

Κεφάλαιο 2

Γράφοι τριάδων και το πρότυπο RDF

Τα Ανοικτά Συνδεδεμένα Δεδομένα, και γενικότερα ο Σημασιολογικός Ιστός, οργανώνουν τα δεδομένα ακολουθώντας το πρότυπο *Resource Description Framework* (RDF) του οργανισμού World Wide Web Consortium ([W3C](#)). Στο κεφάλαιο αυτό θα εξετάσουμε πώς από τα συνήθη μοντέλα οργάνωσης *δομημένης* (structured) πληροφορίας μπορούμε να μεταβούμε στις *τριάδες* (triples) και στους *γράφους* (graphs) του προτύπου RDF.

Ειδικότερα, θα μελετήσουμε τις βασικές δομές του μοντέλου οργάνωσης σε απλούς πίνακες (flat tabular data), καθώς και το σχεσιακό μοντέλο ως μετεξέλιξη των απλών πινάκων. Στη συνέχεια θα συναντήσουμε τις τριάδες και τους γράφους ως βασικά εργαλεία έκφρασης της σημασιολογικής πληροφορίας. Στο δεύτερο μέρος του κεφαλαίου θα ασχοληθούμε με το πρότυπο RDF καθαυτό: πώς σχηματίζονται οι τριάδες και οι γράφοι και πώς εκφράζεται η πληροφορία μέσω *αναγνωριστικών URI* και *απλών τιμών* (literals). Στο τέλος θα μάθουμε για τις διάφορες μορφές *αναπαράστασης* (serialization) των δεδομένων RDF.

Σημείωση: Γιατί λέμε «η RDF»;

Στο κείμενο που ακολουθεί, αναφερόμαστε στο πρότυπο RDF χρησιμοποιώντας συχνά τις εκφράσεις «η RDF» ή «σύμφωνα με την RDF». Γιατί συμβαίνει αυτό; Κατά σύμβαση η RDF θεωρείται «γλώσσα προγραμματισμού», οπότε είναι φυσικό να αναφερόμαστε σε αυτή με τον ίδιο τρόπο που μιλάμε για «την Python, την Javascript ή την C». Θα πρέπει πάντοτε να θυμόμαστε, όμως, ότι στην πραγματικότητα η RDF δεν είναι γλώσσα, αλλά *μοντέλο οργάνωσης δεδομένων* (data model)!

2.1 Τυπικά μοντέλα δομημένης πληροφορίας

Ένα τυπικό μοντέλο οργάνωσης δομημένης πληροφορίας καλείται να εκφράσει δεδομένα όπως:

Εργαζόμενος129 - μισθός - 1200

Σμόλικας - υψόμετρο - 2637

UserX - email - userx@example.com

Στα προηγούμενα παραδείγματα δεδομένων εμφανίζεται ένα πολύ κοινό σχήμα οργάνωσης: μια συγκεκριμένη οντότητα, για μια συγκεκριμένη ιδιότητα, παρουσιάζει μια ορισμένη τιμή. Οι λεπτομέρειες εξαρτώνται από την εκάστοτε εφαρμογή, αλλά το γενικό σχήμα της τριάδας δεν αλλάζει. Οι τριάδες αυτές υπάρχουν σε όλα τα οικεία μοντέλα οργάνωσης δομημένων δεδομένων.

Ας πάρουμε, για παράδειγμα, την οργάνωση σε πίνακα των δεδομένων ενός ωρολογίου διδακτικού προγράμματος, όπως φαίνεται στον **πίνακα 2.1**:

Πίνακας 2.1: Παράδειγμα ωρολογίου προγράμματος

Ημέρα	Έναρξη	Λήξη	Αίθουσα	Μάθημα
Δευτέρα	10:00	12:00	Αιθ.1	Προγραμματισμός C
Τρίτη	10:00	12:00	Αιθ.2	Σηματολογικός Ιστός
Τρίτη	12:00	14:00	Αιθ.1	Δομές Δεδομένων

Με την πρώτη ματιά, ίσως είναι δύσκολο να εντοπισθούν οι τριάδες στον **πίνακα 2.1**. η λύση βρίσκεται στο γεγονός ότι κάθε γραμμή του πίνακα αναφέρεται έμμεσα σε μια διαφορετική οντότητα. Έστω ότι η οντότητα που αντιπροσωπεύει κάθε γραμμή είναι μία *διάλεξη* ενός μαθήματος, η οποία γίνεται σε ορισμένη ώρα, ημέρα και αίθουσα. Για λόγους ευκολίας, ας προσθέσουμε την έμμεση αυτή αναφορά κάθε γραμμής στον **πίνακα 2.2**:

Πίνακας 2.2: Οι οντότητες των *διαλέξεων* στο ωρολόγιο πρόγραμμα

	Ημέρα	Έναρξη	Λήξη	Αίθουσα	Μάθημα
<i>διάλεξη1</i>	Δευτέρα	10:00	12:00	Αιθ.1	Προγραμματισμός C

	Ημέρα	Έναρξη	Λήξη	Αίθουσα	Μάθημα
διάλεξη2	Τρίτη	10:00	12:00	Αιθ.2	Σημαιολογικός Ιστός
διάλεξη3	Τρίτη	12:00	14:00	Αιθ.1	Δομές Δεδομένων

Σ' αυτή τη δεύτερη μορφή του [πίνακα 2.2](#) είναι εύκολο να παρατηρήσουμε τις τριάδες. Κάθε κεντρικό κελί στη διασταύρωση της γραμμής R και της στήλης C περιέχει την τιμή που παίρνει η αντίστοιχη έμμεση οντότητα της γραμμής R για την ιδιότητα που προσδιορίζεται από την επικεφαλίδα της στήλης C . Από τον προηγούμενο πίνακα προκύπτουν 15 (3×5) τριάδες, όπως η ακόλουθη:

διάλεξη2 (οντότητα) – Αίθουσα (ιδιότητα) – Αιθ.2 (τιμή)

Το μοντέλο οργάνωσης σε απλό πίνακα, λόγω των περιορισμών που επιβάλλει η μορφή του, δεν αποτελεί σήμερα το βασικό μοντέλο οργάνωσης δεδομένων. Εάν χρειαστεί να αποθηκεύσουμε πολλαπλές τιμές για την ίδια ιδιότητα της ίδιας οντότητας –φανταστείτε η [διάλεξη1](#) του Προγραμματισμού C να έχει ταυτόχρονα δύο εργαστηριακούς βοηθούς– τότε θα πρέπει να καταφύγουμε σε μη πρότυπες λύσεις, όπως να προσθέσουμε δύο τιμές σε ένα κελί ή να έχουμε δύο στήλες με την ίδια επικεφαλίδα. Παρ' όλα αυτά, οι απλοί πίνακες είναι ιδανικοί για την ανταλλαγή μεγάλου όγκου δεδομένων μεταξύ συστημάτων (μορφές csv και tsv).

Το κυρίαρχο μοντέλο δεδομένων σήμερα είναι αναμφισβήτητο το *σχεσιακό*. Αυτό μπορεί να θεωρηθεί μετεξέλιξη των απλών πινάκων: εδώ πρόκειται για πολλαπλούς πίνακες που, σύμφωνα με τη σχεσιακή θεωρία, διασυνδέονται μεταξύ τους μέσω τιμών-κλειδιών. Το σχεσιακό μοντέλο, μέσω των διαφόρων μορφών *κανονικοποίησης* (normalization), αποφεύγει τα προβλήματα που ήδη αναφέρθηκαν στην περίπτωση των απλών πινάκων. Πού βρίσκονται, όμως, οι τριάδες στο σχεσιακό μοντέλο; Ας δούμε ένα παράδειγμα στον [πίνακα 2.3](#), πάντοτε πάνω στο ωρολόγιο πρόγραμμά μας:

Πίνακας 2.3: Τα μαθήματα των διαλέξεων

ID	Μάθημα	Εξάμηνο	Τύπος
1	Προγραμματισμός C	A	Εργαστήριο
2	Δομές Δεδομένων	Γ	Θεωρία
3	Σημαιολογικός Ιστός	H	Θεωρία

Ο [πίνακας 2.3](#) των μαθημάτων συνοδεύεται από τον [πίνακα 2.4](#) των διαλέξεων:

Πίνακας 2.4: Οι διαλέξεις σε σχέση με τα μαθήματα

ID	Ημέρα	Έναρξη		Αίθουσα	Μάθημα_ID
		Λήξη			
1	Δευτέρα	10:00	12:00	Αιθ.1	1
2	Τρίτη	10:00	12:00	Αιθ.2	3
3	Τρίτη	12:00	14:00	Αιθ.1	2
4	Τετάρτη	15:00	17:00	Αιθ.2	3

Σε αυτόν τον πίνακα οι διαλέξεις συνδέονται με το αντίστοιχο μάθημα μέσω του αναγνωριστικού αριθμού Μάθημα_ID του μαθήματος. Μέσω του αριθμού αυτού μπορούμε να συγχωνεύσουμε τους δύο πίνακες -καταστρατηγώντας, βέβαια, το σχεσιακό μοντέλο- και να καταλήξουμε στον απο-κανονικοποιημένο **πίνακα 2.5**:

Πίνακας 2.5: Ο απο-κανονικοποιημένος πίνακας διαλέξεων-μαθημάτων

ID	Ημέρα	Έναρξη		Αίθουσα	Μάθημα	Εξάμηνο	Τύπος
		Λήξη					
1	Δευτέρα	10:00	12:00	Αιθ.1	Προγραμματισμός C	A	Εργαστήριο
2	Τρίτη	10:00	12:00	Αιθ.2	Σημειολογικός Ιστός	H	Θεωρία
3	Τρίτη	12:00	14:00	Αιθ.1	Δομές Δεδομένων	Γ	Θεωρία
4	Τετάρτη	15:00	17:00	Αιθ.2	Σημειολογικός Ιστός	H	Θεωρία

Αν και έχουμε καταστρέψει τη σχεσιακή μας οργάνωση, καταλήξαμε στη μορφή απλού πίνακα, απ' όπου μπορούμε άμεσα να εξάγουμε τις ζητούμενες τριάδες!

2.2 Δοκιμάστε κι εσείς!

Μπορείτε να εξάγετε τριάδες από οποιαδήποτε είδος και μορφή πληροφορίας και όχι μόνο από δεδομένα σε μορφή πίνακα ή σύμφωνα με το σχεσιακό μοντέλο.

