέτουμε στον Πίνακα 3.11 τις επιτυχημένες εκτελέσεις του αυτομάτου με μέτρηση για το κάλυμμα κορυφών πάνω στο δέντρο που αντιστοιχεί στο γράφο και τη δεντροδιάσπαση του Σχήματος 3.1. Επισημαίνουμε ότι για λόγους απλοποίησης του συμβολισμού γράφουμε τις καταστάσεις στη μορφή  $(I, q, i)$ .

Δίνουμε αρχικά κάποιες από τις συνθήκες που ικανοποιούνται σε κάθε κόμβο και στις οποίες βασίζεται ο υπολογισμός των εκτελέσεων.

- $n_1: V_\gamma = \{1, 2, 3\}, \text{NotEdge}_{\gamma \setminus \{2\}}, \text{NotEdge}_{\gamma \setminus \{1,2\}}, \text{NotEdge}_{\gamma \setminus \{1,3\}}, \text{NotEdge}_{\gamma \setminus \{2,3\}}, \text{NotEdge}_{\gamma \setminus \{1,2,3\}}$
- $n_2: V_\gamma = \{3\}, \text{NotEdge}_{\gamma \setminus \emptyset}, \text{NotEdge}_{\gamma \setminus \{3\}}, \text{AdjInPar}_{\gamma \setminus \emptyset} = \{1, 3\}$
- $n_3: V_\gamma = \{2\}, \text{NotEdge}_{\gamma \setminus \emptyset}, \text{NotEdge}_{\gamma \setminus \{2\}}, \text{AdjInPar}_{\gamma \setminus \emptyset} = \{2\}$
- $n_4: V_\gamma = \{2\}, \text{NotEdge}_{\gamma \setminus \emptyset}, \text{NotEdge}_{\gamma \setminus \{2\}}, \text{AdjInPar}_{\gamma \setminus \emptyset} = \{1, 3\}$
- $n_5: V_\gamma = \{2, 3\}, \text{NotEdge}_{\gamma \setminus \{2\}}, \text{NotEdge}_{\gamma \setminus \{3\}}, \text{NotEdge}_{\gamma \setminus \{2,3\}}, \text{Adj}(3) = \{1, 2\}, \text{Adj}(2) = \{1, 3\}, \text{AdjInPar}_{\gamma \setminus \{2\}} = \text{SameInPar}(\text{Adj}(3)) = \{3\}, \text{AdjInPar}_{\gamma \setminus \{3\}} = \{3\}.$

### 3.5.2 Επίλυση μέσω της datalog

Τα datalog προγράμματα που παρουσιάστηκαν στις προηγούμενες ενότητες και περιγράφουν τον γενικό τρόπο επίλυσης των προβλημάτων που μελετάμε μέσω της datalog, προϋποθέτουν την πλήρη παράθεση των μεταβάσεων του αντίστοιχου αυτομάτου. Θυμίζουμε ότι στα προγράμματα αυτά έχουμε έναν κανόνα  $r_{t_\gamma}$  για κάθε μετάβαση  $t_\gamma$  που ορίζεται ως προς το χρώμα  $\gamma$  του δέντρου  $\mathcal{T}_{I,A}$ . Αντίθετα, τα datalog προγράμματα που παρουσιάζονται σε αυτή την ενότητα, ορίζονται με βάση τη συμπαγή αναπαράσταση των μεταβάσεων του αυτομάτου για το κάλυμμα κορυφών. Πιο συγκεκριμένα, αντί να υπολογίσουμε τον χρωματισμό του  $\mathcal{T}_{I,A}$  μέσω του Λήμματος 3.3.1, γράφουμε νέους datalog κανόνες που παράγουν τις συνθήκες που εμφανίζονται στους συμπαγείς ορισμούς των μεταβάσεων. Επισημαίνουμε ότι οι συνθήκες αυτές ορίζονται ως προς τους σάκους της δεδομένης δεντροδιάσπασης και διαμερίζουν τα χρώματα του  $\mathcal{T}_{I,A}$ . Έτσι, παρόλο που και στις δύο περιπτώσεις ο αριθμός των στιγμιοτυποποιήσεων

Πίνακας 3.7: Τα επαγωγικά βήματα της κατασκευής του αυτομάτου για το κάλυμμα κορυφών.

(α') Ένα αυτόματο για τον  $S(x, y, Z) \equiv \text{Edge}(x, y) \wedge \neg \text{In}(x, Z) \wedge \neg \text{In}(y, Z)$ .

$q_1, q_2, (\gamma, \bar{e}) \mapsto q, j, \ell \in W, I, J, L \subseteq W$	
Μετάβαση	Συνθήκες που ικανοποιούνται
$q_0, q_0, \gamma, (\emptyset, \emptyset, L) \mapsto q_0$	$set(L)$
$q_0, q_0, \gamma, (\{j\}, \emptyset, L) \mapsto q_1^I$	$vertex(j) \ \& \ AdjInPar(j) = I \neq \emptyset \ \& \ set(L) \ \& \ j \notin L$
$q_0, q_0, \gamma, (\emptyset, \{j\}, L) \mapsto q_2^I$	
$q_0, q_0, \gamma, (\{j\}, \{l\}, L) \mapsto q_a$	$Edge(j, l) \ \& \ set(L) \ \& \ j, l \notin L$
$q_0, q_1^J, \gamma, (\emptyset, \emptyset, L) \mapsto q_1^I$	$SameInPar(J) = I \neq \emptyset \ \& \ set(L)$
$q_1^J, q_0, \gamma, (\emptyset, \emptyset, L) \mapsto q_2^I$	
$q_0, q_1^I, \gamma, (\emptyset, \{j\}, L) \mapsto q_a$	$vertex(j) \ \& \ j \in I \ \& \ set(L) \ \& \ j \notin L$
$q_1^I, q_0, \gamma, (\emptyset, \{j\}, L) \mapsto q_a$	
$q_0, q_2^I, \gamma, (\{j\}, \emptyset, L) \mapsto q_a$	$vertex(j) \ \& \ j \in I \ \& \ set(L) \ \& \ j \notin L$
$q_2^I, q_0, \gamma, (\{j\}, \emptyset, L) \mapsto q_a$	
$q_0, q_a, \gamma, (\emptyset, \emptyset, L) \mapsto q_a$	$set(L)$
$q_a, q_0, \gamma, (\emptyset, \emptyset, L) \mapsto q_a$	
$q_1, q_2, \gamma, \bar{I} \mapsto q_f$	όλες οι άλλες περιπτώσεις

(β') Ένα μη-ντετερμινιστικό αυτόματο για τον  $\psi(Z) \equiv \exists xy S(x, y, Z)$ .

$q_1, q_2, (\gamma, \bar{e}) \mapsto q, j, \ell \in W, I, J, L \subseteq W$	
Μετάβαση	Συνθήκες που ικανοποιούνται
$q_0, q_0, \gamma, L \mapsto q_0$	$set(L)$
$q_0, q_0, \gamma, L \mapsto q^I$	$vertex(j) \ \& \ AdjInPar(j) = I \neq \emptyset \ \& \ set(L) \ \& \ j \notin L$
$q_0, q_0, \gamma, L \mapsto q_a$	
$q_0, q^J, \gamma, L \mapsto q^I$	$SameInPar(J) = I \neq \emptyset \ \& \ set(L)$
$q^J, q_0, \gamma, L \mapsto q^I$	
$q_0, q^I, \gamma, L \mapsto q_a$	$vertex(j) \ \& \ j \in I \ \& \ set(L) \ \& \ j \notin L$
$q^I, q_0, \gamma, L \mapsto q_a$	
$q_0, q_a, \gamma, L \mapsto q_a$	$set(L)$
$q_a, q_0, \gamma, L \mapsto q_a$	
$q_1, q_2, \gamma, I \mapsto q_f$	όλες οι άλλες περιπτώσεις

Πίνακας 3.8: Ένα ντετερμινιστικό αυτόματο για τον  $\psi(Z) \equiv \exists xyS(x, y, Z)$ .

$q_1, q_2, (\gamma, \bar{e}) \mapsto q, I, S, L \subseteq W$	
Μετάβαση	Συνθήκες που ικανοποιούνται
$Q^\emptyset, Q^\emptyset, \gamma, L \mapsto Q_a$	$set(L) \ \& \ Edge_{\gamma \setminus L}$
$Q^\emptyset, Q^\emptyset, \gamma, L \mapsto Q^I$	$set(L) \ \& \ NotEdge_{\gamma \setminus L} \ \& \ AdjInPar_{\gamma \setminus L} = I$
$Q^J, Q^S, \gamma, L \mapsto Q_a$	$set(L) \ \& \ Edge_{\gamma \setminus L}$
$Q^J, Q^S, \gamma, L \mapsto Q_a$	$set(L) \ \& \ NotEdge_{\gamma \setminus L} \ \& \ (J \cup S) \cap (V_\gamma \setminus L) \neq \emptyset$
$Q^J, Q^S, \gamma, L \mapsto Q^I$	$set(L) \ \& \ NotEdge_{\gamma \setminus L} \ \& \ (J \cup S) \cap (V_\gamma \setminus L) = \emptyset \ \& \ SameInPar(J \cup S) \cup AdjInPar_{\gamma \setminus L} = I$
$Q^J, Q_a, \gamma, L \mapsto Q_a$ $Q_a, Q^J, \gamma, L \mapsto Q_a$	$set(L)$
$q_1, q_2, \gamma, I \mapsto Q_f$	όλες οι άλλες περιπτώσεις

Πίνακας 3.9: Η ερμηνεία των καταστάσεων του  $w\text{-dec-Assign}_{\psi_{VC}(Z)}$ .

Κατάσταση	Ερμηνεία
$Q^I$	υπάρχει ένα σύνολο $A \subseteq V_n \setminus Z_n$ που αποτελείται από τις κορυφές του $V_n$ που δεν έχουν επιλεγεί από την τρέχουσα εκτέλεση πάνω στο $T_n$ , και το σύνολο $\{a_p^i \mid i \in I\}$ αποτελείται από εκείνες τις κορυφές του $V \setminus V_n$ που συνδέονται με ακμή με τις κορυφές του $A$
$Q_f$	αποτυχία αποδοχής: μία ανάθεση που δεν ικανοποιεί τον τύπο έχει επιλεγεί

Πίνακας 3.10: Ένα ντετερμινιστικό αυτόματο για τον τύπο  $\phi_{VC}(Z)$ .

$q_1, q_2, (\gamma, \bar{e}) \mapsto q, I, S, L \subseteq W; F = \{Q^\emptyset\}$	
Μετάβαση	Συνθήκες που ικανοποιούνται
$\gamma, L \mapsto Q^I$	$set(L) \ \& \ NotEdge_{\gamma \setminus L} \ \& \ AdjInPar_{\gamma \setminus L} = I$
$Q^J, Q^S, \gamma, L \mapsto Q^I$	$set(L) \ \& \ NotEdge_{\gamma \setminus L} \ \& \ (J \cup S) \cap (V_\gamma \setminus L) = \emptyset \ \& \ SameInPar(J \cup S) \cup AdjInPar_{\gamma \setminus L} = I$
$q_1, q_2, \gamma, I \mapsto Q_f$	όλες οι άλλες περιπτώσεις

του βασικού κορμού των αντίστοιχων datalog προγραμμάτων —αντιστοιχεί στις αναθέσεις που υπολογίζουν τα αυτόματα— είναι ο ίδιος, τα προγράμματα που ορίζονται άμεσα από τις συμπαγείς μορφές των αυτομάτων έχουν σημαντικά μικρότερο αριθμό κανόνων. Ο αριθμός

Πίνακας 3.11: Οι επιτυχημένες εκτελέσεις του αυτομάτου με μέτρηση πάνω στο  $\mathcal{T}_{\mathcal{L},\mathcal{A}}$ .