Δοκιμάστε να σκεφτείτε το είδος των τριάδων (οντότητες, ιδιότητες και τιμές) που μπορούν να εξαχθούν:

- Από ένα σύστημα βαθμολογίας φοιτητών (Υπόδειξη: Σκεφτείτε το σχεσιακό μοντέλο).
- Από την πληροφορία ενός e-shop (Ποια πληροφορία συνοδεύει τα προϊόντα; σκεφτήκατε τις κριτικές και τη βαθμολογία των χρηστών;)
- Η πασίγνωστη Wikipedia είναι φτιαγμένη για να διαβάζεται από ανθρώπους κι όχι από εφαρμογές. Παρ' όλα αυτά έχει χαρακτηριστικά μέρη που ταιριάζουν ιδανικά στη μορφή των τριάδων. Μπορείτε να τα αναγνωρίσετε; (Η λέξη-κλειδί είναι «infobox»)
- Ένα οικείο παράδειγμα: Πάρτε οποιαδήποτε εφαρμογή social media. Ποιες τριάδες μπορούν να εξαχθούν; (Μην ξεχνάτε ότι οι τριάδες συνδέουν πράγματα)
- Εξωτικές μορφές πληροφορίας: Τι τριάδες μπορείτε να εξάγετε από ένα λογοτεχνικό έργο; Ένα μουσειακό έκθεμα; Έναν χάρτη;

2.3 Οργάνωση δεδομένων σε τριάδες

Στην προηγούμενη ενότητα είδαμε ότι κατά την οργάνωση της δομημένης πληροφορίας εμφανίζεται ένα βασικό μοντέλο οργάνωσης: οι τριάδες (triples). Στη συνέχεια θα μελετήσουμε πιο τυπικά τις τριάδες και θα προσπαθήσουμε να συμπληρώσουμε τα κομμάτια που λείπουν, έτσι ώστε να μπορέσουν οι τριάδες να χρησιμοποιηθούν στον Σημασιολογικό Ιστό και στα Ανοικτά Συνδεδεμένα Δεδομένα.

Σύμφωνα με όσα είδαμε προηγουμένως, μια τριάδα (οντότητα, ιδιότητα, τιμή) αποτελεί μια δήλωση (assertion) ότι η συγκεκριμένη οντότητα, για την επιλεγμένη ιδιότητα, έχει ορισμένη τιμή.¹

Η έννοια της δήλωσης που προσδιορίζει κάθε τριάδα μπορεί να περιγραφεί με όρους μαθηματικής λογικής. Το διμελές κατηγορημα (binary predicate) $P(x, y)$, μια συνάρτηση P δηλαδή που είναι αληθής για τα δύο ορίσματα x, y , ισοδυναμεί με την τριάδα (x, P, y) . Για παράδειγμα, η τριάδα:

(Σμόλικας, υψόμετρο, 2637)

ισοδυναμεί με το κατηγορημα:

υψόμετρο(Σμόλικας, 2637)

¹Η δήλωση αυτή περιγράφεται στο πλαίσιο των βάσεων δεδομένων και ως μοντέλο EAV (Entity-Attribute-Value model).

Η προηγούμενη τριάδα μπορεί επίσης να αντιμετωπιστεί ως μια πρόταση σε φυσική γλώσσα:

(Ο) Σμόλικας (έχει) υψόμετρο 2637 (μέτρα).

όπου το *υποκείμενο* (subject) είναι το όρος Σμόλικας, το *κατηγορημα* (predicate) είναι το υψόμετρο ενώ το *αντικείμενο* (object) είναι ο αριθμός 2637. Ο Σημασιολογικός Ιστός υιοθετεί ακριβώς αυτή την ορολογία και περιγράφει μια τριάδα ως:

(subject, predicate, object)

Στα παραδείγματα του κεφαλαίου, και πριν γνωρίσουμε πρότυπες μορφές γραφής των τριάδων, θα χρησιμοποιήσουμε την αφαιρετική μορφή:

subject – predicate – object

Στο σημείο αυτό ίσως αναρωτηθούμε γιατί η μορφή της τριάδας είναι τόσο επιθυμητή στο πλαίσιο του Σημασιολογικού Ιστού και των Συνδεδεμένων Δεδομένων. Αυτό μπορεί να γίνει εύκολα κατανοητό αν επιστρέψουμε στο αρχικό ζητούμενο που τέθηκε στη εισαγωγή του βιβλίου:

Να μπορούν οι «μηχανές» να καταλάβουν τα δεδομένα που επεξεργάζονται το ίδιο καλά με τον άνθρωπο.

Το έργο αυτό είναι εξαιρετικά δύσκολο, αλλά τα δεδομένα σε μορφή τριάδων αποτελούν το πρώτο βήμα για τη γεφύρωση του σημασιολογικού κενού (semantic gap) στην κατανόηση των δεδομένων μεταξύ ανθρώπου και υπολογιστή.

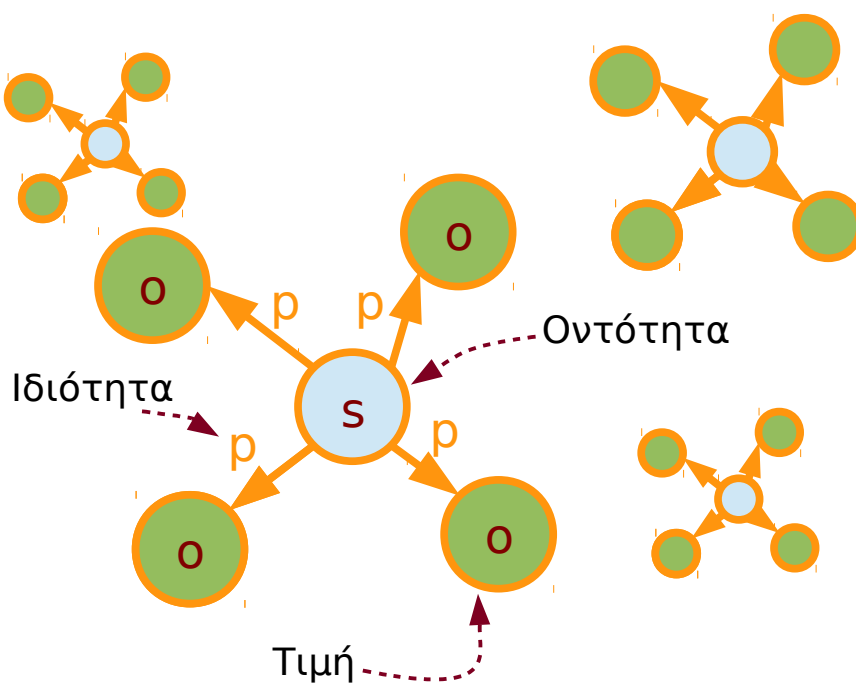
Ας θυμηθούμε τις απαιτήσεις που έχουμε από τα σημασιολογικά δεδομένα:

- *Θα πρέπει να παρέχονται σε «ωμή» μορφή.* Οι τριάδες, παρά την πληθωρικότητα της εμφάνισής τους, απεικονίζουν τα δεδομένα ως έχουν, χωρίς κανένα ενδιάμεσο στάδιο επεξεργασίας.
- *Θα πρέπει να είναι εύκολα αναγνώσιμα από τη «μηχανή».* Οι τριάδες αποτελούν μια ιδανική μορφή οργανωμένης πληροφορίας, με σταθερό αριθμό τριών στοιχείων. Μια τριάδα μπορεί να αναγνωριστεί, να γίνει αντικείμενο επεξεργασίας και να αποθηκευτεί *ανεξάρτητα* από τις υπόλοιπες, περιορίζοντας έτσι τις απαιτήσεις σε υπολογιστικούς πόρους.
- *Θα πρέπει να μεταφέρουν ρητά τη σημασιολογία τους.* Μια τριάδα, όπως την περιγράψαμε προηγουμένως, μετατρέπει μέσω του κατηγορήματος τη σχέση μεταξύ υποκειμένου και αντικειμένου σε *ρητή* πληροφορία. Η τριάδα, ως το μικρότερο κομμάτι σημασιολογικής πληροφορίας με σταθερούς τους ρόλους των τριών μερών της, μεταφέρει ρητά τη σημασία της πληροφορίας που δηλώνει.

Φυσικά οι τριάδες, όπως έχουν παρουσιαστεί ως τώρα, αποτελούν μόνο το πρώτο βήμα: στη συνέχεια θα ενισχύσουμε την εμφάνισή τους, έτσι ώστε να μπορούν να χρησιμοποιηθούν απρόσκοπτα στον παγκόσμιο Σημασιολογικό Ιστό.

2.4 Από τις τριάδες στους γράφους

Στην προηγούμενη ενότητα («Οργάνωση δεδομένων σε τριάδες») μελετήσαμε την τριάδα ως μεμονωμένη οντότητα πληροφορίας. Ακόμα νωρίτερα, στα «Τυπικά μοντέλα δομημένης πληροφορίας», είδαμε πώς μπορούμε να μετασχηματίσουμε δεδομένα που βρίσκονται σε μορφή απλού πίνακα (flat table) σε ένα σύνολο τριάδων. Οι τριάδες που προκύπτουν από τον μετασχηματισμό αυτόν έχουν κατά ομάδες το ίδιο υποκείμενο: την οντότητα που αντιπροσωπεύει κάθε γραμμή του πίνακα. Κάθε οντότητα θα εμφανίζεται ως υποκείμενο τόσες φορές όσος είναι ο αριθμός των στηλών (ιδιοτήτων) του πίνακα. Αν ομαδοποιήσουμε τις τριάδες με ίδιο υποκείμενο προκύπτει το **σχήμα 2.1**: πρόκειται για ομάδες τριάδων σε μορφή «αστέρα» ή «χιονονιφάδας».



Σχήμα 2.1: Μεμονωμένες ομάδες τριάδων με ίδιο υποκείμενο

Όμως η εικόνα αυτή των απομονωμένων ομάδων τριάδων δεν ανταποκρίνεται στην πραγματικότητα!

Ας δούμε ξανά τον **πίνακα 2.6** του αρχικού παραδείγματος:

Πίνακας 2.6: Το ωρολόγιο πρόγραμμα του αρχικού παραδείγματος

	Έναρξη				
	Ημέρα		Λήξη	Αίθουσα	Μάθημα
διάλεξη1	Δευτέρα	10:00	12:00	Αιθ.1	Προγραμματισμός C
διάλεξη2	Τρίτη	10:00	12:00	Αιθ.2	Σηματολογικός Ιστός
διάλεξη3	Τρίτη	12:00	14:00	Αιθ.1	Δομές Δεδομένων

Στις τριάδες που παράγονται θα χρησιμοποιηθούν ως αντικείμενα (το τρίτο μέρος της τριάδας):

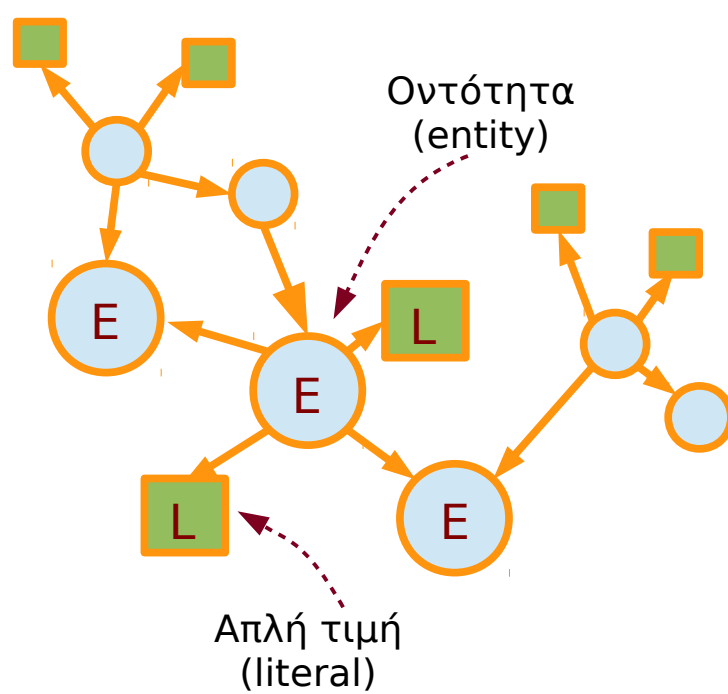
- οι μέρες, π.χ. διάλεξη2 – Ημέρα – Τρίτη
- οι ώρες, π.χ. διάλεξη1 – Λήξη – 12:00
- οι αίθουσες, π.χ. διάλεξη3 – Αίθουσα – Αιθ.1
- και τα μαθήματα, π.χ. διάλεξη2 – Μάθημα – Σηματολογικός Ιστός

Συγκρίνοντας αυτά τα ετερόκλητα είδη στο αντικείμενο των τριάδων, ο προσεκτικός αναγνώστης θα παρατηρήσει ότι *δεν ανήκουν όλα στην ίδια σηματολογική κατηγορία*. Οι αίθουσες και τα μαθήματα κρύβουν μεγαλύτερη πολυπλοκότητα από τις ώρες και ημέρες:

- Μια *αίθουσα* περιλαμβάνει πρόσθετη πληροφορία: είναι αίθουσα ή εργαστήριο; πόσες θέσεις περιλαμβάνει; ποιον άλλον επιπλέον εξοπλισμό διαθέτει;
- Ένα *μάθημα* συνοδεύεται επίσης από άλλα δεδομένα: πόσες διδακτικές μονάδες; υποχρεωτικό ή επιλογής; κατεύθυνσης ή κορμού; κ.ο.κ.

Με άλλα λόγια, οι αίθουσες και τα μαθήματα *είναι στην πραγματικότητα οντότητες*. Αντιθέτως, οι μέρες και οι ώρες είναι μόνο αυτό που δηλώνουν: *απλές τιμές* (literals). Αν χρησιμοποιήσουμε τη νέα αυτή θεώρηση των τριάδων, προκύπτει το **σχήμα 2.2** όπου οι συνδέσεις δεν είναι πλέον σε μορφή «αστέρα»: τώρα πια έχουμε να κάνουμε με έναν *γράφο τριάδων*. Η παρατήρηση αυτή είναι πολύ σημαντική για τον Σηματολογικό Ιστό και τα Συνδεδεμένα Δεδομένα, τα οποία εκμεταλλεύονται στο έπακρο τις διασυνδέσεις μεταξύ οντοτήτων. Το ίδιο το πρότυπο της RDF χρησιμοποιεί ακριβώς αυτό το μοντέλο γράφου τριάδων για να οργανώσει τα δεδομένα του.

Ένας γράφος τριάδων έχει έναν ξεχωριστό *κόμβο* για κάθε οντότητα (entity) και απλή τιμή (literal). Θυμηθείτε ότι οι οντότητες βρίσκονται σε θέσεις υποκειμένου



Σχήμα 2.2: Οι τριάδες ως γράφος

και αντικειμένου στις τριάδες, ενώ οι απλές τιμές περιορίζονται μόνο στη θέση του αντικειμένου! Ο γράφος περιγράφει τις σχέσεις μεταξύ οντοτήτων (ή μεταξύ οντότητας-τιμής) μέσω των ακμών του. Οι ακμές αυτές δεν είναι παρά τα κατηγορήματα των τριάδων. Παρατηρήστε ότι, ενώ κάθε οντότητα αντιπροσωπεύεται από έναν και μοναδικό κόμβο, *δεν συμβαίνει το ίδιο με τις απλές τιμές*. Κάθε απλή τιμή έχει τον δικό της κόμβο, *ακόμα κι αν έχει την ίδια τιμή με άλλες!*

Σημείωση: Ο γράφος είναι προσανατολισμένος.

Ο γράφος που προκύπτει από τη διασύνδεση των τριάδων είναι *κατευθυνόμενος*. Η φορά των ακμών (κατηγορημάτων) είναι από το υποκείμενο προς το αντικείμενο. Από λειτουργική άποψη, όμως, ο προσανατολισμός των ακμών είναι ισοδύναμος και μπορεί να επιλεγεί ανάλογα με τις ανάγκες της εκάστοτε εφαρμογής. Για παράδειγμα, μπορείτε να πείτε εξίσου καλά

Χ.Ανδρέου – διδάσκει – Σημαιολογικός Ιστός

και

Σημαιολογικός Ιστός – διδάσκεται_από – Χ.Ανδρέου

αρκεί αυτό που διαλέξατε να εξυπηρετεί την εφαρμογή σας.

2.5 Η μοναδικότητα της οντότητας

Μέχρι τώρα, είδαμε την οντότητα, η οποία μεταφέρεται στο πρώτο μέρος (υποκείμενο) κάθε τριάδας, να υποδηλώνεται από ένα ασαφώς επιλεγμένο αναγνωριστικό: είτε μιλάμε για τη *διάλεξη*⁷ είτε για το μάθημα με *ID 3*, το αναγνωριστικό αυτό δεν είναι αρκετά ισχυρό για να εκπροσωπήσει σε παγκόσμιο επίπεδο την αντίστοιχη οντότητα. Το ίδιο όμως ισχύει και για κάθε οντότητα σε θέση αντικειμένου (στο τρίτο μέρος των τριάδων).

Μέχρι τώρα θεωρούσαμε ότι το ίδιο όνομα υποδηλώνει την ίδια οντότητα. Αυτό μπορεί να ακούγεται λογικό όσο παραμένουμε στον ίδιο πίνακα, τι θα συμβεί όμως όταν οι τριάδες προέρχονται όχι μόνο από διαφορετικούς πίνακες αλλά και από διαφορετικούς κατασκευαστές; Μπορούμε να υποθέσουμε ότι η Αίθουσα 1 ενός πανεπιστημίου είναι η ίδια με την Αίθουσα 1 ενός δεύτερου πανεπιστημίου;

Στο επόμενο παράδειγμα παραθέτουμε αποσπάσματα από δύο πίνακες.

Πίνακας 2.7: Γεωγραφική πληροφορία

ID	Κορυφή	Υψόμετρο(μ)	Οροσειρά
1	Όλυμπος	2917	Όλυμπος
2	Σμόλικας	2637	Πίνδος
3

Πίνακας 2.8: Φυτολογική πληροφορία

ID	Διάρκεια	Χρώμα	Τοξικότητα
1
2	Αειθαλές	Πράσινο	Μέτρια
3

Ο **πίνακας 2.7** περιέχει γεωγραφική πληροφορία και ο **πίνακας 2.8** περιέχει φυτολογική πληροφορία. Σε κάθε πίνακα οι οντότητες αναγνωρίζονται από έναν αριθμό (ID). Από τους προηγούμενους πίνακες παίρνουμε τις εξής τριάδες:

2 – υψόμετρο – 2637

και

2 – τοξικότητα – μέτρια

Βλέποντας τις δύο τριάδες με το κοινό υποκείμενο 2, κανένας άνθρωπος δεν θα σκεφτεί ότι ο Σμόλικας έχει μέτρια τοξικότητα! Αυτό συμβαίνει λόγω της χρήσης της έμμεσης σημασιολογίας. Δεν θα συμβεί όμως το ίδιο με τον υπολογιστή: *εδώ η σημασιολογία πρέπει να μεταφερθεί ρητά* για να επιτύχουμε το σωστό αποτέλεσμα.

Υποθέστε ότι ο πρώτος πίνακας βρίσκεται στον παγκόσμιο ιστό στη διεύθυνση

<http://mountain.top/summits/>

ενώ ο δεύτερος στη διεύθυνση

<http://phyto.to.gy/samples/>

Μια απλή αλλά και ουσιαστική λύση θα ήταν να προσθέσουμε ως πρόθεμα σε κάθε αναγνωριστικό όνομα οντότητας τη διεύθυνση του πίνακά του. Στην περίπτωση αυτή θα είχαμε τις αντίστοιχες τριάδες:

`http://mountain.top/summits/2` – υψόμετρο – 2637

και

`http://phytolo.gy/samples/2` – τοξικότητα – μέτρια

όπου είναι πλέον σαφής και ρητά δηλωμένη η διαφορετικότητα των δύο οντοτήτων στη θέση υποκειμένου.

Την ιδέα αυτή μπορούμε να επεκτείνουμε με τυπικό τρόπο, χρησιμοποιώντας *αναγνωριστικά URI* για να αναγνωρίσουμε κάθε οντότητα. Τα χαρακτηριστικά ενός URI, όπως αναφέρθηκαν στο κεφάλαιο «Ενιαία Αναγνωριστικά URI», ταιριάζουν απόλυτα στον σκοπό μας.

Εξασφάλιση μοναδικότητας. Είτε πρόκειται για HTTP URIs είτε για URIs χωρίς μηχανισμό προσπέλασης (π.χ., URNs), τα URIs εξασφαλίζουν τη μοναδικότητα του ονόματος της οντότητας. Μόνο ο κάτοχος του *χώρου ονομάτων* που ανήκει το URI θα ονοματοδοτήσει με το URI αυτό μια οντότητα.