Επιτυχημένες εκτελέσεις					Αναθέσεις
$n_4 (a, \underline{b}, c)$	$n_5 (c, \underline{d}, \underline{e})$	$n_2 (a, f, \underline{c})$	$n_3 (g, \underline{h})$	$n_1 (\underline{a}, g, f)$	$\mathcal{Z}$
$(\emptyset, Q^{\{1,3\}}, 0)$	$(\{2\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^{\{1\}}, 2)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{1, 2\}, Q^0, 4)$	$\{d, c, a, g\}$
$(\emptyset, Q^{\{1,3\}}, 0)$	$(\{2\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^{\{1\}}, 2)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{1, 2, 3\}, Q^0, 5)$	$\{d, c, a, g, f\}$
$(\emptyset, Q^{\{1,3\}}, 0)$	$(\{2\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^{\{1\}}, 2)$	$(\{2\}, Q^0, 1)$	$(\{1, 3\}, Q^0, 5)$	$\{d, c, h, a, f\}$
$(\emptyset, Q^{\{1,3\}}, 0)$	$(\{2\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^{\{1\}}, 2)$	$(\{2\}, Q^0, 1)$	$(\{1, 2\}, Q^0, 5)$	$\{d, c, h, a, g\}$
$(\emptyset, Q^{\{1,3\}}, 0)$	$(\{2\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^{\{1\}}, 2)$	$(\{2\}, Q^0, 1)$	$(\{1, 2, 3\}, Q^0, 6)$	$\{d, c, h, a, g, f\}$
$(\emptyset, Q^{\{1,3\}}, 0)$	$(\{3\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^{\{1\}}, 2)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{1, 2\}, Q^0, 4)$	$\{e, c, a, g\}$
$(\emptyset, Q^{\{1,3\}}, 0)$	$(\{3\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^{\{1\}}, 2)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{1, 2, 3\}, Q^0, 5)$	$\{e, c, a, g, f\}$
$(\emptyset, Q^{\{1,3\}}, 0)$	$(\{3\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^{\{1\}}, 2)$	$(\{2\}, Q^0, 1)$	$(\{1, 3\}, Q^0, 5)$	$\{e, c, h, a, f\}$
$(\emptyset, Q^{\{1,3\}}, 0)$	$(\{3\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^{\{1\}}, 2)$	$(\{2\}, Q^0, 1)$	$(\{1, 2\}, Q^0, 5)$	$\{e, c, h, a, g\}$
$(\emptyset, Q^{\{1,3\}}, 0)$	$(\{3\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^{\{1\}}, 2)$	$(\{2\}, Q^0, 1)$	$(\{1, 2, 3\}, Q^0, 6)$	$\{e, c, h, a, g, f\}$
$(\emptyset, Q^{\{1,3\}}, 0)$	$(\{2, 3\}, Q^0, 2)$	$(\{3\}, Q^{\{1\}}, 3)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{1, 2\}, Q^0, 5)$	$\{d, e, c, a, g\}$
$(\emptyset, Q^{\{1,3\}}, 0)$	$(\{2, 3\}, Q^0, 2)$	$(\{3\}, Q^{\{1\}}, 3)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{1, 2, 3\}, Q^0, 6)$	$\{d, e, c, a, g, f\}$
$(\emptyset, Q^{\{1,3\}}, 0)$	$(\{2, 3\}, Q^0, 2)$	$(\{3\}, Q^{\{1\}}, 3)$	$(\{2\}, Q^0, 1)$	$(\{1, 3\}, Q^0, 6)$	$\{d, e, c, h, a, f\}$
$(\emptyset, Q^{\{1,3\}}, 0)$	$(\{2, 3\}, Q^0, 2)$	$(\{3\}, Q^{\{1\}}, 3)$	$(\{2\}, Q^0, 1)$	$(\{1, 2\}, Q^0, 6)$	$\{d, e, c, h, a, g\}$
$(\emptyset, Q^{\{1,3\}}, 0)$	$(\{2, 3\}, Q^0, 2)$	$(\{3\}, Q^{\{1\}}, 3)$	$(\{2\}, Q^0, 1)$	$(\{1, 2, 3\}, Q^0, 7)$	$\{d, e, c, h, a, g, f\}$
$(\{2\}, Q^0, 1)$	$(\{2\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{2\}, Q^0, 4)$	$\{b, d, c, g\}$
$(\{2\}, Q^0, 1)$	$(\{2\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{2, 3\}, Q^0, 5)$	$\{b, d, c, g, f\}$
$(\{2\}, Q^0, 1)$	$(\{2\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{1, 2\}, Q^0, 5)$	$\{b, d, c, g, a\}$
$(\{2\}, Q^0, 1)$	$(\{2\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{1, 2, 3\}, Q^0, 6)$	$\{b, d, c, g, a, f\}$
$(\{2\}, Q^0, 1)$	$(\{2\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\{2\}, Q^0, 1)$	$(\{2\}, Q^0, 5)$	$\{b, d, c, h, g\}$
$(\{2\}, Q^0, 1)$	$(\{2\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\{2\}, Q^0, 1)$	$(\{1, 2\}, Q^0, 6)$	$\{b, d, c, h, a, g\}$
$(\{2\}, Q^0, 1)$	$(\{2\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\{2\}, Q^0, 1)$	$(\{1, 3\}, Q^0, 6)$	$\{b, d, c, h, a, f\}$
$(\{2\}, Q^0, 1)$	$(\{2\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\{2\}, Q^0, 1)$	$(\{2, 3\}, Q^0, 6)$	$\{b, d, c, h, g, f\}$
$(\{2\}, Q^0, 1)$	$(\{2\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\{2\}, Q^0, 1)$	$(\{1, 2, 3\}, Q^0, 7)$	$\{b, d, c, h, a, g, f\}$
$(\{2\}, Q^0, 1)$	$(\{3\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{2\}, Q^0, 4)$	$\{b, e, c, g\}$
$(\{2\}, Q^0, 1)$	$(\{3\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{2, 3\}, Q^0, 5)$	$\{b, e, c, g, f\}$
$(\{2\}, Q^0, 1)$	$(\{3\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{1, 2\}, Q^0, 5)$	$\{b, e, c, g, a\}$
$(\{2\}, Q^0, 1)$	$(\{3\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{1, 2, 3\}, Q^0, 6)$	$\{b, e, c, g, a, f\}$
$(\{2\}, Q^0, 1)$	$(\{3\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\{2\}, Q^0, 1)$	$(\{2\}, Q^0, 5)$	$\{b, e, c, h, g\}$
$(\{2\}, Q^0, 1)$	$(\{3\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\{2\}, Q^0, 1)$	$(\{1, 2\}, Q^0, 6)$	$\{b, e, c, h, a, g\}$
$(\{2\}, Q^0, 1)$	$(\{3\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\{2\}, Q^0, 1)$	$(\{1, 3\}, Q^0, 6)$	$\{b, e, c, h, a, f\}$
$(\{2\}, Q^0, 1)$	$(\{3\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\{2\}, Q^0, 1)$	$(\{2, 3\}, Q^0, 6)$	$\{b, e, c, h, g, f\}$
$(\{2\}, Q^0, 1)$	$(\{3\}, Q^{\{3\}}, 1)$	$(\{3\}, Q^0, 3)$	$(\{2\}, Q^0, 1)$	$(\{1, 2, 3\}, Q^0, 7)$	$\{b, e, c, h, a, g, f\}$
$(\{2\}, Q^0, 1)$	$(\{2, 3\}, Q^0, 2)$	$(\emptyset, Q^{\{1,3\}}, 3)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{1, 2, 3\}, Q^0, 6)$	$\{b, d, e, a, g, f\}$
$(\{2\}, Q^0, 1)$	$(\{2, 3\}, Q^0, 2)$	$(\emptyset, Q^{\{1,3\}}, 3)$	$(\{2\}, Q^0, 1)$	$(\{1, 3\}, Q^0, 6)$	$\{b, d, e, h, a, f\}$
$(\{2\}, Q^0, 1)$	$(\{2, 3\}, Q^0, 2)$	$(\emptyset, Q^{\{1,3\}}, 3)$	$(\{2\}, Q^0, 1)$	$(\{1, 2, 3\}, Q^0, 7)$	$\{b, d, e, h, a, f, g\}$
$(\{2\}, Q^0, 1)$	$(\{2, 3\}, Q^0, 2)$	$(\{3\}, Q^0, 4)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{2\}, Q^0, 5)$	$\{b, d, e, c, g\}$
$(\{2\}, Q^0, 1)$	$(\{2, 3\}, Q^0, 2)$	$(\{3\}, Q^0, 4)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{2, 3\}, Q^0, 6)$	$\{b, d, e, c, g, f\}$
$(\{2\}, Q^0, 1)$	$(\{2, 3\}, Q^0, 2)$	$(\{3\}, Q^0, 4)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{1, 2\}, Q^0, 6)$	$\{b, d, e, c, g, a\}$
$(\{2\}, Q^0, 1)$	$(\{2, 3\}, Q^0, 2)$	$(\{3\}, Q^0, 4)$	$(\emptyset, Q^{\{2\}}, 0)$	$(\{1, 2, 3\}, Q^0, 7)$	$\{b, d, e, c, g, a, f\}$
$(\{2\}, Q^0, 1)$	$(\{2, 3\}, Q^0, 2)$	$(\{3\}, Q^0, 4)$	$(\{2\}, Q^0, 1)$	$(\{2\}, Q^0, 6)$	$\{b, d, e, c, h, g\}$
$(\{2\}, Q^0, 1)$	$(\{2, 3\}, Q^0, 2)$	$(\{3\}, Q^0, 4)$	$(\{2\}, Q^0, 1)$	$(\{1, 2\}, Q^0, 7)$	$\{b, d, e, c, h, a, g\}$
$(\{2\}, Q^0, 1)$	$(\{2, 3\}, Q^0, 2)$	$(\{3\}, Q^0, 4)$	$(\{2\}, Q^0, 1)$	$(\{1, 3\}, Q^0, 7)$	$\{b, d, e, c, h, a, f\}$
$(\{2\}, Q^0, 1)$	$(\{2, 3\}, Q^0, 2)$	$(\{3\}, Q^0, 4)$	$(\{2\}, Q^0, 1)$	$(\{2, 3\}, Q^0, 7)$	$\{b, d, e, c, h, g, f\}$
$(\{2\}, Q^0, 1)$	$(\{2, 3\}, Q^0, 2)$	$(\{3\}, Q^0, 4)$	$(\{2\}, Q^0, 1)$	$(\{1, 2, 3\}, Q^0, 8)$	$\{b, d, e, c, h, a, g, f\}$



αυτός είναι ίδιος με τον αριθμό των πιθανών συνθηκών που απαιτούνται ώστε τα αντίστοιχα αυτόματα να μπορούν να αναπαρασταθούν σε συμπαγή μορφή. Σημειώνεται ότι οι συνθήκες αυτές ορίζουν διαμερίσεις του  $\Gamma_{\tau,w}$ .

Συνοψίζοντας, αξίζει να τονίσουμε ότι η προσέγγιση αυτή πραγματοποιεί αποτιμήσεις πάνω σε δεντροδιασπάσεις χωρίς να έχουμε μια αναλυτική περιγραφή του αυτομάτου. Θα μπορούσαμε μάλιστα να πούμε ότι παράλληλα με την αποτίμηση των αντίστοιχων datalog προγραμμάτων, έχουμε την κατασκευή εκείνου του τμήματος του αυτομάτου το οποίο είναι πραγματικά απαραίτητο για την αποτίμηση στη δεδομένη είσοδο. Είναι προφανές ότι η μέθοδος αυτή γενικεύεται φυσικά σε κάθε στιγμιότυπο προβλήματος στο οποίο έχουμε δεδομένη μία συμπαγή αναπαράσταση για το αυτόματο.

Η παρουσίαση αυτής της προσέγγισης αρχίζει με τη απόδειξη –δίνεται στο λήμμα που ακολουθεί– για την ύπαρξη ενός datalog προγράμματος που, με είσοδο τη βάση δεδομένων  $\mathcal{D}_A \cup \mathcal{D}_I$ , παράγει τις συνθήκες που εμφανίζονται στο ορισμό της συμπαγούς μορφής του αυτομάτου για το κάλυμμα κορυφών και δόθηκαν στον Πίνακα 3.6.

**Λήμμα 3.5.1.** Υπάρχει ένα datalog πρόγραμμα  $\Pi_{VC,w}^{\text{cond}}$  που, με είσοδο τη βάση δεδομένων  $\mathcal{D}_G \cup \mathcal{D}_I$  που αντιστοιχεί σε ένα γράφο  $\mathcal{G}$  και μία ειδική δεντροδιάσπαση  $\mathcal{I}$  του  $\mathcal{G}$  πλάτους  $w$ , παράγει τις συνθήκες που συμμετέχουν στον ορισμό του αυτομάτου για το κάλυμμα κορυφών.