Η μοναδικότητα που εξασφαλίζει η χρήση των URIs δεν βλάπτει την ευελιξία του μηχανισμού ονοματοδοσίας των οντοτήτων. Τίποτα δεν εμποδίζει τη χρήση δύο διαφορετικών αναγνωριστικών για την ίδια οντότητα: η ταυτοσημία (σε αντίθεση με την αμφισημία) είναι απολύτως επιτρεπτή. Επειδή όμως όσα περισσότερα αναγνωριστικά διαθέτει μια οντότητα, τόσο πιο δύσκολη είναι η ταυτοποίησή της από της εφαρμογές, θα πρέπει:

- Στο ίδιο σύνολο δεδομένων, μια οντότητα να έχει, αν είναι δυνατόν, μόνο ένα αναγνωριστικό.
- Αν δύο παραγωγοί δεδομένων ονομάζουν την ίδια οντότητα με διαφορετικό τρόπο, θα πρέπει να παρέχουν κάποια ένδειξη ισοτιμίας μεταξύ των ονομάτων. Το πολύ σημαντικό αυτό χαρακτηριστικό θα μελετήσουμε αργότερα στο κεφάλαιο των «Λεξιλογίων RDF».

Χρήση προτύπων. Οι εφαρμογές του Σημασιολογικού Ιστού και των Συνδεδεμένων Δεδομένων εξαρτώνται σε απόλυτο βαθμό από τη δυνατότητα ανταλλαγής δεδομένων με πρότυπο τρόπο. Συνεπώς και η ονοματοδοσία των οντοτήτων πρέπει να γίνει υποχρεωτικά με πρότυπο τρόπο. Για τον σκοπό αυτόν χρησιμοποιούνται τα URIs, τα οποία αποτελούν ακρογωνιαίο λίθο των επιπέδων του Σημασιολογικού Ιστού.

Σημείωση: URIs και IRIs.

Στην πραγματικότητα τα πρότυπα του Σημασιολογικού Ιστού και των Συνδεδεμένων Δεδομένων χρησιμοποιούν τη διεθνή μορφή των URIs, τα IRIs (Internationalized Resource Identifiers, [RFC3987](#)), τα οποία επιτρέπουν κάθε χαρακτήρα Unicode πλέον των βασικών λατινικών χαρακτήρων. Στο κείμενο, όμως, χρησιμοποιούμε εναλλάξ τους δύο όρους με την ίδια σημασία.

2.6 Ανώνυμοι κόμβοι

Με βάση όσα έχουμε ήδη αναφέρει, φαίνεται υποχρεωτική η ανάθεση ισχυρών αναγνωριστικών URI σε κάθε κόμβο του γράφου τριάδων. Κι όμως, υπάρχει μια περίπτωση «ανωνυμίας», η οποία όχι μόνο δεν απαγορεύεται αλλά και ενθαρρύνεται! Αυτή είναι η περίπτωση των *ανώνυμων κόμβων* (blank nodes).

Δείτε το επόμενο παράδειγμα. Θέλουμε να εκφράσουμε με τριάδες τη γνώση ότι:

Το Ιόνιο Πανεπιστήμιο βρίσκεται σε γεωγραφικό πλάτος 39.620944 και μήκος 19.923716.

Αυτό μπορεί να επιτευχθεί με τις παρακάτω τριάδες:

`http://universiti.es/ionio – latitude – 39.620944`

`http://universiti.es/ionio – longitude – 19.923716`

αλλά επίσης και με τις τριάδες:

`http://universiti.es/ionio – location – loc_1`

`loc_1 – latitude – 39.620944`

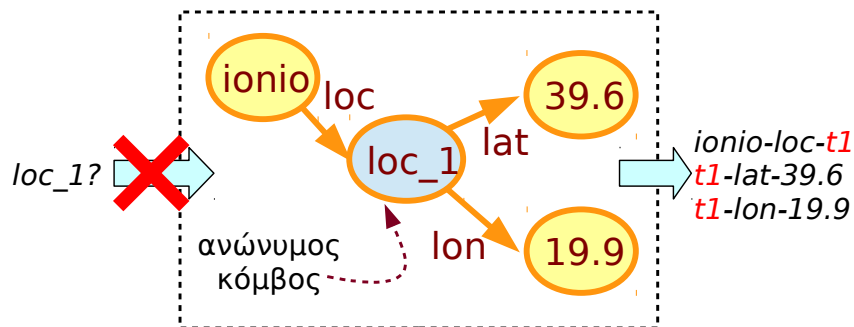
`loc_1 – longitude – 19.923716`

Η πληροφορία δίνεται εξίσου καλά στα δύο παραδείγματα. Γιατί λοιπόν να προτιμήσουμε τη δεύτερη μορφή; Από εννοιολογική άποψη, αυτό που παρέχεται είναι η πληροφορία *θέσης* (location) και όχι ενός μεμονωμένου γεωγραφικού πλάτους ή μήκους. Οι δύο συνιστώσες της θέσης είναι αδιαίρετες· καμία από τις δύο δεν έχει νόημα χωρίς την άλλη. Η δεύτερη μορφή τριάδων εκφράζει σημασιολογικά ακριβώς αυτό το γεγονός.

Έχοντας διαλέξει τη δεύτερη μορφή, ας επικεντρωθούμε στην οντότητα `loc_1`. Η οντότητα αυτή αντιπροσωπεύει μια γεωγραφική θέση και έχει έναν μόνο ρόλο: να συνδέσει το `ionio` με τα 39.620944 και 19.923716. Είναι δηλαδή ένας «τεχνητά» κατασκευασμένος κόμβος, ο οποίος δεν αντιστοιχεί σε κάποια πραγματική οντότητα. Οι κόμβοι αυτοί στην ορολογία του Σημασιολογικού Ιστού ονομάζονται *ανώνυμοι κόμβοι* (blank nodes) και δεν διαθέτουν αναγνωριστικό URI.

Οι ανώνυμοι κόμβοι, παρότι θεωρητικά «ανώνυμοι», πρέπει πρακτικά να πάρουν ένα *τοπικό όνομα* (local name) όταν αποθηκευτούν σε μια βάση δεδομένων ή σ' ένα απλό αρχείο. Αυτό είναι απαραίτητο διότι κάτι πρέπει να χρησιμοποιηθεί στα μέρη της τριάδας όπου εμφανίζεται ο ανώνυμος κόμβος! Δείτε και το προηγούμενο παράδειγμα: το `loc_1` είναι ακριβώς ένα τοπικό όνομα που χρησιμοποιείται για να δηλώσει ότι ο *ίδιος* ανώνυμος κόμβος βρίσκεται σε θέση αντικειμένου στην πρώτη τριάδα και σε θέση αντικειμένου στις δύο επόμενες.

Το τοπικό όνομα που συμβολίζει κάθε ανώνυμο κόμβο έχει *τοπική μόνο ισχύ*: μέσα στη βάση δεδομένων ή στο αρχείο όπου χρησιμοποιείται. Δεν μπορεί να χρησιμοποιηθεί πουθενά αλλού, ούτε για αναζήτηση ούτε για μετάδοση πληροφορίας σχετικής με τον ανώνυμο κόμβο. Μπορείτε να δείτε το τοπικό όνομα ως κάτι περιχαρακωμένο στα όρια της βάσης δεδομένων ή του αρχείου που το περιέχει· δεν μπορεί σε καμιά περίπτωση να διασχίσει τα σύνορα που το περικλείουν, προς οποιαδήποτε κατεύθυνση (σχήμα 2.3).



Σχήμα 2.3: Τα ονόματα των ανώνυμων κόμβων παραμένουν μέσα στα τοπικά όρια

Η δυνατότητα χρήσης ανώνυμων κόμβων μάς απαλλάσσει από την υποχρέωση να δίνουμε ισχυρό URI σε κάθε κόμβο του γράφου μας. Από την άλλη πλευρά, όμως, καθιστά δυσκολότερες ορισμένες λειτουργίες σε γράφους, π.χ., δεν μπορούμε με λεξικογραφική σύγκριση των τριάδων δύο γράφων να δούμε εάν οι γράφοι ταυτίζονται. Δεν υπάρχει «χρυσός κανόνας» για τη χρήση ή μη των ανώνυμων κόμβων. Μια πρακτική αντιμετώπιση είναι να αναρωτηθούμε:

«Θα ζητηθεί ποτέ πληροφορία μεμονωμένα για τον κόμβο X;»

Αν η απάντηση είναι *ναι*, τότε πρέπει να αναθέσουμε στον κόμβο X ένα ισχυρό URI.

Αν είναι *όχι*, ο κόμβος X είναι σοβαρός υποψήφιος για να μετατραπεί σε ανώνυμο κόμβο. Εφαρμόστε τον παραπάνω πρακτικό κανόνα:

- Στο προηγούμενο παράδειγμα. Θα υπάρξει ποτέ ερώτηση «Τι ξέρεις για το loc_1 ;»
 - Εξαιρετικά απίθανο! Η πιο λογική σχετική ερώτηση θα ήταν «Ποια η θέση του Ιονίου Πανεπιστημίου;»
- Στο γνώριμό μας ωρολόγιο πρόγραμμα: «Τι ξέρεις για το $diálexi2$;»
 - Ομοίως απίθανο. Μάλλον περιμένετε ερωτήσεις του τύπου «Πού διδάσκεται ο Σημασιολογικός Ιστός στις 10 το πρωί της Τρίτης;»

Σημείωση: Σχέσεις με περισσότερα από δύο μέλη.

Όπως είδαμε στην ενότητα «**Οργάνωση δεδομένων σε τριάδες**» μια τριάδα (s, p, o) αντιπροσωπεύει τη διμελή σχέση $p(s, o)$. Σίγουρα θα έχετε αναρωτηθεί πώς μπορούν να αναπαρασταθούν σχέσεις με περισσότερα μέλη. Οι ανώνυμοι κόμβοι μπορούν να δώσουν τη λύση. Είδαμε ήδη πώς η τριμελής σχέση $location(ionio, 39.620944, 19.923716)$ μπορεί να αναλυθεί με τη βοήθεια του ανώνυμου κόμβου loc_1 σε διμελείς σχέσεις:

$location(ionio, loc_1)$

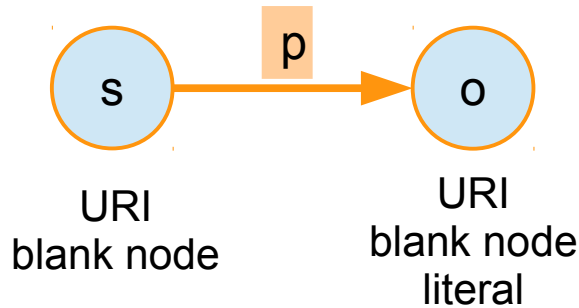
$latitude(loc_1, 39.620944)$

$longitude(loc_1, 19.923716)$

2.7 Κατηγορήματα τριάδων: μια πρώτη ματιά στα λεξιλόγια

Στην ενότητα «**Η μοναδικότητα της οντότητας**» είδαμε την ανάγκη χρήσης αναγνωριστικών URI για τη μοναδική αναγνώριση των οντοτήτων. Οι οντότητες αυτές εμφανίζονται στις τριάδες σε θέση υποκειμένου ή αντικειμένου. Αν προσθέσουμε και τις περιπτώσεις των ανώνυμων κόμβων (σε θέση υποκειμένου ή αντικειμένου) αλλά και των απλών τιμών (literals, πάντα σε θέση αντικειμένου), τα διάφορα είδη των υποκειμένων και αντικειμένων μιας τριάδας εμφανίζονται στο **σχήμα 2.4**.