Το  $\Pi_{VC,w}^{\text{cond}}$  παράγει για κάθε κόμβο της δεντροδιάσπασης το πολύ  $c_{VC}^w$  γεγονότα, όπου

$$c_{VC}^w = 3 \cdot 2^{w+1} + w^2 + 5(w + 1).$$

**Απόδειξη.** Η ορθότητα του προγράμματος που ακολουθεί επαληθεύεται άμεσα.

$\Pi_{VC,w}^{\text{cond}}$ .

$$ver_{j,u,v}^k : \quad \text{vertex-}j(x) \leftarrow \text{decomp}^u(x, x_1, \dots, x_u), \text{decomp}^v(y, y_1, \dots, y_v), \\ \text{succ}_k(y, x), \{\neg(x_j = y_i) \mid i \in W_v\}$$

για  $k = 0, 1$ .

$$ver_{j,u}^r : \quad \text{vertex-}j(x) \leftarrow \text{decomp}^u(x, x_1, \dots, x_u), \text{root}(x)$$

για  $j \in W_u$ .

$$set_I : \quad \text{set-}I(x) \leftarrow \{\text{vertex-}j(x) \mid j \in I\}.$$

$$Vpos_I : \quad \text{Vpos-}I(x) \leftarrow \{\text{vertex-}j(x) \mid j \in I\}, \{\neg \text{vertex-}j(x) \mid j \in W \setminus I\}$$

για  $j, u, v \in W, j \leq u, I, J \subseteq W, W_i = \{1, \dots, i\} \subseteq W, i = u, v$ .

$$S_{I,J,L,u,v,\bar{j}}^k : \quad \text{SameInPar-}IJ(x) \leftarrow \text{decomp}^u(x, x_1, \dots, x_u), \text{decomp}^v(y, y_1, \dots, y_v), \\ \text{succ}_k(y, x), \text{Vpos-}L(x), \{(x_{i_k} = y_{j_k}) \mid 1 \leq k \leq \ell\}$$

για  $k = 0, 1$  &  $I, L \subseteq W_u, J \subseteq W_v$  τ.ω.  $|I \setminus L| = |J|$ ,

$$I \setminus L = \{i_1, \dots, i_\ell\}, \bar{j} = (j_1, \dots, j_\ell) \in J^\ell, j_i \neq j_k, 1 \leq i, k \leq \ell.$$

$$adj_{u,i,J} : \quad \text{Adj-}iJ(x) \leftarrow \text{decomp}^u(x, x_1, \dots, x_u), \{\text{edg}(x_i, x_j) \mid j \in J\},$$

$$\begin{aligned}
& \{\neg \text{edg}(x_i, x_j) \mid j \in W_u \setminus J\} \\
& \text{για } u \in W, J \subseteq W_u. \\
\text{Adj}_{i,J} : & \quad \text{AdjInPar-}iJ(x) \leftarrow \text{Adj-}iI(x), \text{SameInPar-}IJ(x). \\
\text{edge}_{u,i,j} : & \quad \text{edge-}ij(x) \leftarrow \text{decomp}^u(x, x_1, \dots, x_u), \text{vertex-}i(x), \text{vertex-}j(x), \\
& \quad \text{edg}(x_i, x_j) \\
& \text{για } u \in W, i, j \in W_u. \\
\text{Adj}V \setminus_{I,J,L,F} : & \quad \text{AdjInParVpos} \setminus L-I(x) \leftarrow \text{Vpos-}J(x), \{\text{AdjInPar-}jI_j(x) \mid j \in J \setminus L\} \\
& \text{για } I, J \subseteq W, L \subseteq J, \& F = \{I_j \mid j \in J \setminus L\} \subseteq \mathcal{P}(I) \text{ τ.ω. } \cup_{j \in J} I_j = I. \\
\text{edge}V \setminus_{i,j,L} : & \quad \text{EdgeVpos} \setminus L(x) \leftarrow \text{edge-}ij(x) \\
& \text{για } i, j \in W \setminus L, L \subseteq W. \quad \square
\end{aligned}$$

**Παράδειγμα 3.5.2.** Ας θεωρήσουμε ξανά το γράφο  $\mathcal{G}$  και την αντίστοιχη δεντροδιάσπαση  $\mathcal{I}$  του Σχήματος 3.1. Τα γεγονότα που παράγονται από το παραπάνω πρόγραμμα και αντιστοιχούν στον κόμβο  $n_4$  της δεδομένης δεντροδιάσπασης είναι τα παρακάτω.

$$\begin{aligned}
& \text{vertex-}2(n_4), \text{ set-}\{2\}(n_4), \text{ set-}\emptyset(n_4), \text{ Vpos-}\{2\}(n_4), \\
& \text{SameInPar-}\{1\}\{1\}(n_4), \text{ SameInPar-}\{2\}\emptyset(n_4), \text{ SameInPar-}\emptyset\emptyset(n_4), \\
& \text{SameInPar-}\{3\}\{3\}(n_4), \text{ SameInPar-}\{1, 2\}\{1\}(n_4), \text{ SameInPar-}\{1, 3\}\{1, 3\}(n_4), \\
& \text{SameInPar-}\{2, 3\}\{3\}(n_4), \text{ SameInPar-}\{1, 2, 3\}\{1, 3\}(n_4), \\
& \text{Adj-}1\{2, 3\}(n_4), \text{ Adj-}2\{1, 3\}(n_4), \text{ Adj-}3\{1, 2\}(n_4), \\
& \text{AdjInPar-}1\{3\}(n_4), \text{ AdjInPar-}2\{1, 3\}(n_4), \text{ AdjInPar-}3\{1\}(n_4), \\
& \text{AdjInParVpos} \setminus \emptyset\{1, 3\}(n_4), \text{ AdjInParVpos} \setminus \{2\}\emptyset(n_4).
\end{aligned}$$

Στη συνέχεια παρουσιάζουμε datalog προγράμματα που, χρησιμοποιώντας τα άτομα που παράγονται μέσω του Λήμματος 3.5.1, υπολογίζουν τις λύσεις στα προβλήματα αποτίμησης που μελετάμε και ορίζονται μέσω του MSO τύπου που ορίζει καλύμματα κορυφών.

Αξίζει να σημειωθεί ότι στα προγράμματα μας δεν υπάρχουν κανόνες που αντιστοιχούν στις κλάσεις των μεταβάσεων που οδηγούν σε αποτυχία αποδοχής. Πιο συγκεκριμένα, δεν γράφουμε κανόνες για τις μεταβάσεις της μορφής

$$q_1, q_2, \gamma, \bar{I} \mapsto Q_f$$

έχοντας στόχο την περαιτέρω μείωση του μεγέθους των προγραμμάτων μας. Ειδικά για τα προβλήματα που σχετίζονται με το κάλυμμα κορυφών, προκύπτει τελικά ότι το συμπλήρωμα των μεταβάσεων που δεν οδηγούν σε αποτυχία αποδοχής είναι ακριβώς οι μεταβάσεις που συμμετέχουν στις επιτυχημένες εκτελέσεις. Κατά συνέπεια, ο υπολογισμός των πιθανών καταστάσεων παρέχει άμεσα και τις επιτυχημένες καταστάσεις του αυτομάτου για το κάλυμμα κορυφών.

**Βελτιστοποιημένη datalog λύση για το VERTEX COVER.** Παρουσιάζουμε αρχικά στην Πρόταση 3.5.4 τη λύση στο πρόβλημα υπολογισμού καλυμμάτων κορυφών που ορίζεται παρακάτω.

<p style="text-align: center; margin: 0;">VERTEX COVER</p> <p style="margin: 10px 0;"><b>Είσοδος:</b> Ένας γράφος <math>\mathcal{G} = (V, E)</math>.</p> <p style="margin: 10px 0;"><b>Έξοδος:</b> Τα καλύμματα κορυφών του <math>\mathcal{G}</math>.</p>
---

**Πρόταση 3.5.1.** Υπάρχει ένα datalog<sup>-</sup> ερώτημα  $Q_{VC}^w = (\Pi_{VC}^w, \text{VertCov})$ , βαθμού 2, το οποίο επιλύει το VERTEX COVER πρόβλημα για κάθε γράφο  $\mathcal{G}$  με ειδική δεντροδιάσπαση  $\mathcal{I} = (\mathcal{T}, (\bar{a}_n)_{n \in T})$  πλάτους  $w$ . Πιο συγκεκριμένα, έχουμε

$$\text{το } B \text{ είναι κάλυμμα κορυφών αν } B \in Q_{VC}^w(\mathcal{D}_{\mathcal{G}} \cup \mathcal{D}_{\mathcal{I}}).$$

Η πολυπλοκότητα του υπολογισμού είναι το πολύ  $(c_{VC}^w + a) \cdot |T|$ , όπου το  $a$  είναι ο αριθμός των καλυμμάτων κορυφών του δεδομένου γράφου  $\mathcal{G}$  και  $c_{VC}^w = 3 \cdot 2^{w+1} + w^2 + 5(w + 1)$ .

**Απόδειξη.** Οι επιτυχημένες καταστάσεις των αυτομάτων για το κάλυμμα κορυφών ταυτίζονται με εκείνο το υποσύνολο των πιθανών καταστάσεων που υπολογίζεται μόνο μέσω των μεταβάσεων που δεν οδηγούν σε αποτυχία αποδοχής. Έτσι, στην περίπτωση του καλύμματος κορυφών, το αποτέλεσμα της αποτίμησης του μη-βελτιστοποιημένου προγράμματος, που ορίζεται για κάθε μετάβαση που δεν οδηγεί σε αποτυχία αποδοχής, ταυτίζεται με το αποτέλεσμα της αποτίμησης του βελτιστοποιημένου προγράμματος, που προκύπτει μέσω αναγραφής με φίλτρα τις επιτυχημένες καταστάσεις του αντίστοιχου αυτομάτου. Συνεπώς, αρκεί να ορίσουμε το

$$\Pi_{VC}^w = \Pi_{VC,w}^{\text{cond}} \cup A_{VC}^w$$

όπου το  $A_{VC}^w$  υπολογίζει τις αναθέσεις που αντιστοιχούν σε εκείνες τις πιθανές εκτελέσεις που συνθέτονται χωρίς μεταβάσεις αποτυχίας. Οι αναθέσεις αυτές είναι ακριβώς όλα τα καλύμματα κορυφών του δεδομένου γράφου.

$$\begin{aligned}
 A_{VC}^w. \quad a_{L,I} : \quad & \text{VertCov-}Q^I(x, y) \leftarrow \text{leaf}(x), \text{set-}L(x), \neg \text{EdgeVpos} \setminus L(x), \\
 & \text{AdjInParVpos} \setminus L-I(x), \text{decomp-}L(x, y) \\
 a_{J,S,L,I \cup P} : \quad & \text{VertCov-}Q^I(x, y) \leftarrow \text{VertCov-}Q^J(x_1, u), \text{VertCov-}Q^S(x_2, v), \text{succ}_0(x, x_1), \\
 & \text{succ}_1(x, x_2), \text{set-}L(x), \neg \text{EdgeVpos} \setminus L(x), \\
 & \{\neg \text{vertex-}j(x) \mid j \in (J \cup S) \setminus L\}, \\
 & \text{SameInPar-}(J \cup S)U(x), \text{AdjInParVpos} \setminus L-P(x), \\
 & \text{decomp-}L(x, z), \text{union}(y, z, u, v)
 \end{aligned}$$

$$vc : \quad \text{VertCov}(x) \leftarrow \text{VertCov-}Q^\emptyset(x, y), \text{root}(x)$$

όπου οι κανόνες  $a_{L,I}$  ορίζονται για κάθε  $L, I \subseteq W$ , ενώ οι κανόνες  $a_{J,S,L,I|U,P}$  για  $J, S, I, U, P \subseteq W$  τ.ω.  $|(J \cup S) \setminus L| = |U|$ ,  $U \cup P = I$  και  $U \cap P = \emptyset$ .

Αποδεικνύεται άμεσα ότι ισχύουν τα εξής: το  $\text{VertCov-}Q^I(n, C)$  συμπεραίνεται από το πρόγραμμα  $\Pi_{VC}^w$  με είσοδο τη βάση δεδομένων  $\mathcal{D}_G \cup \mathcal{D}_I$  αν το σύνολο  $C$  είναι ο περιορισμός στο  $V_n = \{v \in V \mid \text{node}(v) \in T_n\}$  ενός καλύμματος κορυφών  $B$  του γράφου  $\mathcal{G} = (V, E)$  αν το  $C$  είναι ένα κάλυμμα κορυφών του  $\mathcal{G}|_{V_n}$ .  $\square$

**Παράδειγμα 3.5.3.** Ας θεωρήσουμε ξανά το γράφο  $\mathcal{G}$  και την αντίστοιχη δεντροδιάσπαση  $\mathcal{I}$  του Σχήματος 3.1. Τα γεγονότα που παράγονται από το παραπάνω πρόγραμμα και αντιστοιχούν στα φύλλα της δεδομένης δεντροδιάσπασης είναι τα παρακάτω.

$n_4$ :  $\text{VertCov-}Q^{\{1,3\}}(n_4, \emptyset)$ ,  $\text{VertCov-}Q^\emptyset(n_4, \{b\})$  από στιγμιοτυποποιήσεις των κανόνων  $a_{L,I}$  για  $L = \emptyset$  και  $L = \{2\}$  αντίστοιχα.

$n_5$ :  $\text{VertCov-}Q^{\{3\}}(n_5, \{d\})$ ,  $\text{VertCov-}Q^{\{3\}}(n_5, \{e\})$ ,  $\text{VertCov-}Q^\emptyset(n_5, \{d, e\})$  από στιγμιοτυποποιήσεις των κανόνων  $a_{L,I}$  για  $L = \{2\}$ ,  $L = \{3\}$  και  $L = \{2, 3\}$  αντίστοιχα.

$n_3$ :  $\text{VertCov-}Q^{\{2\}}(n_3, \emptyset)$ ,  $\text{VertCov-}Q^\emptyset(n_3, \{h\})$  από στιγμιοτυποποιήσεις των κανόνων  $a_{L,I}$  για  $L = \emptyset$  και  $L = \{2\}$  αντίστοιχα.

**Βελτιστοποιημένη datalog λύση για το  $k$ -VERTEX COVER.** Στην πρόταση που ακολουθεί παρουσιάζουμε μία βελτιστοποιημένη datalog λύση για το πρόβλημα  $k$ -VERTEX COVER και δίνουμε ένα παράδειγμα επίλυσης.

$k$ -VERTEX COVER

**Είσοδος:** Ένας γράφος  $\mathcal{G} = (V, E)$  & ένας θετικός ακέραιος  $k$ .

**Ερώτηση:** Έχει ο γράφος  $\mathcal{G}$  ένα κάλυμμα κορυφών μεγέθους το πολύ  $k$ ;

**Πρόταση 3.5.2.** Υπάρχει ένα ερώτημα  $Q_{k-VC}^w = (\Pi_{k-VC}^w, k-VC\text{-yes})$  της μοναδιαίας datalog<sup>-</sup> που επιλύει το  $k$ -VERTEX COVER για κάθε γράφο  $\mathcal{G}$  με ειδική δεντροδιάσπαση  $\mathcal{I} = (\mathcal{T}, (\bar{a}_n)_{n \in \mathcal{T}})$  πλάτους  $w$ . Πιο συγκεκριμένα, έχουμε ότι

η απάντηση στο  $k$ -VERTEX COVER είναι «ναι» αν  $\Pi_{k-VC}^w(\mathcal{D}_G \cup \mathcal{D}_I) \models k-VC\text{-yes}$ .

Η απάντηση στο πρόβλημα υπολογίζεται μετά από το πολύ  $(c_{VC}^w + 2^{3(w+1)}) \cdot |T|$  στιγμιοτυποποιήσεις, όπου  $c_{VC}^w = 3 \cdot 2^{w+1} + w^2 + 5(w+1)$ .

**Απόδειξη.** Ορίζουμε το  $\Pi_{k-VC}^w = \Pi_{VC,w}^{\text{cond}} \cup P_{k-VC}^w$ .

$$\begin{aligned}
 P_{k-VC}^w. \quad r_{L,I} : & \text{pot-}Q^I \ell(x) \leftarrow \text{leaf}(x), \text{set-}L(x), \neg \text{EdgeVpos} \setminus L(x), \\
 & \text{AdjInParVpos} \setminus L-I(x) \\
 & \text{για } L, I \subseteq W, |L| = \ell \leq k. \\
 r_{Jj,Ss,L,I|U,P} : & \text{pot-}Q^I i(x) \leftarrow \text{pot-}Q^J j(x_1), \text{pot-}Q^S s(x_2), \text{succ}_0(x, x_1), \text{succ}_1(x, x_2), \\
 & \text{set-}L(x), \neg \text{EdgeVpos} \setminus L(x), \text{SameInPar-}(J \cup S)U(x), \\
 & \{\neg \text{vertex-}j(x) \mid j \in (J \cup S) \setminus L\}, \text{AdjInParVpos} \setminus L-P(x) \\
 & \text{για } J, S, I, U, P \subseteq W \text{ τ.ω. } |(J \cup S) \setminus L| = |U|, U \cup P = I, U \cap P = \emptyset, \\
 & \quad \& \text{ για } j, s, i \leq k \text{ τ.ω. } i = j + s + |L|. \\
 r_i : & k\text{-}VC\text{-yes} \leftarrow \text{pot-}Q^\emptyset i(x), \text{root}(x) \\
 & \text{για } i \leq k.
 \end{aligned}$$

Αποδεικνύεται εύκολα ότι το  $Q_{k-VC}^w = (\Pi_{k-VC}^w, k\text{-}VC\text{-yes})$  ικανοποιεί την επιθυμητή συνθήκη και ότι η αποτίμηση του πραγματοποιείται μετά από το πολύ  $(c_{VC}^w + 2^{3(w+1)}) \cdot |T|$  στιγμιοτυποποιήσεις.  $\square$

**Παράδειγμα 3.5.4.** Ας θεωρήσουμε ξανά το γράφο  $\mathcal{G}$  και την αντίστοιχη δεντροδιάσπαση  $\mathcal{I}$  του Σχήματος 3.1. Τα γεγονότα που παράγονται από το παραπάνω πρόγραμμα και αντιστοιχούν στα φύλλα της δεδομένης δεντροδιάσπασης είναι τα παρακάτω.

$$\begin{aligned}
 n_4: & \text{pot-}Q^{\{1,3\}}0(n_4), \text{pot-}Q^\emptyset 1(n_4) \text{ από στιγμιοτυποποιήσεις των κανόνων } r_{L,I} \\
 & \text{για } L = \emptyset \text{ και } L = \{2\} \text{ αντίστοιχα.} \\
 n_5: & \text{pot-}Q^{\{3\}}1(n_5) \text{ και } \text{pot-}Q^\emptyset 2(n_5) \text{ από στιγμιοτυποποιήσεις των κανόνων } r_{L,I} \\
 & \text{για } L = \{2\}, L = \{3\} \text{ και } L = \{2, 3\} \text{ αντίστοιχα.} \\
 n_3: & \text{pot-}Q^{\{2\}}0(n_3), \text{pot-}Q^\emptyset 1(n_3) \text{ από στιγμιοτυποποιήσεις των κανόνων } r_{L,I} \\
 & \text{για } L = \emptyset \text{ και } L = \{2\} \text{ αντίστοιχα.}
 \end{aligned}$$

Ανάλογα πραγματοποιούνται οι υπολογισμοί στους εσωτερικούς κόμβους. Για παράδειγμα, στον κόμβο  $n_2$  παράγονται τα παρακάτω γεγονότα.

$$\begin{aligned}
 n_2: & \text{pot-}Q^{\{1\}}2(n_2) \text{ από στιγμιοτυποποίηση του κανόνα } r_{Jj,Ss,L,I|U,P} \\
 & \text{για } Jj = \{1, 3\}0, Ss = \{3\}1, L = \{3\}, P = \emptyset, U = \{1\}, \\
 & \text{pot-}Q^{\{1\}}3(n_2) \text{ από στιγμιοτυποποίηση του κανόνα } r_{Jj,Ss,L,I|U,P} \\
 & \text{για } Jj = \{1, 3\}0, Ss = \emptyset 2, L = \{3\}, P = \emptyset, U = \{1\}, \\
 & \text{pot-}Q^\emptyset 3(n_2) \text{ από στιγμιοτυποποίηση του κανόνα } r_{Jj,Ss,L,I|U,P} \\
 & \text{για } Jj = \emptyset 1, Ss = \{3\}1, L = \{3\}, P = \emptyset, U = \emptyset,
 \end{aligned}$$

pot- $Q^{\{1,3\}}3(n_2)$  από στιγμιοτυποποίηση του κανόνα  $r_{Jj,Ss,L,I|U,P}$

για  $Jj = \emptyset 1, Ss = \emptyset 2, L = \emptyset, P = \{1, 3\}, U = \emptyset,$

pot- $Q^{\emptyset}4(n_2)$  από στιγμιοτυποποίηση του κανόνα  $r_{Jj,Ss,L,I|U,P}$

για  $Jj = \emptyset 1, Ss = \emptyset 2, L = \{3\}, P = \emptyset, U = \emptyset,$

...

**Βελτιστοποιημένη datalog<sup>min</sup> λύση για το MINIMUM CARDINALITY VERTEX COVER.** Στη συνέχεια παρουσιάζεται η λύση στο πρόβλημα MINIMUM CARDINALITY VERTEX COVER που ορίζεται παρακάτω.

MINIMUM CARDINALITY VERTEX COVER

**Είσοδος:** Ένας γράφος  $\mathcal{G} = (V, E)$ .

**Εξοδος:** Το μέγεθος των καλυμμάτων κορυφών του  $\mathcal{G}$  με ελάχιστο μέγεθος.

**Πρόταση 3.5.3.** Υπάρχει ένα datalog<sup>min</sup> ερώτημα  $Q_{C \min VC}^w = (\Pi_{C \min VC}^w, C\text{-min-VC})$  βαθμού 2 που επιλύει το πρόβλημα MINIMUM CARDINALITY VERTEX COVER για κάθε γράφο  $\mathcal{G}$  με ειδική δεντροδιάσπαση  $\mathcal{I} = (\mathcal{T}, (\bar{a}_n)_{n \in T})$  πλάτους  $w$ . Πιο συγκεκριμένα, έχουμε ότι

το  $i$  είναι το ελάχιστο μέγεθος αν  $i \in Q_{C \min VC}^w(\mathcal{D}_{\mathcal{G}}^s \cup \mathcal{D}_{\mathcal{I}})$ .

Η λύση υπολογίζεται μετά από το πολύ  $(2^{3(w+1)} + 5 \cdot 2^{w+1} + w^2 + 5(w+1)) \cdot |T|$  στιγμιοτυποποιήσεις.

**Απόδειξη.** Ορίζουμε το  $\Pi_{C \min VC}^w = \Pi_{VC,w}^{\text{cond}} \cup P_{C \min VC}^w$ .

$$\begin{array}{ll}
 P_{C \min VC}^w \cdot & p_{L,I} : \quad \text{pot-}Q^I(x, \ell) \leftarrow \text{leaf}(x), \text{"VC-condition-LI"} \\
 & m_I : \quad \text{min-pot-}Q^I(x, \min(i)) \leftarrow \text{pot-}Q^I(x, i) \\
 & p_{J,S,L,I|U,P} : \quad \text{pot-}Q^I(x, i) \leftarrow \text{min-pot-}Q^J(x_1, i_1), \text{min-pot-}Q^S(x_2, i_2), \\
 & \quad \text{succ}_0(x, x_1), \text{succ}_1(x, x_2), \text{sum}(i, \ell, i_1, i_2), \\
 & \quad \text{"VC-condition-JSLI|UP"} \\
 & s_I : \quad \text{min-suc}(i) \leftarrow \text{min-pot-}Q^I(x, i), \text{root}(x) \\
 & r : \quad \text{C-min-VC}(\min(i)) \leftarrow \text{min-suc}(i)
 \end{array}$$

όπου οι κανόνες  $p_{L,I}, m_I, s_I$  γράφονται για όλα τα  $L, I \subseteq W$ , και οι κανόνες  $p_{J,S,L,I|U,P}$  γράφονται για σύνολα  $J, S, I, U, P \subseteq W$  τ.ω.  $|(J \cup S) \setminus L| = |U|$ ,  $U \cup P = I$  και

$U \cap P = \emptyset$ . Το "VC-condition-LI" αποτελεί συντομογραφία για την ακολουθία ατόμων  $\text{set-}L(x)$ ,  $\neg \text{EdgeVpos} \setminus L(x)$ ,  $\text{AdjInParVpos} \setminus L-I(x)$ , ενώ το "VC-condition-JSLI|UP" αποτελεί συντομογραφία για την ακολουθία ατόμων  $\text{set-}L(x)$ ,  $\neg \text{EdgeVpos} \setminus L(x)$ ,  $\{\neg \text{vertex-}j(x) \mid j \in (J \cup S) \setminus L\}$ ,  $\text{SameInPar-}(J \cup S)U(x)$ ,  $\text{AdjInParVpos} \setminus L-P(x)$ .

Αποδεικνύεται εύκολα ότι το  $Q_{C \min VC}^w = (\Pi_{C \min VC}^w, C\text{-min-VC})$  ικανοποιεί την επιθυμητή συνθήκη και ότι η αποτίμηση του πραγματοποιείται μετά από το πολύ  $(2^{3(w+1)} + 5 \cdot 2^{w+1} + w^2 + 5(w+1)) \cdot |T|$  στιγμιοτυποποιήσεις.  $\square$

**Βελτιστοποιημένη datalog<sup>min</sup> λύση για το MINIMUM VERTEX COVER.** Στη συνέχεια παρουσιάζεται η λύση στο πρόβλημα MINIMUM VERTEX COVER που είναι το τελευταίο πρόβλημα που μελετάμε και ορίζεται παρακάτω.