Τι συμβαίνει, όμως, με το μεσαίο μέρος μιας τριάδας, το κατηγορήμα; Πρέπει και αυτό να αναγνωριστεί με μοναδικό τρόπο; Πόσο σημαντικό είναι αυτό;



Σχήμα 2.4: URIs, ανώνυμοι κόμβοι και απλές τιμές στις θέσεις υποκειμένου - αντικειμένου

Κατά τον μετασχηματισμό του παραδείγματος του ωρολογίου προγράμματος σε γράφο τριάδων («**Από τις τριάδες στους γράφους**»), χρησιμοποιήσαμε ως ακμές διασύνδεσης κόμβων τα κατηγορήματα Ημέρα, Αίθουσα, Μάθημα, Διδάσκων κτλ. Τα κατηγορήματα αυτά περιέχουν μια πολύ σημαντική πληροφορία: *το νόημα της σύνδεσης* μεταξύ δύο οντοτήτων. Ένας άνθρωπος που γνωρίζει ελληνικά ή διαθέτει ένα ελληνικό λεξικό μπορεί να καταλάβει τις έννοιες αυτές και να δράσει κατάλληλα. Για να επιτύχει το ίδιο μια τυπική εφαρμογή, θα πρέπει να συμπεριλάβουμε κατά τον προγραμματισμό της την ίδια πληροφορία:

```
if (..user asks about classrooms..)
  then { ..search triples with "Αίθουσα" in second part.. }
```

Η εικόνα αυτή, αν και συνήθης στις τυπικές εφαρμογές, δεν είναι κατάλληλη για μια εφαρμογή του Σημασιολογικού Ιστού. Η προηγούμενη εφαρμογή:

- *δεν μπορεί να επεξεργαστεί δεδομένα από τρίτες πηγές.* Τι θα συμβεί αν η είσοδος περιλαμβάνει κατηγορήματα σε άλλη γλώσσα;
- *δεν μπορεί να επεξεργαστεί δεδομένα για τα οποία δεν σχεδιάστηκε.* Πώς θα γράψουμε τον κώδικα αν δεν ξέρουμε εκ των προτέρων τα κατηγορήματα;

Με μια πρώτη ματιά, ο στόχος μοιάζει ανέφικτος. Πράγματι, δεν μπορούμε να φτιάξουμε μια εφαρμογή για πλήρως άγνωστα κατηγορήματα! Θα πρέπει να υπάρχει *τουλάχιστον* μια βάση κοινής γνώσης, ένας *ελάχιστος κοινός σημασιολογικός παρονομαστής* για να αρχίσουμε.

Τον ρόλο αυτόν τον παίζουν τα *ευρέως γνωστά λεξιλόγια* (vocabularies). Με τον όρο «λεξιλόγιο» εννοούμε σύνολα από προσυμφωνημένες τιμές (λέξεις) κοινής αποδοχής, με προκαθορισμένη σημασία η καθεμία. Τα λεξιλόγια αυτά καθιερώνονται (ή υιοθετούνται) από διεθνείς οργανισμούς, ενισχύοντας έτσι την εγκυρότητά τους. Εάν διαλέξουμε κατάλληλες τιμές από ένα τέτοιο λεξιλόγιο

για τα κατηγορήματα της εφαρμογής μας, αυξάνεται σημαντικά η δυνατότητα επεξεργασίας δεδομένων τρίτων με σημασιολογική επιτυχία.

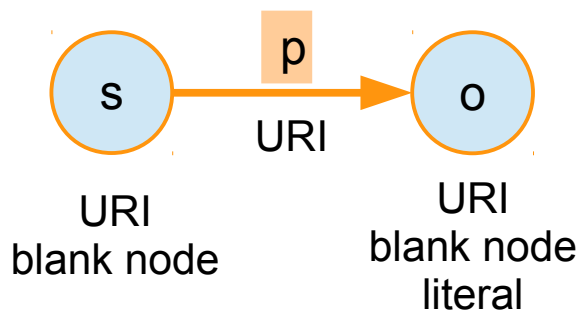
Ακόμα και ο θεωρητικά ανέφικτος στόχος της επεξεργασίας τελείως άγνωστων δεδομένων δεν είναι τόσο άπιαστος. Όπως θα δούμε στο κεφάλαιο «Λεξιλόγια RDF», υπάρχει τρόπος, κατά την εισαγωγή ενός νέου λεξιλογίου, να δηλωθεί η ισοδυναμία του με κάποιο προϋπάρχον. Αν η αλυσίδα ισοδυναμιών τηρηθεί επιμελώς και η εφαρμογή μας έχει κωδικοποιηθεί με ένα παλαιότερο λεξιλόγιο, θα μπορέσει επιτυχώς να χειριστεί και τα άγνωστα σε αυτή μεταγενέστερα λεξιλόγια.

Ένα λεξιλόγιο για παγκόσμια χρήση θα πρέπει να περιλαμβάνει μοναδικά ορισμένες τιμές. Είναι επόμενο λοιπόν να χρησιμοποιούνται URIs στη συγκρότηση των λεξιλογίων, για τον ίδιο ακριβώς λόγο που χρησιμοποιούνται για την αναγνώριση οντοτήτων. Όλες οι τιμές που ανήκουν στο ίδιο λεξιλόγιο διαθέτουν ένα URI με κοινό πρόθεμα, όπως στο εξής παράδειγμα ενός (υποθετικού) λεξιλογίου περιγραφής εργασιακών σχέσεων:

```
http://ex.com/evocab#Employee  
http://ex.com/evocab#Employer  
http://ex.com/evocab#salary  
http://ex.com/evocab#worksAt
```

Το πρόθεμα εδώ είναι το `http://ex.com/evocab#` και προσδιορίζει μονοσήμαντα το χρησιμοποιούμενο λεξιλόγιο. Το κοινό πρόθεμα δημιουργεί έναν *χώρο ονομάτων* (namespace), μέλη του οποίου είναι όλες οι τιμές που περιέχει το συγκεκριμένο λεξιλόγιο.

Η σημασία των λεξιλογίων είναι τόσο μεγάλη που θα αφιερώσουμε ξεχωριστό κεφάλαιο σε αυτά. Προς το παρόν σημειώνουμε ότι τα κατηγορήματα των τριάδων έχουν πλέον μετατραπεί σε URIs, οπότε η τελική εικόνα του τι μπορεί να χρησιμοποιηθεί σε κάθε μέρος της τριάδας (υποκείμενο, κατηγορήματα και αντικείμενο) απεικονίζεται στο [σχήμα 2.5](#).



Σχήμα 2.5: Τα πιθανά είδη τιμών στα μέρη της τριάδας

Σημείωση: Τα λεξιλόγια είναι κρίσιμα για τα Συνδεδεμένα Δεδομένα.

Ο Σημασιολογικός Ιστός και ειδικότερα τα Συνδεδεμένα Δεδομένα *βασίζονται* στη μαζική χρήση ευρέως γνωστών λεξιλογίων για την άντληση κατηγορημάτων (αλλά και κλάσεων οντοτήτων, όπως θα δούμε αργότερα). Παρά τη σοβαρότητα του θέματος αυτού, δεν υπάρχει *υποχρέωση* χρήσης τέτοιων λεξιλογίων. Αντιθέτως, μπορείτε κάλλιστα να εφεύρετε και να χρησιμοποιήσετε τα δικά σας. Αν θέλετε όμως τα δεδομένα σας να έχουν σημασιολογική αξία σε παγκόσμιο επίπεδο, θα πρέπει πάντα να αναζητήσετε ήδη υπάρχοντα λεξιλόγια που θα είναι σχετικά με την εφαρμογή σας, προτού καταλήξετε στην κατασκευή ενός νέου δικού σας!

2.8 Ελέγξτε τις γνώσεις σας

1. Ένα URI μπορεί να αναφέρεται
 - i. σε μία μόνο οντότητα, με μοναδικό τρόπο.
 - ii. σε δύο ή περισσότερες οντότητες.
 - iii. σε δύο ή περισσότερες οντότητες, αρκεί την κάθε χρονική στιγμή να αναφέρεται σε μία μόνο από αυτές.
2. Μια τριάδα (s, p, o) μπορεί να περιέχει σε θέση υποκειμένου (s)
 - i. μόνο αναγνωριστικά URIs.
 - ii. αναγνωριστικά URIs και ανώνυμους κόμβους.
 - iii. αναγνωριστικά URIs, ανώνυμους κόμβους και απλές τιμές (literals).
3. Γιατί είναι σημαντικό να επιλέξετε κατηγορήματα τριάδων από ένα ευρέως γνωστό λεξιλόγιο;
 - i. Για να εξασφαλίσετε την εγκυρότητα των δεδομένων σας.
 - ii. Γιατί είναι υποχρεωτική η χρήση URIs στο δεύτερο μέρος (κατηγόρημα) μιας τριάδας.
 - iii. Για να αυξήσετε τη σημασιολογική αξία των δεδομένων σας προς τρίτους.
4. Κατά την παραγωγή ενός συνόλου δεδομένων σε μορφή τριάδων, δεν βρίσκετε ένα ήδη υπάρχον και ευρέως γνωστό λεξιλόγιο για τα κατηγορήματά σας. Τι είναι καλύτερο να κάνετε;