MINIMUM VERTEX COVER

**Είσοδος:** Ένας γράφος  $\mathcal{G} = (V, E)$ .

**Έξοδος:** Τα καλύμματα κορυφών του  $\mathcal{G}$  με το ελάχιστο μέγεθος.

**Πρόταση 3.5.4.** Υπάρχει ένα datalog<sup>min</sup> ερώτημα  $Q_{\min VC}^w = (\Pi_{\min VC}^w, \text{min-VertCov})$  βαθμού 2 που επιλύει το πρόβλημα MINIMUM VERTEX COVER για κάθε γράφο  $\mathcal{G}$  με ειδική δεντροδιάσπαση  $\mathcal{I} = (\mathcal{T}, (\bar{a}_n)_{n \in \mathcal{T}})$  πλάτους  $w$ . Πιο συγκεκριμένα έχουμε ότι

$$\text{το } B \text{ είναι μία λύση ανν } B \in Q_{\min VC}^w(\mathcal{D}_{\mathcal{G}; \mathcal{I}}^{\text{ext}}).$$

Το πρόβλημα επιλύεται μετά από το πολύ  $(2^{3(w+1)} + 5 \cdot 2^{w+1} + w^2 + 5(w+1) + a) \cdot |T| + a$  στιγμιοτυποποιήσεις.

**Απόδειξη.** Ορίζουμε το  $\Pi_{C \min VC}^w = \Pi_{VC, w}^{\text{cond}} \cup P_{C \min VC}^w \cup S_{\min VC}^w \cup S_{\min VC}^w$ .

$$S_{\min VC}^w \cdot \quad s_I : \quad \text{min-suc-}Q^I(x, i) \leftarrow C\text{-min-VC}(i), \text{min-pot-}Q^I(x, i), \text{root}(x)$$

$$s_{S_1, S_2, L, I|U, P}^j : \quad \text{min-suc-}Q^{S_j}(x_j, i_j) \leftarrow \text{min-suc-}Q^I(x, i), \text{min-pot-}Q^{S_1}(x_1, i_1),$$

$$\text{min-pot-}Q^{S_2}(x_2, i_2), \text{succ}_0(x, x_1), \text{succ}_1(x, x_2),$$

$$\text{sum}(i, \ell, i_1, i_2), \text{"VC-condition-}S_1 S_2 LI|UP\text{"}$$

$$\text{για } j = 1, 2 \text{ \& } \ell = |L|$$

$$A_{\min VC}^w \cdot \quad a_{L, I} : \quad \text{assign-}Q^I(x, y) \leftarrow \text{min-suc-}Q^I(x, i), \text{leaf}(x), \text{"VC-condition-LI"},$$

$$\text{decomp-}L(x, y), \text{card}(y, i)$$

$$a_{J, S, L, I|U, P} : \quad \text{assign-}Q^I(x, y) \leftarrow \text{min-suc-}Q^I(x, i), \text{succ}_0(x, x_1), \text{succ}_1(x, x_2),$$

$$\text{assign-}Q^J(x_1, y_1), \text{assign-}Q^S(x_2, y_2),$$

$$\begin{aligned}
& \text{union}(y, v, y_1, y_2), \text{decomp-L}(x, v), \\
& \text{card}(y, i), \text{"VC-condition-JSLI|UP"} \\
r_I : & \quad \text{min-VertCov}(y) \leftarrow \text{assign-}Q^I(x, y), \text{root}(x)
\end{aligned}$$

όπου οι κανόνες  $s_I, a_{L,I}, r_I$  γράφονται για κάθε  $L, I \subseteq W$ , οι κανόνες  $a_{J,S,L,I|U,P}$  γράφονται για  $J, S, I, U, P \subseteq W$  τ.ω.  $|(J \cup S) \setminus L| = |U|$ ,  $U \cup P = I$  και  $U \cap P = \emptyset$ , και ανάλογοι περιορισμοί ισχύουν για τους κανόνες  $s_{S_1, S_2, L, I|U, P}^j$ . Επιπρόσθετα, το "VC-condition-LI" αποτελεί συντομογραφία για την ακολουθία ατόμων  $\text{set-L}(x)$ ,  $\neg \text{EdgeVpos} \setminus L(x)$ ,  $\text{AdjInParVpos} \setminus L-I(x)$ , και το "VC-condition-JSLI|UP" αποτελεί συντομογραφία για την ακολουθία ατόμων  $\text{set-L}(x)$ ,  $\neg \text{EdgeVpos} \setminus L(x)$ ,  $\{\neg \text{vertex-}j(x) \mid j \in (J \cup S) \setminus L\}$ ,  $\text{SameInPar-}(J \cup S)U(x)$ ,  $\text{AdjInParVpos} \setminus L-P(x)$ .

Αποδεικνύεται εύκολα ότι το  $Q_{\text{min VC}}^w = (\Pi_{\text{min VC}}^w, \text{min-VertCov})$  ικανοποιεί την επιθυμητή συνθήκη και ότι η αποτίμησή του πραγματοποιείται μετά από το πολύ  $(2^{3(w+1)} + 5 \cdot 2^{w+1} + w^2 + 5(w+1) + a) \cdot |T| + a$  στιγμιοτυποποιήσεις.  $\square$



## 4.

---

# Επίλογος

Με την παρούσα δουλειά, πραγματοποιήσαμε μία αναλυτική μελέτη των λύσεων που βασίζονται στη σχεσιακή άλγεβρα και στην datalog τα οποία επιλύουν προβλήματα αποτίμησης της MSO σε δομές φραγμένου δεντροπλάτους. Εκτός από τα κλασικά προβλήματα αποτίμησης, μελετήσαμε προβλήματα βελτιστοποίησης που ορίζονται μέσω τύπων της MSO στις αντίστοιχες δομές. Σε κάθε περίπτωση, κεντρικό πυρήνα της προσέγγισής μας αποτέλεσαν δομικά αποτελέσματα από τη θεωρία αυτομάτων τα οποία είναι η θεωρητική βάση της δουλειάς αυτής. Πιο συγκεκριμένα, βασικό μέρος της δουλειάς αποτελούν κατασκευές αυτομάτων που ουσιαστικά αντιστοιχούν σε Thatcher-Wright αυτόματα που αποδέχονται επεκτάσεις δεντροδιασπάσεων που κωδικοποιούν τις αναθέσεις που ικανοποιούν έναν MSO τύπο σε μία δεδομένη δομή. Μέσω των κατασκευών αυτών τα αρχικά προβλήματα εκφράζονται στη σχεσιακή άλγεβρα και στην datalog και επιλύονται με αλγόριθμους βελτιστοποίησης των βάσεων δεδομένων όπως ο αλγόριθμος του Yannakakis και οι μέθοδοι αναγραφής της datalog. Η προσέγγισή μας αποδεικνύει την ύπαρξη μιας FPT γενικής λύσης για τα προβλήματα αποτίμησης που ορίζονται από οποιονδήποτε MSO τύπο η οποία βασίζεται αποκλειστικά στην datalog. Ενώ μέσω των διαφορετικών αναγωγών των προβλημάτων αποτίμησης της MSO σε προβλήματα αποτίμησης της datalog που παρέχουμε, αποδεικνύουμε παράλληλα ενδιαφέροντα αποτελέσματα εκφρασιμότητας για την datalog και την MSO.

Μία ενδιαφέρουσα πρώτη κατεύθυνση για μελλοντική δουλειά αποτελεί η μελέτη υλοποιήσεων των προτεινόμενων προσεγγίσεων σε προγραμματιστικό περιβάλλον. Καθώς η προσέγγισή μας αφορά όλη την κλάση των MSO τύπων, η έκρηξη καταστάσεων του τελικού αυτομάτου αποτελεί γενικά αναπόφευκτο γεγονός [FG04]. Έτσι, η απαίτηση για χώρο στις πρακτικές υλοποιήσεις αυτής της προσέγγισης αποτελεί γενικά δομικό πρόβλημα. Μία πιθανή λύση στο πρόβλημα αυτό, αποτελεί ο προσδιορισμός κλάσεων τύπων της MSO που ενώ έχουν ενδιαφέρουσα εκφραστική δύναμη, οι αντίστοιχες κατασκευές των αυτομάτων είναι πρακτικά εφικτές, και κατά συνέπεια μπορούν να υπάρξουν αποτελεσματικές υλοποιήσεις για αυτές. Ανάλογα αποτελέσματα μπορεί να επιτευχθούν μέσω συστηματικών προσεγγίσεων για

ενδιαφέροντα προβλήματα που ορίζονται από συγκεκριμένους MSO τύπους. Στις περιπτώσεις αυτές, η απευθείας κατασκευή αυτομάτων, η βελτιστοποίηση της επαγωγικής κατασκευής τους αλλά και έξυπνες συμπαγείς αναπαραστάσεις των μεταβάσεων μπορούν να οδηγήσουν σε λύσεις για το πρόβλημα χώρου κατά την υλοποίηση των αντίστοιχων αλγορίθμων.

Οι υλοποιήσεις αυτές μπορούν να αποτελέσουν χρήσιμα εργαλεία τόσο στο πλαίσιο της XML όσο και σε αυτό της θεωρίας γραφημάτων. Επιπρόσθετα, η συγκριτική μελέτη των λύσεων που προκύπτουν μέσω αυτής της προσέγγισης για συγκεκριμένα προβλήματα, καθώς και η σύγκριση των αποτελεσμάτων αυτών με ανάλογα της υπάρχουσας βιβλιογραφίας θεωρούμε ότι θα δώσει ενδιαφέροντα συμπεράσματα.

Μελλοντικό στόχο αποτελεί επίσης η συστηματική μελέτη εκείνου του υποσυνόλου της datalog που εκφράζει τα προβλήματα που μελετήσαμε. Πιο συγκεκριμένα, μας ενδιαφέρει ο ακριβής προσδιορισμός του, η πιθανή περιγραφή του από συγκεκριμένους συντακτικούς παραμέτρους αλλά και η εκφραστική ικανότητα και η πολυπλοκότητα αυτού του τμήματος της datalog. Θεωρούμε ιδιαίτερα ενδιαφέρον θέμα τη σύγκριση του υποσυνόλου της datalog που περιγράφει τις λύσεις των προβλημάτων αποτίμησης της MSO σε δομές φραγμένου δεντροπλάτους που μελετήσαμε στα προηγούμενα κεφάλαια, με datalog προγράμματα που υλοποιούν αλγορίθμους για άλλα προβλήματα που ορίζονται πάνω σε ίδιου τύπου δομές. Επιπρόσθετα, η μελέτη των χαρακτηριστικών των αντίστοιχων datalog προγραμμάτων για συγκεκριμένους MSO τύπους και η σύγκρισή τους με υπάρχοντες αλγορίθμους δυναμικού προγραμματισμού της βιβλιογραφίας μπορεί να οδηγήσει σε χρήσιμα αποτελέσματα.

Ενδιαφέρουσα κατεύθυνση αποτελεί επίσης η προσπάθεια εφαρμογής της προσέγγισής μας για την επίλυση προβλημάτων αποτίμησης που ορίζονται μέσω της MSO και δεν μελετήσαμε στα προηγούμενα κεφάλαια, αλλά και για την επίλυση προβλημάτων αποτίμησης που ορίζονται από επεκτάσεις της MSO αλλά και από άλλες λογικές.

# Παραρτήματα



# Α΄.

## Επαγωγικοί ορισμοί αυτομάτων

Δίνονται αναλυτικά οι επαγωγικοί ορισμοί αυτομάτων για τους MSO τύπους.

**A. Σύζευξη.** Θεωρούμε αρχικά την περίπτωση της σύζευξης και υποθέτουμε ότι

$$\phi(\bar{X}, \bar{Y}, \bar{Z}) \equiv \phi_1(\bar{X}, \bar{Y}) \wedge \phi_2(\bar{X}, \bar{Z})$$

όπου κάθε τύπος  $\phi_i$ ,  $i = 1, 2$ , έχει  $k_i$  ελεύθερες μεταβλητές, δηλαδή θεωρούμε ότι τα μεγέθη των διανυσμάτων  $(\bar{X}, \bar{Y})$  και  $(\bar{X}, \bar{Z})$  ελευθέρων μεταβλητών είναι  $k_1$  και  $k_2$  αντίστοιχα, και ότι το διάνυσμα  $\bar{X}$  αντιστοιχεί στη λίστα με τις κοινές ελεύθερες μεταβλητές των τύπων  $\phi_1$  και  $\phi_2$ . Στη συνέχεια, συμβολίζουμε με  $\mathcal{A}_{\phi_i} = (\Gamma, Q_{k_i}, \Delta_0^i, \Delta^i, F_{k_i})$  το αυτόματο του τύπου  $\phi_i$ ,  $i = 1, 2$ , όπου τα  $Q_{k_i} = Q_i \times E^{k_i}$  και  $F_{k_i} = F_i \times E^{k_i}$  είναι τα σύνολα καταστάσεων και τελικών καταστάσεων αντίστοιχα.