- i. Να τροποποιήσετε το σχήμα των δεδομένων σας έως ότου να μπορεί να περιγραφεί από ένα υπάρχον λεξιλόγιο.
 - ii. Να κατασκευάσετε το δικό σας λεξιλόγιο.
 - iii. Να κατασκευάσετε το δικό σας λεξιλόγιο, δηλώνοντας ταυτόχρονα τη σχέση του με άλλα προϋπάρχοντα λεξιλόγια, στο βαθμό που αυτό είναι εφικτό.
5. Σε έναν γράφο τριάδων που βρίσκεται on-line στη διεύθυνση <http://ex.com/graph1> υπάρχει ανώνυμος κόμβος με το τοπικό όνομα `node23`. Ποιο από τα παρακάτω είναι αληθές;
 - i. Ο κόμβος αυτός δεν έχει καμιά χρησιμότητα για τον χρήστη, δεν μπορεί να τον χρησιμοποιήσει για να πάρει οποιαδήποτε πληροφορία από τον γράφο.
 - ii. Ο κόμβος χρησιμεύει για τη συγκρότηση της πληροφορίας ότι άλλοι κόμβοι συνδέονται μεταξύ τους μέσω του `node23`.
 - iii. Ο χρήστης μπορεί να πληροφορηθεί για το ποιοι άλλοι κόμβοι συνδέονται στον `node23` επισκεπτόμενος τη διεύθυνση <http://ex.com/graph1/node23>.
6. Σας δίνονται δύο γράφοι (σύνολα τριάδων) και πρέπει να τα συγχωνεύσετε. Οι δύο γράφοι περιέχουν URIs, ανώνυμους κόμβους και απλές τιμές (literals). Τι από τα παρακάτω θα κάνετε;
 - i. Θα γίνει απλή συγχώνευση: ο νέος γράφος θα περιέχει τις τριάδες και από τους δύο αρχικούς γράφους. Τα URIs ταυτίζονται μέσω του ονόματος, ενώ οι ανώνυμοι κόμβοι και οι απλές τιμές δεν ταυτίζονται ποτέ.
 - ii. Τίποτα. Δεν μπορεί να γίνει συγχώνευση εφόσον υπάρχουν ανώνυμοι κόμβοι γιατί δεν γνωρίζουμε αν οι τελευταίοι ταυτίζονται ή όχι.
 - iii. Η συγχώνευση θα γίνει συμπεριλαμβάνοντας στον νέο γράφο τις τριάδες των δύο αρχικών γράφων. Την ίδια στιγμή όμως θα πρέπει να ελεγχθούν τα τοπικά ονόματα των ανώνυμων κόμβων κάθε αρχικού γράφου, εάν συμπίπτουν με τοπικά ονόματα του άλλου γράφου. Στην περίπτωση αυτή θα πρέπει να χρησιμοποιηθεί ένα νέο τοπικό όνομα.

2.9 Το πρότυπο RDF

Το πρότυπο *Resource Description Framework* (RDF) προσδιορίζει ένα μοντέλο οργάνωσης πληροφορίας για τον Σημασιολογικό Ιστό. Τη στιγμή που γράφεται το βιβλίο αυτό, η RDF βρίσκεται στην έκδοση 1.1 [1] (Φεβρουάριος 2014), ως **Σύσταση** (Recommendation²) του οργανισμού W3C.

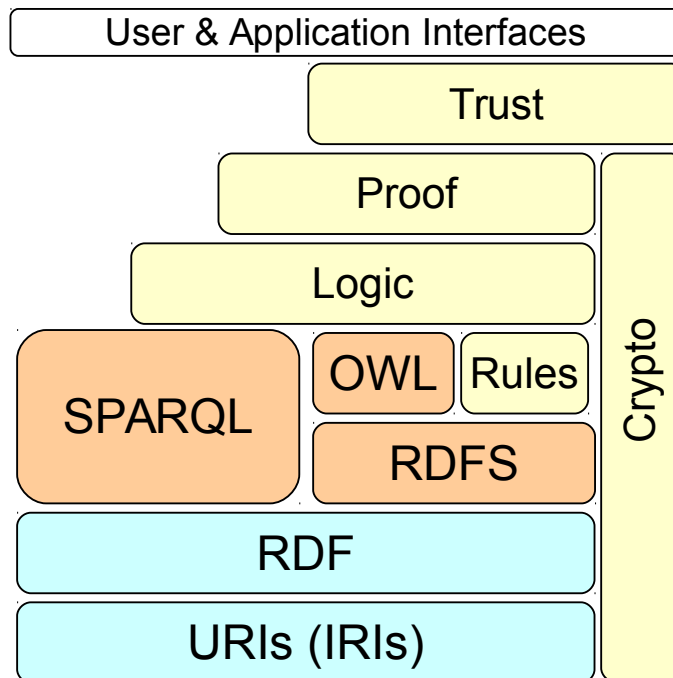
²Μην σας ξεγελά η λέξη Recommendation! Στον W3C η Σύσταση ισοδυναμεί με ισχυρό πρότυπο. Θα περάσουν πολλά χρόνια μέχρι μια πρόταση να φτάσει στην κατάσταση της Σύστασης: η RDF 1.0 που αντικαταστάθηκε από το τρέχον πρότυπο RDF ήταν σύσταση του 2004!

Το πρότυπο RDF βρίσκεται στρατηγικά τοποθετημένο στη γενική εικόνα των επιπέδων του Σημασιολογικού Ιστού («semantic cake»), όπως φαίνεται στο [σχήμα 2.6](#). Χρησιμοποιεί τα αναγνωριστικά URIs για τη μονοσήμαντη ονομασία των οντοτήτων και ταυτόχρονα αποτελεί τη βάση για όλα τα ανώτερα σημασιολογικά επίπεδα.

Η RDF αποτελεί επίσης βασικό συστατικό του κινήματος των Συνδεδεμένων Δεδομένων. Θυμηθείτε τον τρίτο από τους «τέσσερις κανόνες»:

Κανόνας 3: «Όταν κάποιος επισκέπτεται τη διεύθυνση ενός URI, δώστε χρήσιμη πληροφορία χρησιμοποιώντας τα πρότυπα (RDF*, SPARQL)».

Εδώ με τον όρο «RDF*» εννοείται η RDF και το συνοδευτικό πρότυπο RDFS (RDF Schema, θα αναφερθούμε σ' αυτό στο επόμενο κεφάλαιο), τα οποία προσδιορίζονται ως το πρότυπο ανταλλαγής σημασιολογικών δεδομένων.



Σχήμα 2.6: Το πρότυπο RDF στα επίπεδα του Σημασιολογικού Ιστού

Η RDF προτυποποιεί τη χρήση των τριάδων και των γράφων τριάδων, όπως αυτά έχουν παρουσιαστεί στις προηγούμενες ενότητες, σε ένα μοντέλο, το οποίο ονομάζεται *αφηρημένη σύνταξη* (abstract syntax) της RDF:

- Μια *τριάδα RDF* περιλαμβάνει το υποκείμενο (subject), το κατηγορήμα (predicate) και το αντικείμενο (object). Κάθε τριάδα ισοδυναμεί με μια *δήλωση RDF* (RDF statement), ότι δηλαδή μια σχέση ισχύει μεταξύ του υποκειμένου και του αντικειμένου της συγκεκριμένης τριάδας. Ένας *γράφος RDF* απαρτίζεται από ένα σύνολο τριάδων RDF. Προσέξτε ότι, ως σύνολο (με τη μαθηματική έννοια), οι τριάδες ενός γραφού RDF *δεν είναι διατεταγμένες*: δεν υπάρχει δηλαδή τριάδα που να προηγείται κάποιας άλλης!
- Τα τρία ξεχωριστά μεταξύ τους είδη που βρίσκονται στις διάφορες θέσεις των τριάδων, τα URIs, οι ανώνυμοι κόμβοι και οι απλές τιμές (literals), ονομάζονται συλλογικά *όροι RDF* (RDF terms). Τα URIs μπορούν να εμφανιστούν σε κάθε μέρος της τριάδας, οι ανώνυμοι κόμβοι σε θέση υποκειμένου και αντικειμένου και οι απλές τιμές μόνο σε θέση αντικειμένου.
- Στο μοντέλο RDF δεν υπάρχει η έννοια του χρόνου: η ύπαρξη μιας τριάδας δεν υπονοεί κανέναν χρονικό προσδιορισμό. Το μόνο βέβαιο είναι ότι η τριάδα ισχύει κατά τη στιγμή της δήλωσής της. Για τον λόγο αυτόν ένας *γράφος RDF* μπορεί να θεωρηθεί ως ένα χρονικό στιγμιότυπο (snapshot) δεδομένων RDF. Αν απαιτείται η προσθήκη χρονικής πληροφορίας, αυτό μπορεί να γίνει με το κατάλληλο λεξιλόγιο και τη χρήση πρόσθετων τριάδων.
- Μια *πηγή RDF* (RDF source) είναι μια βάση δεδομένων RDF ή ένα έγγραφο δεδομένων RDF που περιέχει γραφούς RDF. Μια πηγή RDF μπορεί να αλλάζει το περιεχόμενό της (προσθήκη, διαγραφή ή αλλαγή τριάδων) με την πάροδο του χρόνου.
- Η RDF 1.1 εισάγει την έννοια του *συνόλου δεδομένων RDF* (RDF dataset): μια ομάδα γραφών RDF, όπου εκτός από τον εξ ορισμού γράφο (default graph) μπορεί να υπάρχουν και άλλοι *επώνυμοι γράφοι* (named graphs). Θα δούμε πιο αναλυτικά το θέμα αυτό στην ενότητα («*Επώνυμοι Γράφοι*»)

Σημείωση: Triplestores και RDF Repositories.

Εκτός από τους όρους που προσδιορίζει το πρότυπο RDF θα συναντήσετε συχνά στον κόσμο των Συνδεδεμένων Δεδομένων τους εξής (ανεπίσημους) όρους:

- *Triplestore*: Πρόκειται για βάση δεδομένων ειδικά σχεδιασμένη για την αποθήκευση δεδομένων RDF. Οι βάσεις αυτές έχουν να ανταγωνιστούν τις ήδη υπάρχουσες σχεσιακές βάσεις και άλλες με λιγότερο συμβατική μορφή, όπως οι βάσεις γραφών. Θα πρέπει πάντα να θυμόμαστε ότι δεν είναι απαραίτητο οι τριάδες RDF να αποθηκευτούν ως *τριάδες*! Αρκεί να υπάρχει ένα ενδιάμεσο επίπεδο μετάφρασης, το οποίο θα αναλαμβάνει τη μετατροπή δεδομένων RDF από και προς το εκάστοτε εσωτερικό σχήμα αποθήκευσης.

- *RDF Repository*: Ένα αποθετήριο (πηγή δεδομένων RDF) στον παγκόσμιο ιστό. Εκτός από τη βάση δεδομένων RDF, παρέχει τυπικά υπηρεσίες προσπέλασης οντοτήτων μέσω URIs και αναζήτησης μέσα στους περιεχόμενους RDF γράφους.