Ας υποθέσουμε ότι ο τύπος  $\phi(\bar{X}, \bar{Y}, \bar{Z})$  έχει  $k$  ελεύθερες μεταβλητές. Τότε, το αυτόματο

$$\mathcal{A}_{\phi} = (\Gamma, Q_k, \Delta_0, \Delta, F_k)$$

του  $\phi$  με σύνολο καταστάσεων  $Q_k = Q \times E^k$  και σύνολο τελικών καταστάσεων  $F_k = F \times E^k$ , ορίζεται μέσω των αυτομάτων  $\mathcal{A}_{\phi_1}$  και  $\mathcal{A}_{\phi_2}$  ως εξής:

$$\begin{aligned} Q &= Q_1 \times Q_2 \\ \Delta_0 &= \{(\gamma, (q_1, q_2)\bar{e}_1\bar{e}_2) \mid \text{υπάρχουν } (\gamma, q_i\bar{e}_i) \in \Delta_0^i, i = 1, 2\} \\ \Delta &= \{((p_1, p_2), (s_1, s_2), \gamma, (q_1, q_2)\bar{e}_1\bar{e}_2) \mid \text{υπάρχουν } (p_i, s_i, \gamma, q_i\bar{e}_i) \in \Delta^i, i = 1, 2\} \\ F &= F_1 \times F_2. \end{aligned}$$

**B. Υπαρξιακή ποσοδείκτηση.** Ας υποθέσουμε ότι

$$\phi(\bar{X}) \equiv \exists Y \psi(\bar{X}, Y)$$

όπου  $\bar{X} = (X_1, \dots, X_k)$  και  $(\bar{X}, Y) = (X_1, \dots, X_k, Y)$ . Και ας θεωρήσουμε ότι

$$\mathcal{A}_\psi = (\Gamma, Q_{k+1}, \Delta_0^\psi, \Delta^\psi, F_{k+1})$$

όπου το  $Q_{k+1} = Q_\psi \times E^{k+1}$  είναι το σύνολο καταστάσεων και το  $F_{k+1} = F_\psi \times E^{k+1}$  το σύνολο των τελικών καταστάσεων. Διακρίνουμε τις παρακάτω δύο περιπτώσεις.

*Περίπτωση 1.* Η κατασκευή του  $\mathcal{A}_\phi$  αποτελεί ενδιάμεσο βήμα της κατασκευής ενός αυτομάτου  $\mathcal{A}_\varphi$ , δηλαδή ο  $\phi$  αποτελεί υποτύπο ενός τύπου  $\varphi$  για τον οποίο θέλουμε να κατασκευάσουμε αυτόματο, και ισχύει ότι ο  $\phi$  εμφανίζεται με άρνηση στον  $\varphi$  ή η άρνηση εφαρμόζεται σε έναν υποτύπο του  $\varphi$  στον οποίο εμφανίζεται ο  $\phi$ . Στην περίπτωση αυτή, το αυτόματο

$$\mathcal{A}_\phi = (\Gamma, Q_k, \Delta_0, \Delta, F_k)$$

ορίζεται με τη βοήθεια του αυτομάτου  $\mathcal{A}_\psi$  σε δύο βήματα:

*Βήμα α'.* Ορίζουμε αρχικά το αυτόματο  $\mathcal{A}'_\phi = (\Gamma, Q'_k, \Delta'_0, \Delta', F'_k)$  μέσω του αυτομάτου  $\mathcal{A}_\psi$ .

$$\begin{aligned} Q'_k &= Q_\psi \times E^k \\ \Delta'_0 &= \{(\gamma, q\bar{e}) \mid \text{υπάρχουν } (\gamma, q\bar{e}e) \in \Delta_0^\psi\} \\ \Delta' &= \{(q_1, q_2, \gamma, q\bar{e}) \mid \text{υπάρχουν } (q_1, q_2, \gamma, q\bar{e}e) \in \Delta^\psi\} \\ F'_k &= F_\psi \times E^k. \end{aligned}$$

*Βήμα β'.* Στη συνέχεια, εφαρμόζουμε μία μορφή «καταστευής δυναμοσυνόλου» στο  $\mathcal{A}'_\phi$  ώστε να προκύψει το τελικό αυτόματο  $\mathcal{A}_\phi = (\Gamma, Q_k, \Delta_0, \Delta, F_k)$  που ορίζεται παρακάτω. Σημειώνουμε ότι  $\mathcal{P}(Q_\psi) = \{Q \mid Q \subseteq Q_\psi\}$ .

$$\begin{aligned} Q_k &= \mathcal{P}(Q_\psi) \times E^k \\ \Delta_0 &= \bigcup_{(\gamma, \bar{e}) \in \Gamma \times E^k} \{(\gamma, Q\bar{e}) \mid Q = \{q \mid (\gamma, q\bar{e}) \in \Delta'_0\}\} \\ \Delta &= \bigcup_{(Q_1, Q_2, \gamma, \bar{e}) \in (\mathcal{P}(Q_\psi))^2 \times \Gamma \times E^k} \{(Q_1, Q_2, \gamma, Q\bar{e}) \mid Q = \{q \mid \text{υπάρχουν } q_1 \in Q_1, \\ &\quad q_2 \in Q_2 \text{ τ.ω. } (q_1, q_2, \gamma, q\bar{e}) \in \Delta'\}\} \\ F_k &= \{Q \in \mathcal{P}(Q_\psi) \mid Q \cap F_\psi \neq \emptyset\} \times E^k. \end{aligned}$$

*Περίπτωση 2.* Είτε μας ενδιαφέρει αποκλειστικά η κατασκευή του  $\mathcal{A}_\phi$  ή η κατασκευή του  $\mathcal{A}_\phi$  αποτελεί ενδιάμεσο βήμα της κατασκευής ενός αυτομάτου  $\mathcal{A}_\varphi$  χωρίς όμως να ισχύουν οι συνθήκες της παραπάνω περίπτωσης. Στην περίπτωση αυτή, το αυτόματο  $\mathcal{A}_\phi = (\Gamma, Q_k, \Delta_0, \Delta, F_k)$  συμπίπτει με το αυτόματο  $\mathcal{A}'_\phi$  του Βήματος (α') της Περίπτωσης 1.

**Γ. Άρνηση.** Ας υποθέσουμε τώρα ότι

$$\phi(\bar{X}) \equiv \neg\psi(\bar{X})$$

---

όπου  $\bar{X} = (X_1, \dots, X_k)$ . Και ως θεωρήσουμε ότι  $\mathcal{A}_\psi = (\Gamma, Q_k^\psi, \Delta_0^\psi, \Delta^\psi, F_k^\psi)$ . Στην περίπτωση αυτή, το αυτόματο

$$\mathcal{A}_\phi = (\Gamma, Q_k, \Delta_0, \Delta, F_k)$$

ορίζεται από το αυτόματο  $\mathcal{A}_\psi$  ως εξής:

$$\begin{aligned} Q_k &= Q_k^\psi \\ \Delta_0 &= \Delta_0^\psi \\ \Delta &= \Delta^\psi \\ F_k &= Q_k^\psi \setminus F_k^\psi. \end{aligned}$$





**B'.**

## Ο αλγόριθμος των Flum Frick Grohe

Δίνουμε μία σύντομη περιγραφή του FFG (Flum Frick Grohe) αλγορίθμου [FFG02] ο οποίος υπολογίζει τις αναθέσεις που ικανοποιούν έναν MSO τύπο  $\phi(X_1, \dots, X_k)$  σε ένα  $\Gamma$ -χρωματισμένο δέντρο  $\mathcal{T}$ . Ο αλγόριθμος χρησιμοποιεί ως είσοδο τη μετάφραση του τύπου  $\phi$  σε ένα ντετερμινιστικό bottom-up  $\Gamma_k$ -αυτόματο,  $\Gamma_k = \Gamma \times \{0, 1\}^k$ ,

$$A_\phi = (Q, \delta_0, \delta, F)$$

που αναγνωρίζει την κλάση δέντρων  $\mathcal{T}_{\Gamma, \phi}$  που αποτελείται από τα  $\Gamma_k$ -δέντρα που είναι  $\phi$ -επεκτάσεις του  $\mathcal{T}$ , δηλαδή αποτελούν επεκτάσεις του  $\mathcal{T}$  με αναθέσεις που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$ . Δεδομένου του  $A_\phi$  και ενός  $\Gamma$ -δέντρου  $\mathcal{T} = ((T, s_1, s_2), c)$ , ο FFG αλγόριθμος υπολογίζει τις υποκείμενες αναθέσεις των  $\phi$ -επεκτάσεων του  $\mathcal{T}$ : αυτές είναι ακριβώς οι αναθέσεις που ικανοποιούν τον  $\phi$  στο  $\mathcal{T}$ . Ο αλγόριθμος πραγματοποιεί τρεις διασχίσεις στο δέντρο εισόδου  $\mathcal{T}$  σε ισάριθμα βήματα

**Βήμα I.** (*Bottom-Up*) Υπολογισμός των πιθανών καταστάσεων σε κάθε κόμβο  $n \in T$ . Με κατεύθυνση από τα φύλλα προς τη ρίζα υπολογίζονται:

- αν ο  $n$  είναι φύλλο:  $\text{pot-states}(n) = \{\delta_0((c(n), \bar{\epsilon})) \mid \bar{\epsilon} \in \{0, 1\}^k\}$
- αν ο  $n$  είναι εσωτερικός κόμβος τ.ω.  $s_i(n) = n_i, i = 1, 2$

$$\text{pot-states}(n) = \{\delta(q_1, q_2, (c(n), \bar{\epsilon})) \mid q_1 \in \text{pot-states}(n_1), \\ q_2 \in \text{pot-states}(n_2), \bar{\epsilon} \in \{0, 1\}^k\}.$$

*Παρατήρηση.* Όταν  $\text{pot-states}(r) \cap F = \emptyset$ , όπου  $r$  είναι η ρίζα, τότε ο αλγόριθμος τερματίζει:

$$\text{sat-assign}(\phi, \mathcal{T}) = \emptyset.$$

**Βήμα II.** (*Top-Down*) Υπολογισμός των επιτυχημένων καταστάσεων σε κάθε κόμβο  $n \in T$ . Με κατεύθυνση από τη ρίζα προς τα φύλλα υπολογίζονται:

- αν  $r$  είναι η ρίζα του δέντρου  $T$ :  $\text{succ-states}(r) = \text{pot-states}(r) \cap F$ .
- αν ο  $n_1$  είναι το αριστερό παιδί του  $n$  και το  $n_2$  είναι το δεξί παιδί του  $n$ , τότε:

$$\text{succ-states}(n_1) = \{q_1 \in \text{pot-states}(n_1) \mid \exists q_2 \in \text{pot-states}(n_2), \exists \bar{\epsilon} \in \{0, 1\}^k \\ \text{τ.ω. } \delta(q_1, q_2, (c(n), \bar{\epsilon})) \in \text{succ\_states}(n)\}.$$

Μόλις υπολογιστεί το  $\text{succ\_states}(n_1)$ , υπολογίζουμε το  $\text{succ-states}(n_2)$  με εντελώς συμμετρικό τρόπο.

**Βήμα III.** (*Bottom-Up*) Υπολογισμός των αναθέσεων που ικανοποιούν τον  $\phi$  στο  $T$ . Θεωρούμε ότι  $\text{assign}(n, \bar{\epsilon}) = ([n]^{\epsilon_1}, \dots, [n]^{\epsilon_k})$ , όπου  $[n]^0 = \emptyset$ ,  $[n]^1 = \{n\}$  και  $\bar{\epsilon} = (\epsilon_1, \dots, \epsilon_k)$ . Με κατεύθυνση από τα φύλλα προς τη ρίζα έχουμε τους υπολογισμούς:

- αν ο  $n$  είναι φύλλο και  $q \in \text{succ-states}(n)$ , τότε:

$$\text{sat-assign}(n, q) = \{\text{assign}(n, \bar{\epsilon}) \mid \delta_0(c(n), \bar{\epsilon}) = q\}.$$

- αν ο  $n$  είναι εσωτερικός κόμβος τ.ω.  $q \in \text{succ-states}(n)$ :

$$\text{sat-assign}(n, q) = \{\bar{A}_1 \cup \bar{A}_2 \cup \text{assign}(n, \bar{\epsilon}) \mid \exists q_i \in \text{succ-states}(n_i), i = 1, 2, \text{ τ.ω.} \\ \delta(q_1, q_2, (c(n), \bar{\epsilon})) = q \text{ και } \bar{A}_1 \in \text{sat-assign}(n_1, q_1), \\ \bar{A}_2 \in \text{sat\_assign}(n_2, q_2)\}.$$

Αν  $r$  είναι η ρίζα του δέντρου  $T$ , τότε:

$$\text{sat-assign}(\phi, T) = \bigcup_{q \in \text{succ-states}(r)} \text{sat-assign}(r, q).$$

Γ'.

## Αυτόματα Αποδόμησης

Πίνακας Γ'.1: Οι μεταβάσεις του  $w\text{-dec-Assign}_{\neg\text{Edge}}$ .

$q_1, q_2, (\gamma, \bar{e}) \mapsto q, j, \ell \in W, I, J \subseteq W$		
Μετάβαση		Συνθήκες που ικανοποιούνται
$q_0, q_0, \gamma, (\emptyset, \emptyset)$	$\mapsto q_0$	
$q_0, q_0, \gamma, (\{j\}, \emptyset)$	$\mapsto q_1^I$	$vertex(j) \ \& \ AdjInPar(j) = I$
$q_0, q_0, \gamma, (\emptyset, \{j\})$	$\mapsto q_2^I$	
$q_0, q_0, \gamma, (\{j\}, \{\ell\})$	$\mapsto q_a$	$NotEdge(j, \ell)$
$q_0, q_1^J, \gamma, (\emptyset, \emptyset)$	$\mapsto q_1^I$	$SameInPar(J) = I$
$q_1^J, q_0, \gamma, (\emptyset, \emptyset)$		
$q_0, q_2^J, \gamma, (\emptyset, \emptyset)$	$\mapsto q_2^I$	
$q_2^J, q_0, \gamma, (\emptyset, \emptyset)$		
$q_0, q_1^I, \gamma, (\emptyset, \{j\})$	$\mapsto q_a$	$vertex(j) \ \& \ j \notin I$
$q_1^I, q_0, \gamma, (\emptyset, \{j\})$		
$q_0, q_2^I, \gamma, (\{j\}, \emptyset)$		
$q_2^I, q_0, \gamma, (\{j\}, \emptyset)$		
$q_0, q_a, \gamma, (\emptyset, \emptyset)$	$\mapsto q_a$	
$q_a, q_0, \gamma, (\emptyset, \emptyset)$		
$q_1, q_2, \gamma, \bar{e}$	$\mapsto q_f$	όλες οι άλλες περιπτώσεις

Πίνακας Γ'.2: Οι μεταβάσεις του  $w\text{-dec-Assign}_{\neg \ln(x,X)}$ .

$q_1, q_2, (\gamma, \bar{I}) \mapsto q, i \in W, I \subseteq W, Q = \{q_0, q_a, q_f\}$	
Μετάβαση	Συνθήκες που ικανοποιούνται στο $\gamma \in \Gamma_{\tau_G, w}$
$q_0, q_0, \gamma, (\emptyset, \emptyset) \mapsto q_0$	
$q_0, q_0, \gamma, (\emptyset, I) \mapsto q_0$	$set(I)$
$q_0, q_0, (\{i\}, \emptyset) \mapsto q_a$	$vertex(i)$
$q_0, q_0, \gamma, (\{i\}, I) \mapsto q_a$	$vertex(i) \ \& \ set(I) \ \& \ i \notin I$
$q_0, q_a, \gamma, (\emptyset, \emptyset) \mapsto q_a$	
$q_a, q_0, \gamma, (\emptyset, \emptyset) \mapsto q_a$	
$q_0, q_a, \gamma, (\emptyset, I) \mapsto q_a$	$set(I)$
$q_a, q_0, \gamma, (\emptyset, I) \mapsto q_a$	
$q_1, q_2, \gamma, \bar{I} \mapsto q_f$	όλες οι άλλες περιπτώσεις

Πίνακας Γ'.3: Οι μεταβάσεις του  $w\text{-dec-Assign}_{\text{Element}^1}$ .

$q_1, q_2, (\gamma, \bar{e}) \mapsto q, i \in W, Q = \{q_0, q_a, q_f\}$	
Μετάβαση	Συνθήκες που ικανοποιούνται στο $\gamma \in \Gamma_{\tau_G, w}$
$q_0, q_0, \gamma, \emptyset \mapsto q_0$	
$q_0, q_0, \gamma, \{i\} \mapsto q_a$	$vertex(i)$
$q_0, q_a, \gamma, \emptyset \mapsto q_a$	
$q_a, q_0, \gamma, \emptyset \mapsto q_a$	
$q_1, q_2, \gamma, I \mapsto q_f$	όλες οι άλλες περιπτώσεις

Πίνακας Γ'.4: Οι μεταβάσεις του  $w\text{-dec-Assign}_{\text{Set}^1}$ .

$q_1, q_2, (\gamma, \bar{e}) \mapsto q, I \subseteq W, Q = \{q_0, q_a, q_f\}$	
Μετάβαση	Συνθήκες που ικανοποιούνται στο $\gamma \in \Gamma_{\tau_G, w}$
$q_0, q_0, \gamma, \emptyset \mapsto q_0$	
$q_0, q_0, \gamma, I \mapsto q_a$	$set(I)$
$q_0, q_a, \gamma, \emptyset \mapsto q_a$	
$q_a, q_0, \gamma, \emptyset \mapsto q_a$	
$q_0, q_a, \gamma, I \mapsto q_a$	$set(I)$
$q_a, q_0, \gamma, I \mapsto q_a$	
$q_a, q_a, \gamma, \emptyset \mapsto q_a$	
$q_a, q_a, \gamma, I \mapsto q_a$	$set(I)$
$q_1, q_2, \gamma, I \mapsto q_f$	όλες οι άλλες περιπτώσεις

## Ευρετήριο Συμβόλων

---

$AdjInPar_{\gamma \setminus J}$ , 125	$Dec_w(T, A)$ , 92
$AdjInPar(j)$ , 125	$D_\gamma$ , 31
$Adj(j)$ , 125	$\mathcal{D}_L$ , 107
$A_\phi$ , 26	dom, 33
$\mathcal{A}_{\phi,w}$ , 96	$\mathcal{D}_T$ , 58
$\overline{A}_n$ , 99	$\mathcal{D}_T$ , 67
ar( $\Pi$ ), 56	$\mathcal{D}_T^{\text{pre}}$ , 58
Assign $_\phi$ , 27	$e$ , 31
a-trans $_n$ , 44	$\bar{e}$ , 31
$\overline{B}_\varepsilon$ , 22	$\varepsilon_{\overline{B}}$ , 22, 93
$c$ , 18, 94	EDB( $\Pi$ ), 54
$c_A$ , 94	$Edge_{\gamma \setminus J}$ , 125
Const( $\Pi$ ), 54	$Edge(j, \ell)$ , 125
$\mathcal{C}_T$ , 20	$\mathcal{E}_{k,w}(T)$ , 92
$D$ , 31	$element(j)$ , 100
$\Delta$ , 20	enc $_k$ , 22
$\delta$ , 20	enc $_{k,w}$ , 92
$D_0$ , 31	$\varepsilon_o$ , 22
$\Delta_0$ , 20	$E_T$ , 40
$\delta_0$ , 20	$E_{T_n}$ , 40
$D'_0$ , 31	Eval( $L, C$ ), 2
$D'$ , 31	$E(X, Z)$ , 19
$\mathcal{D}_A$ , 107	$F$ , 20
$\mathcal{D}_{A;\mathcal{I}}^{\text{ext}}$ , 122	$\phi(\mathcal{A})$ , 2
$\mathcal{D}_{A;\mathcal{I}}^U$ , 114	$\phi_{\text{enc}}(\mathcal{T})$ , 22
$\mathcal{D}_A^{\text{sum}}$ , 121	$F_k$ , 23
dec $_k$ , 22	$F_{k,w}$ , 96
dec $_{k,w}$ , 93	$\phi(\mathcal{T})$ , 20
dec $_{u,v,w}$ , 98	$\Gamma$ , 18

- $g_{\phi, \mathcal{T}}$ , 39  
 $G_{\mathcal{T}}$ , 66  
 $\mathcal{G}_t$ , 38  
 $\Gamma_{\tau, w}$ , 94  
 $\Gamma_{\tau, w, A}$ , 94  
 $\mathcal{I}; \mathcal{A}$ , 94  
 $\text{IDB}(\Pi)$ , 54  
 $I_\epsilon$ , 91  
 $i_{\phi, \mathcal{T}}$ , 39  
 $I_n^i$ , 93  
 $\text{itrans}_n$ , 38  
 $\text{MC}(L, \mathcal{C})$ , 1  
 $\text{MSO}[\tau_{\Gamma}]$ , 19  
 $\text{node}(a)$ , 92  
 $\text{NotEdge}_{\gamma \setminus J}$ , 125  
 $\text{pair}(a)$ , 92  
 $\Pi_{c \min \phi}^w$ , 121  
 $\Pi_{c \min \phi}^{w, m}$ , 120  
 $\Pi(\mathcal{D})$ , 55  
 $(\Pi, \mathcal{D}) \models$ , 55  
 $\Pi_\phi$ , 75  
 $\Pi'_\phi$ , 72  
 $\Pi_\phi^1$ , 73  
 $\Pi_\phi^\epsilon$ , 75  
 $\Pi_\phi^k$ , 67  
 $\Pi_{\phi, k}^w$ , 114  
 $\Pi_{\phi, m}$ , 58  
 $\Pi_{\phi, m}^{\text{mon}}$ , 69  
 $\Pi_{\phi, m}^R$ , 61  
 $\Pi_{\phi, m}^S$ , 63  
 $\Pi_\phi^w$ , 109  
 $P_\gamma(X)$ , 19  
 $\Pi_{\min \phi}^w$ , 122  
 $\text{Pos}(n)$ , 92  
 $(\Pi, Q)$ , 55  
 $\text{Pre}_t$ , 58  
 $\text{p-trans}_n$ , 42  
 $\Pi_{\tau, w}$ , 108  
 $\pi_X(r)$ , 34  
 $\Pi_\psi^{\exists \leq k, w}$ , 117  
 $Q$ , 20, 31, 105  
 $Q_{c \min \phi}^w$ , 121  
 $Q(\mathcal{D})$ , 55  
 $Q(d)$ , 34  
 $q\bar{e}$ , 23  
 $Q_\phi$ , 73  
 $Q_\phi^k$ , 67  
 $Q_{\phi, k}^w$ , 114  
 $Q_{\phi, m}$ , 58  
 $Q_{\phi, m}^{\epsilon, j}$ , 69  
 $Q_{\phi, m}^R$ , 61  
 $Q_{\phi, m}^S$ , 63  
 $Q_\phi^w$ , 109  
 $q\bar{I}$ , 97  
 $q_j \bar{e}_j$ , 23  
 $Q_k$ , 23  
 $Q_{k, w}$ , 96  
 $Q_{\min \phi}^w$ , 122  
 $Q_t$ , 40  
 $Q_\psi^{\exists \leq k, w}$ , 117  
 $\rho$ , 20  
 $r_1 \boxtimes r_2$ , 34  
 $r_{\phi, \mathcal{T}}$ , 40  
 $\text{RInPar}(j)$ , 100  
 $R(j)$ , 100  
 $\mathcal{R}_t$ , 40  
 $R[X]$ , 33  
 $\text{SameInPar}(J)$ , 100, 125  
 $\text{set}(I)$ , 100, 125  
 $s_{\phi, \mathcal{T}}$ , 40  
 $s_i$ , 18  
 $S_i(X, Y)$ , 19  
 $\text{sort}$ , 33  
 $\text{s-trans}_n$ , 42  
 $\mathcal{T}$ , 18  
 $t$ , 18

$T$ , 18  
 $\tau_\Gamma$ , 18  
 $\mathcal{T}_{\mathcal{I};\mathcal{A}}$ , 94  
 $\mathcal{T}_n$ , 18  
 $T_n$ , 18  
 $\text{tp}_q^{\text{MSO}}(\mathcal{A})$ , 4  
 $\text{tp}_q^{\text{MSO}}(\mathcal{A}, \bar{a})$ , 4  
 $\text{trans}_n$ , 38  
 $\mathcal{T}_{\tau,w}$ , 95  
 $t[X]$ , 33  
  
 $\text{vertex}(j)$ , 125  
 $V_\gamma$ , 125  
 $V_n$ , 127  
  
 $W$ , 92  
 $w\text{-}D$ , 105  
 $w\text{-}\Delta$ , 96, 98  
 $w\text{-}\delta$ , 98  
 $w\text{-}D_0$ , 105  
 $w\text{-}\Delta_0$ , 96, 98  
 $w\text{-}\delta_0$ , 98  
 $w\text{-}D'_0$ , 105  
 $w\text{-}D_0^c$ , 105  
 $w\text{-}D'$ , 105  
 $w\text{-}D^c$ , 105  
 $w\text{-dec-}A_\phi$ , 98  
 $w\text{-dec-Assign}_\phi$ , 96  
 $w\text{-dec-Assign}_\phi^{\text{count}}$ , 98  
 $w\text{-dec-Assign}_R$ , 103  
 $w\text{-dec-Assign}_{\text{In}}$ , 102  
 $w\text{-dec-}A_{\text{Element}^u}$ , 105  
 $w\text{-dec-}A_{\text{Set}^k}$ , 104  
  