2.10 Οι χώροι ονομάτων της RDF

Η RDF, εκτός από την προτυποποίηση των τριάδων και των γράφων, παρέχει ένα λεξιλόγιο, κυρίως για οντολογικούς σκοπούς –για την κατασκευή άλλων λεξιλογίων, τη δήλωση κλάσεων και ιδιοτήτων. Αυτά θα τα εξετάσουμε με λεπτομέρεια στο κεφάλαιο («Λεξιλόγια RDF»). Το ίδιο λεξιλόγιο περιέχει μια σειρά από βοηθήματα (utilities), κάποια από τα οποία θα δούμε στη συνέχεια.

Οι χώροι ονομάτων που χρησιμοποιεί η RDF (RDF namespaces) είναι:

<http://www.w3.org/1999/02/22-rdf-syntax-ns#> (συντομογραφικά, `rdf:`)

<http://www.w3.org/2000/01/rdf-schema#> (συντομογραφικά, `rdfs:`)

Η ύπαρξη δύο ξεχωριστών χώρων ονομάτων είναι, όπως παραδέχεται **το ίδιο το πρότυπο**, ένας «ενοχλητικός αναχρονισμός», ο οποίος όμως διατηρείται για λόγους συμβατότητας. Πληροφορικά, ο χώρος `rdfs:` προέρχεται από το συνοδό πρότυπο RDF Schema, το οντολογικό τμήμα της RDF.

Ας δούμε τώρα μια σειρά από βοηθητικά κατηγορήματα:

- `rdfs:label`: Το πιο γνωστό κατηγορήμα, το οποίο μας επιτρέπει να συνδέσουμε μια οντότητα με μια αναγνώσιμη από τον άνθρωπο ετικέτα (λεζάντα). Π.χ.

`http://mountain.top/summits/2 – rdfs:label – "Σμόλικας"`

Θα πρέπει να τονιστεί ότι το `rdfs:label` δεν απευθύνεται πρωταρχικά στη «μηχανή» αλλά στον άνθρωπο. Παρ' όλα αυτά, το κείμενο της ετικέτας μπορεί να χρησιμοποιηθεί και από τις εφαρμογές.

- `rdfs:comment`: Απευθύνεται επίσης στον άνθρωπο, τεκμηριώνοντας μια οντότητα με μεγαλύτερη λεπτομέρεια.
- `rdfs:seeAlso`: Χρησιμοποιείται σε μια τριάδα για να δείξει ότι το αντικείμενο περιέχει πρόσθετη πληροφορία για το αντικείμενο της τριάδας. Για παράδειγμα:

`http://mountain.top/summits/2 – rdfs:seeAlso –
http://el.wikipedia.org/wiki/Σμόλικας`

Το αντικείμενο του `rdfs:seeAlso` μπορεί να είναι οποιουδήποτε τύπου πληροφοριακή πηγή.

Σημείωση: `rdfs:label`, πολύτιμο αλλά όχι μοναδικό κατηγορήμα.

Είδαμε προηγουμένως ότι το κατηγορήμα `rdfs:label` παρέχει τη δυνατότητα σύνδεσης μιας οντότητας με μια ανθρώπινα αναγνώσιμη ετικέτα. Η πληροφορία αυτή είναι πολύτιμη για τις εφαρμογές που θέλουν να απεικονίσουν την οντότητα με έναν χρήσιμο τρόπο. Η δυνατότητα εφαρμογής πολλαπλών `rdfs:label` στην ίδια οντότητα σε διαφορετικές γλώσσες αυξάνει την ευχρηστία των εφαρμογών.

Θα πρέπει να θυμόμαστε, όμως, ότι σε πολλά σύνολα δεδομένων RDF χρησιμοποιούνται και άλλα κατηγορήματα για τον ίδιο σκοπό. Για παράδειγμα, στο σετ δεδομένων BTC2011³ μπορούμε να βρούμε *τουλάχιστον* τα κατηγορήματα του [πίνακα 2.9](#), με ισοδύναμη με το `rdfs:label` χρήση.

Πίνακας 2.9: Κατηγορήματα λεκτικής περιγραφής οντοτήτων

Κατηγορήμα	Αριθμός
<code>http://www.w3.org/2000/01/rdf-schema#label</code>	8.540.192
<code>http://purl.org/dc/elements/1.1/title</code>	2.062.880
<code>http://www.w3.org/2004/02/skos/core#prefLabel</code>	259.895
<code>http://open.vocab.org/terms/sortLabel</code>	58.519
<code>http://www.fao.org/aims/aos/languagecode.owl#hasEnglishName</code>	7.642
<code>http://www.w3.org/2004/02/skos/core#altLabel</code>	1.517
<code>http://zeitkunst.org/bibtex/0.1/bibtex.owl#title</code>	963
<code>http://www.w3.org/1999/02/22-rdf-syntax-ns#label</code>	6

Παρατηρήστε την εσφαλμένη χρήση του `rdf:label` (τελευταία γραμμή του [πίνακα 2.9](#), αντί του ορθού `rdfs:label`): ο «ενοχλητικός αναχρονισμός» σε δράση!

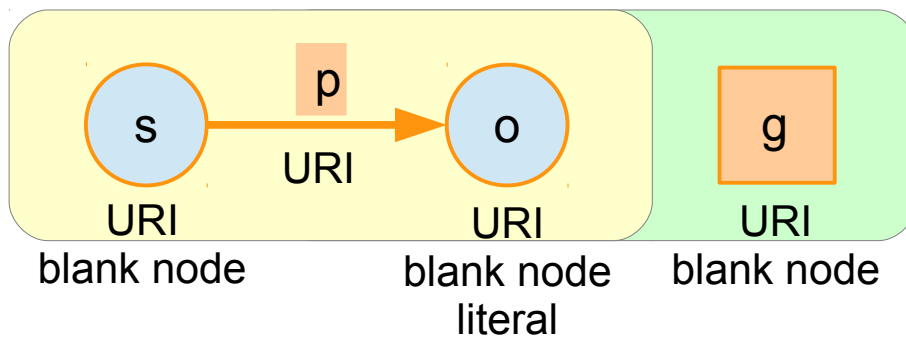
2.11 Επώνυμοι γράφοι

Η RDF 1.1 προσθέτει ένα νέο χαρακτηριστικό που έλειπε από την προηγούμενη έκδοσή της: τους *επώνυμους γράφους* (named graphs). Θα πρέπει να τονίσουμε

³Billion Triple Challenge (BTC): μια σειρά από ετήσιους διαγωνισμούς στον χειρισμό πολύ μεγάλων σετ δεδομένων RDF, στο πλαίσιο του συνεδρίου ISWC. Τα δεδομένα που διατίθενται στους διαγωνιζόμενους έχουν ανακτηθεί από πολλές διασυνδεδεμένες πηγές RDF.

εξαρχής ότι οι όροι «όνομα γράφου» και «επώνυμος γράφος» είναι τελείως παραπλανητικοί - δεν αφορούν τη δυνατότητα ονομασίας κάποιου γράφου τριάδων με τη στενή έννοια. Αντιθέτως, οι επώνυμοι γράφοι είναι ένας μηχανισμός για τη διαίρεση των τριάδων RDF σε υποσύνολα *χωρίς καθορισμένη σημασιολογία*. Το τελευταίο αφήνεται στη διακριτική ευχέρεια των εκάστοτε εφαρμογών.

Η ιδέα είναι απλή και απεικονίζεται στο [σχήμα 2.7](#): στη γνωστή τριάδα RDF προστίθεται ένα τέταρτο μέλος *g* (URI ή blank node). Έχουμε πλέον να κάνουμε με *τετράδα* (quad) RDF.



Σχήμα 2.7: Η τετράδα (quad) ως επέκταση της τριάδας RDF

Όλες οι τριάδες με το ίδιο τέταρτο μέλος *g* θεωρούμε ότι ανήκουν στον ίδιο επώνυμο γράφο με όνομα *g*. Με τον τρόπο αυτόν, το σύνολο των τριάδων μπορεί να χωριστεί σε υποσύνολα· καθένα αντιστοιχεί σε έναν επώνυμο γράφο. Για λόγους συμβατότητας, μαζί με τους επώνυμους γράφους επιτρέπεται η ύπαρξη και ενός γράφου χωρίς όνομα, του *εξ ορισμού γράφου* (default graph). Προσέξτε ότι όλοι αυτοί οι γράφοι, εκτός από τα κοινά URIs, μπορούν να μοιράζονται και ανώνυμους κόμβους!

Σε αντίθεση με τα περισσότερα χαρακτηριστικά της RDF, τα οποία βασίζονται στη μαθηματική λογική, η τετράδα προστέθηκε στο πρότυπο για να ικανοποιήσει τις πρακτικές ανάγκες των εφαρμογών. Για λόγους ευελιξίας, η RDF δεν ορίζει τι σημαίνει το τέταρτο μέλος, ούτε προσδιορίζει τι πρέπει να κάνουν οι σημασιολογικές εφαρμογές όταν συναντήσουν τετράδες. Στην πραγματικότητα, η RDF 1.1. έρχεται εκ των υστέρων να «προτυποποιήσει» μια *ad hoc* πρακτική προσθήκης τέταρτου μέλους στις τριάδες, η οποία ήδη εφαρμοζόταν άτυπα από τις εφαρμογές.

Γιατί, όμως, να θέλει κάποιος ένα τέταρτο μέλος στις τριάδες;

- *Για λόγους διαχείρισης δεδομένων.* Στις βάσεις δεδομένων RDF απαιτείται πολλές φορές να διαχωρίσουμε τις τριάδες σε υποσύνολα: είτε γιατί θέλουμε να περιγράψουμε καταγωγή (provenance) των τριάδων από

διαφορετικές πηγές, είτε γιατί πρέπει να σημειώσουμε διαφορετικά χρονικά στιγμιότυπα των ίδιων δεδομένων, είτε τέλος γιατί απαιτούνται διαφορετικά προνόμια προσπέλασης των τριάδων. Στην περίπτωση αυτή, ο ρόλος του τέταρτου μέλους *g* είναι να περιέχει το «όνομα» του κάθε υπο-γράφου RDF.

- Για να γίνουν δηλώσεις επί των τριάδων (*reification*). Έχουμε δει ότι μια τριάδα RDF αποτελεί στην ουσία μια δήλωση, ότι δηλαδή μια σχέση μεταξύ υποκειμένου και αντικειμένου ισχύει. Συχνά θέλουμε να κάνουμε μια δήλωση πάνω στη δήλωση της τριάδας. Για παράδειγμα, να αναθέσουμε μια πιθανότητα, ένα επίπεδο εμπιστοσύνης ή οποιοδήποτε άλλο χαρακτηριστικό στην ισχύ της τριάδας. Εδώ, το τέταρτο μέλος *g* δρα ως «αναγνωριστικό» της συγκεκριμένης τριάδας και μπορεί με τη σειρά του να χρησιμοποιηθεί σε άλλες τριάδες σε θέση υποκειμένου ή αντικειμένου. Για παράδειγμα,

```
http://mounta.in/Everest – http://voc.ab#υψόμετρο_μ – 8848 – bnode123
bnode123 – http://voc.ab#απόκλιση_μ – 0.5
bnode123 – http://voc.ab#έτοςΜέτρησης – 1955
```

Οι δύο παραπάνω περιπτώσεις χρήσης του τέταρτου μέλους, αν και χαρακτηριστικές, δεν είναι οι μοναδικές. Κάθε εφαρμογή είναι ελεύθερη να αποδώσει το δικό της νόημα στο *g*.