 $Y_n$ , 38  
 $Y_T$ , 40  
 $Y_{T_n}$ , 40  
  
 $Z_n$ , 127





# Ευρετήριο Όρων

---

- datalog, 54
  - κανόνας, 54
  - κεφαλή, 54
  - στιγμιοτυποποίηση, 54
  - σώμα, 54
  - με άρνηση, 57
  - πρόγραμμα, 54
  - αποτίμηση, 55
  - βαθμός, 55
  - διαστρωματωμένο, 57
  - ημιθετικό, 106
  - μοναδιαίο, 55
  - στρώμα, 57
  - σχήμα, 54
- MSO, 19
  - τύπος, 19
  - ατομικός, 19
- άτομο, 54
- αλγόριθμος
  - FFG, 46
  - FPT, 2
  - Yannakakis, 35
- ανάθεση, 22
  - κωδικοποιημένη, 22
- απεικόνιση
  - datalog ορίσιμη, 57, 106
  - ανάθεσης, 22, 93
  - αποκωδικοποίησης, 93
  - κωδικοποίησης, 92
- αυτόματο
  - αναθέσεων, 23
    - για τον  $\phi$ , 27
    - υπολογίζει, 24
  - αποδόμησης  $w$ -dec-Assign<sub>In</sub>, 102
  - αποδόμησης  $w$ -dec-Assign<sub>R</sub>, 103
  - αποδόμησης  $w$ -dec-Assign<sub>Element<sup>k</sup></sub>, 105
  - αποδόμησης  $w$ -dec-Assign<sub>Set<sup>k</sup></sub>, 104
  - αποδόμησης για τον  $\phi$ , 96
    - εκτέλεση, 97
    - με μέτρηση, 98
    - υπολογίζει, 97
  - δέντρου
    - μη ντετερμινιστικό, 20
    - ντετερμινιστικό, 20
- βάση δεδομένων
  - datalog, 54
  - είσοδος, 54
  - πεδίο ορισμού, 54
  - στο σχεσιακό μοντέλο, 34
  - ακυκλικό σχήμα, 35
  - ισοδύναμη, 35
  - καθολικά συνεπής, 35
  - σχήμα, 33
- γεγονός, 54
  - συμπεραίνεται ή υπολογίζεται, 55
- δέντρο
  - αναγνωρίζεται, 20
  - δομής, 94
  - υποκείμενο, 18
  - χρωματισμένο, 18

- δεντροδιάσπαση, 87
  - ειδική, 87
  - πλάτος, 87
  - σάκος, 87
- δεντροπλάτος, 3
- διάνυσμα
  - ανάθεσης, 22, 24
  - καταστάσεων, 24
- διάσχιση
  - προδιατεταγμένη, 19
- δομή, 18
  - δεντροπλάτος, 87
  - πεπερασμένη, 18
- εκτέλεση, 20
  - επιτυχημένη, 20
  - μέρος
    - ανάθεσης, 23, 97
    - καταστάσεων, 23, 97
- εξάρτηση
  - δομική, 34
  - συναρτησιακή, 34
- ερώτημα
  - datalog, 55
    - αποτίμηση, 56
    - βαθμός, 55
    - ισοδύναμο, 55
    - μοναδιαίο, 55
    - στόχος, 55
  - min-MSO-ορίσιμο, 88
  - MSO ή MSO-ορίσιμο, 21
    - boolean, 21
    - επιλογής κόμβων, 21
    - μονομελές, 21
  - σχεσιακής άλγεβρας, 34
    - ακυκλικό συζευκτικό, 36
    - αποτίμηση, 34
- θεώρημα
  - Thatcher & Wright, 20
- κάλυμμα κορυφών, 123
  - ελάχιστο, 123
- κατάσταση
  - έγκυρη, 32
  - επιτυχημένη, 32
  - πιθανή, 32
  - τελική, 20
- κατηγορήμα
  - IDB ή προθεσιακό, 54
  - EDB ή εκτασιακό, 54
  - φίλτρου, 56
- κλάση δέντρων
  - αναγνωρίσιμη, 20
  - ορίσιμη, 20
- κωδικοποίηση
  - ανάθεσης, 93
- κόμβος, 18
  - αντιπροσωπευτικός, 92
- μετάβαση, 31, 105
  - έγκυρη, 32
  - επιτυχημένη, 32
  - πιθανή, 32
- πλειάδα, 33
- προβολή, 34
- πρόβλημα
  - FPT, 2
  - $k$ -VERTEX COVER, 134
  - MINIMUM CARDINALITY VERTEX COVER, 136
  - MINIMUM VERTEX COVER, 137
  - VERTEX COVER, 133
  - αποτίμησης, 2
    - datalog ερώτημα, 56
    - datalog πρόγραμμα, 55
    - max-card-MSO, 88
    - max-MSO, 88
    - min-card-MSO, 88
    - min-MSO, 88
    - MSO ερωτημάτων σε δέντρα, 21
    - ερωτήματος της άλγεβρας, 35
    - της MSO σε δέντρα, 21
  - βελτιστοποίησης MSO-ορίσιμο, 88
  - ελέγχου μοντέλου, 1
    - $k$ -card-MSO, 88

- ερωτήματος της άλγεβρας, 35
- της MSO σε δέντρα, 21
- στιγμιότυποποίηση
  - άσκοπη, 56
  - ενεργή, 56
  - συναφής, 56
- συνάρτηση
  - αποκωδικοποίησης, 22
  - κωδικοποίησης, 22
  - μετάβασης, 20
  - στοιχειώδης, 3
  - χρωματισμού, 18
- σχέση
  - μετάβασης, 20
  - στο σχεσιακό μοντέλο, 33
    - γνώρισμα, 33
    - σχήμα, 33
    - όνομα, 33
- σύμβολο κατηγορήματος, 18
  - βαθμός, 18
- υπογραφή, 18
- υποδέντρο, 18
  - πλήρες, 19
- υποεκτέλεση, 23
- υποστιγμιότυπο, 35
- φίλτρο, 56
  - βέλτιστο, 57
- φιλτράρισμα επίλυσης, 11, 56
- φυσική συνένωση, 34



## Βιβλιογραφία

---

- [AHV95] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
- [ALS91] S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12:308–340, 1991.
- [Bid91] N. Bidoit. Negation in rule-based database languages: a survey. *Theoretical Computer Science*, 78(1):3–83, 1991.
- [BMSU86] F. Bancilhon, D. Maier, Y. Sagiv, and J.D. Ullman. Magic sets and other strange ways to implement logic programs. In *Proc. 5th ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, pages 1–15. ACM Press, 1986.
- [Bod93] Hans L. Bodlaender. A tourist guide through treewidth. *Acta Cybernetica*, 11:1–23, 1993.
- [Bod96] H.L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25:1305–1317, 1996.
- [Bod07] Hans L. Bodlaender. Treewidth: structure and algorithms. In *Proceedings of the 14th international conference on Structural information and communication complexity, SIROCCO'07*, pages 11–25, 2007.
- [CD10a] B. Courcelle and I. Durand. Tractable constructions of finite automata from monadic second-order formula. In *Workshop on Logical Approaches to Barriers in Computing and Complexity*, February 2010.
- [CD10b] B. Courcelle and I. Durand. Verifying monadic second order graph properties with tree automata. In *European LISP Symposium in Portugal*, May 2010.
- [CE] B. Courcelle and J. Engelfriet. *Graph structure and monadic second-order logic, a language theoretic approach*. Book to be published by Cambridge University Press.
- [Cou90] B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 193–242. Elsevier Science Publishers, 1990.

- [DF99] R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.
- [Don70] J.E. Doner. Tree acceptors and some of their applications. *Journal of Computer and System Sciences*, 4:406–451, 1970.
- [FFG02] J. Flum, M. Frick, and M. Grohe. Query evaluation via tree decompositions. *Journal of the ACM*, 49(6):716–752, 2002.
- [FG04] M. Frick and M. Grohe. The complexity of first-order and monadic second-order logic revisited. *Annals of Pure and Applied Logic*, 130(1–3):3–31, 2004.
- [FG06] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer-Verlag, 2006.
- [FGK03] M. Frick, M. Grohe, and C. Koch. Query evaluation on compressed trees. In *Proc. 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 188–197, June 2003.
- [FK05] E. Foustoucos and L. Kalantzi. Mso querying over trees via datalog evaluation. In *Proc. 5th Panhellenic Logic Symposium*, pages 68–77, July 2005.
- [FK09a] E. Foustoucos and L. Kalantzi. Database algorithms solving the monadic second-order logic evaluation problem on finite trees. In *Proc. 4th Athens Colloquium on Algorithms and Complexity (ACAC 2009)*, August 2009.
- [FK09b] E. Foustoucos and L. Kalantzi. k-ary queries over trees via k-ary datalog. In *Proc. 7th Panhellenic Logic Symposium*, July 2009.
- [FK09c] E. Foustoucos and L. Kalantzi. The monadic second-order logic evaluation problem on finite colored trees: a database-theoretic approach. *Fundamenta Informaticae*, 92(3):193–231, 2009.
- [FK10] E. Foustoucos and L. Kalantzi. Automata-theoretic and datalog-based solutions of monadic second-order logic evaluation problems over structures of bounded-treewidth. Extended abstract presented at the *2nd Workshop on Graph Decompositions: Theoretical, Algorithmic and Logical Aspects*, October 2010.
- [FK11a] E. Foustoucos and L. Kalantzi. Automata-theoretic and datalog-based solutions of monadic second-order logic evaluation problems over structures of bounded-treewidth. Technical Report, University of Patras, February 2011.
- [FK11b] E. Foustoucos and L. Kalantzi. Automata-theoretic and datalog-based solutions of monadic second-order logic evaluation problems over structures of bounded-treewidth. In *Proc. 8th Panhellenic Logic Symposium*, July 2011.
- [Flu01] J. Flum. Tree-decompositions and the model-checking problem. *Bulletin of the EATCS*, pages 78–98, 2001.

- 
- [GK04] G. Gottlob and C. Koch. Monadic datalog and the expressive power of web information extraction languages. *Journal of the ACM*, 51(1):74–113, 2004.
- [GPW10] G. Gottlob, R. Pichler, and F. Wei. Monadic datalog over finite structures with bounded treewidth. *ACM Trans. Comput. Log.*, 12(1):3, 2010.
- [Gro07] M. Grohe. Logic, graphs, and algorithms. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(091), 2007.
- [GS03] M. Grohe and N. Schweikardt. Comparing the succinctness of monadic query languages over finite trees. In *Proc. 12th Annual Conference of the European Association for Computer Science Logic (CSL'03)*, volume 2803 of Lecture Notes in Computer Science, pages 226–240. Springer, 2003.
- [Kre09] Stephan Kreutzer. Algorithmic meta-theorems. *Electronic Colloquium on Computational Complexity (ECCC)*, 16, 2009.
- [Mak04] J. A. Makowsky. Algorithmic uses of the feferman-vaught theorem. *Annals of Pure and Applied Logic*, 126(1–3):59–213, 2004.
- [Nie06] R. Niedermeier. *Invitation to Fixed Parameter Algorithms*. Oxford University Press, 2006.
- [NS02] F. Neven and T. Schwentick. Query automata over finite trees. *Theor. Comput. Sci.*, 275(1-2):633–674, 2002.
- [PRW10] R. Pichler, S. Rummele, and S. Woltran. Counting and enumeration problems with bounded treewidth. In *Logic for Programming, Artificial Intelligence, and Reasoning, 15th International Conference, LPAR-16, Dakar, Senegal, April 25 - May 1, 2010. Proceedings*, pages 387–404, 2010.
- [Rei02] K. Reinhardt. The complexity of translating logic to finite automata. In E. Gradel, W. Thomas, and Th. Wilke, editors, *Automata, Logics, and Infinite Games*, volume 2500 of Lecture Notes in Computer Science, pages 235–242. Springer-Verlag, 2002.
- [RS86] N. Robertson and P. D. Seymour. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7:309–322, 1986.
- [Sto74] L.J. Stockmeyer. *The Complexity of Decision Problems in Automata Theory*. Ph.D. Thesis, Department of Electrical Engineering, MIT, 1974.
- [Tho97a] W. Thomas. Ehrenfeucht games, the composition method, and the monadic theory of ordinal words. In Jan Mycielski, Grzegorz Rozenberg, and Arto Salomaa, editors, *Structures in Logic and Computer Science, A Selection of Essays in Honor of Andrzej Ehrenfeucht*, volume 1261 of Lecture Notes in Computer Science, pages 118–143. Springer, 1997.

- [Tho97b] W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, chapter 7, pages 386–455. Springer-Verlag, 1997.
- [TW68] J. Thatcher and J. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory*, 2(1):57–81, 1968.
- [Ull89] J. Ullman. *Principles of database and knowledge-base systems*, volume I and II. Computer Science Press, 1989.
- [Var82] M.Y. Vardi. The complexity of relational query languages. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, STOC '82, pages 137–146, 1982.
- [Yan81] M. Yannakakis. Algorithms for acyclic database schemas. In *Proc. 7th International Conference on Very Large Data Bases*, pages 82–94. IEEE Press, 1981.