Σημείωση: Reification - Ο «καλός», ο «κακός» και ο «άσχημος».

Όπως έχουμε ήδη αναφέρει, η διαδικασία *δηλώσεων πάνω στις δηλώσεις* (*reification*) είναι κάτι που μπορεί να επιτευχθεί μέσω των τετράδων RDF. Αυτή η «καλή» δυνατότητα, όμως, όμως δεν υπήρχε πριν από την RDF 1.1. Τι μπορούσε να γίνει μέχρι τότε;

Θα μπορούσατε (και συνήθως αυτό γινόταν) να χρησιμοποιήσετε το δικό σας σχήμα, προσθέτοντας τεχνητούς ανώνυμους κόμβους, έτσι ώστε να μιμηθείτε μια πολυμελή σχέση (βλ. «Ανώνυμοι κόμβοι»). Η μέθοδος αυτή είναι αποδοτική και ευέλικτη. Χαρακτηρίζεται «κακή» ως προς την *ad hoc* εφαρμογή της και μόνον!

Η «άσχημη» -και γι' αυτό ελάχιστα χρησιμοποιούμενη- μέθοδος βασιζόταν στα κατηγορήματα *rdf:subject*, *rdf:predicate* και *rdf:object* που παρέχει το λεξιλόγιο της RDF. Με τη βοήθεια των τελευταίων κατηγορημάτων, μπορείτε να περιγράψετε την τριάδα (*s, p, o*) ως εξής:

```
g – rdf:subject – s
g – rdf:predicate – p
g – rdf:object – o
```

Στη συνέχεια, μπορείτε να προσθέσετε άλλες τριάδες με το `g` ως υποκείμενο ή αντικείμενο. Η μέθοδος αυτή απαιτεί τρεις νέες τριάδες για να περιγράψετε την αρχική και περιπλέκει τον χειρισμό των τριάδων. Για τον λόγο αυτόν δεν θα τη συναντήσετε σχεδόν ποτέ!

2.12 Τύποι δεδομένων στην RDF

Η RDF προτυποποιεί τους *τύπους δεδομένων* (datatypes) για τις απλές τιμές (literals) που εμφανίζονται σε θέση αντικειμένου (ο) στις τριάδες. Στα παραδείγματα που είδαμε μέχρι τώρα, κάθε απλή τιμή περιέχει μια σειρά χαρακτήρων κειμένου, όπως "Αίθουσα 1" ή "2917". Τα εισαγωγικά σκόπιμα υπενθυμίζουν ότι οι διάφορες μορφές αποθήκευσης της RDF έχουν μορφή κειμένου, συνεπώς όλες οι απλές τιμές, ανεξάρτητα από το είδος τους, θα έχουν τη μορφή σειράς χαρακτήρων (string).

Η αναπαράσταση των απλών τιμών με strings, όμως, *δεν είναι αρκετή* για την επιτυχή επεξεργασία των σημασιολογικών δεδομένων. Το γιατί συμβαίνει αυτό μπορούμε να το δούμε με ένα παράδειγμα: Αν ζητήσουμε από ένα άνθρωπο και έναν υπολογιστή να προσθέσει τα "1" και "2", ποιο θα είναι το αποτέλεσμα;

"1" + "2" = ?

(άνθρωπος): 3

(«μηχανή»): "12"

Το αποτέλεσμα είναι τελείως διαφορετικό! Η έμμεση σημασιολογία, την οποία ως άνθρωποι μπορούμε να αντλούμε από τα δεδομένα μας, θα μας οδηγήσει σε μια διαφορετική απάντηση από εκείνη που θα έδινε ο υπολογιστής. Για άλλη μια φορά θα πρέπει να θυμηθούμε ότι ο σκοπός των Σημασιολογικών Δεδομένων είναι η *ρητή δήλωση* της έμμεσης σημασίας των δεδομένων. Έτσι, η RDF αφιερώνει ένα μεγάλο μέρος του κειμένου του προτύπου της στους τύπους των δεδομένων που επιτρέπεται να έχουν οι απλές τιμές.

Στην RDF κάθε απλή τιμή μπορεί να απεικονιστεί ως ένα ζεύγος (λεκτικό, τύπος):

- Το *λεκτικό* (lexical form) είναι το κείμενο (σειρά χαρακτήρων) που περιέχεται στην απλή τιμή.
- Ο *τύπος* δεδομένων δηλώνεται με ένα URI. Η RDF υιοθετεί -αλλά δεν επιβάλλει- μια σειρά τύπων δεδομένων από το προϋπάρχον πρότυπο [XML Schema](http://www.w3.org/2001/XMLSchema#). Το πρότυπο XML Schema χρησιμοποιεί τον χώρο διευθύνσεων (namespace) <http://www.w3.org/2001/XMLSchema#>

(συντομογραφικά, `xsd:`) για να δηλώσει τύπους δεδομένων, όπως `xsd:string`, `xsd:boolean`, `xsd:integer`, `xsd:date`, `xsd:time` κτλ. Η RDF υποστηρίζει τους περισσότερους από τους τύπους αυτούς για να προσδώσει ρητή σημασιολογία στις απλές τιμές των τριάδων.

Για παράδειγμα, η απλή τιμή

```
"1.663E-4"^^<http://www.w3.org/2001/XMLSchema#double>
```

απεικονίζει το ζεύγος

```
(λεκτικό="1.663E-4", τύπος=xsd:double)
```

Στη συνήθη περίπτωση που δεν δηλώνεται τύπος δεδομένων, υπονοείται έμμεσα ο τύπος `xsd:string`. Για παράδειγμα, η απλή τιμή

```
"Αίθουσα 1"
```

αντιστοιχεί στο ζεύγος

```
(λεκτικό="Αίθουσα 1", τύπος=xsd:string)
```

Η ρητή δήλωση του τύπου δεδομένων για μια απλή τιμή προσδιορίζει τον *τρόπο χειρισμού* των δεδομένων. Εάν γνωρίζετε τον τύπο δεδομένων, μπορείτε να κάνετε τον σωστό *μετασχηματισμό* από το λεκτικό (lexical form) στην πραγματική τιμή (value) που θα χειριστεί η εφαρμογή σας.

Τυπικά, κάθε τύπος δεδομένων προσδιορίζει τα εξής:

- Έναν *χώρο λεκτικών συμβόλων* (lexical space). Περιλαμβάνει τις τιμές σε μορφή κειμένου που μπορούν να εμφανιστούν για τον συγκεκριμένο τύπο. Για παράδειγμα, για τον τύπο `xsd:boolean`:

```
χώρος λεκτικών συμβόλων = {"true", "false", "1", "0"}
```

- Έναν *χώρο τιμών* (value space). Είναι οι πραγματικές τιμές, στις οποίες θα μετασχηματιστούν τα λεκτικά. Για τον τύπο `xsd:boolean`:

```
χώρος τιμών = {true, false} (προσοχή: χωρίς εισαγωγικά!)
```

- Έναν *πίνακα μετασχηματισμού* (lexical-to-value mapping) που καθορίζει πώς θα μετασχηματιστεί το κάθε λεκτικό. Για τον τύπο `xsd:boolean`:

`<"true" → true>, <"false" → false>, <"1" → true>, <"0" → false>`

Εξοπλισμένοι με τους τύπους δεδομένων της RDF, ας επαναλάβουμε την ερώτηση που θέσαμε στην αρχή της ενότητας:

`"1"^^xsd:integer + "2"^^xsd:integer = ?`

(άνθρωπος): 3

(«μηχανή»): `"1"^^xsd:integer → 1, "2"^^xsd:integer → 2, 1+2 = 3`

Επιτυχία! Δηλώνοντας τους τύπους δεδομένων κάναμε ένα ακόμα βήμα για τη γεφύρωση του σημασιολογικού κενού μεταξύ ανθρώπου και μηχανής.

Ξεφεύγοντας από το προηγούμενο σχήμα της απεικόνισης των λεκτικών μορφών σε τιμές, η RDF υποστηρίζει και έναν διαφορετικό τύπο δεδομένων: τις *σειρές χαρακτήρων με ένδειξη γλώσσας* (`rdf:langString`). Ο τύπος αυτός περιγράφει ουσιαστικά δεδομένα `string` με την προσθήκη μιας ετικέτας ένδειξης γλώσσας ([language tag](#)):

`"Semantic Web"@en`

Παρατηρήστε ότι ο τύπος υπονοείται έμμεσα. Μπορούμε να δούμε τα δεδομένα με αυτόν τον τύπο ως μια σειρά τριών τιμών:

(λεκτικό="Semantic Web", τύπος=`rdf:langString`, γλώσσα="en")

Η αντίστοιχη τιμή (value) της απλής αυτής τιμής είναι:

(`"Semantic Web", "en"`)

Πού μπορεί να φανεί χρήσιμη η ένδειξη γλώσσας; Σε ένα απλό σενάριο, η ίδια πληροφορία (απλή τιμή-κείμενο σε θέση αντικειμένου στις τριάδες RDF) μπορεί να περιγράφεται σε διάφορες γλώσσες. Μια εφαρμογή με πολυγλωσσικά χαρακτηριστικά μπορεί να εμφανίσει στον χρήστη το κατάλληλο κείμενο ανάλογα με την επιθυμητή γλώσσα.

Σημείωση: Πρακτική αντιμετώπιση των τύπων δεδομένων.

Η RDF, ενώ ορίζει πολύ τυπικά –περισσότερο από όσο μπορούμε να αναπτύξουμε στο παρόν βιβλίο– τους τύπους δεδομένων, διαθέτει πολύ ελαστικούς κανόνες για την πρακτική εφαρμογή τους:

- Μια εφαρμογή που επεξεργάζεται δεδομένα RDF δεν απαιτείται να αναγνωρίζει καμία δήλωση τύπου δεδομένων. Αρκεί να μπορεί να χειριστεί απλά strings χωρίς τύπο (έμμεσα, τύπου `xsd:string`) και απλά strings με δήλωση γλώσσας (έμμεσα, τύπου `rdf:langString`).
- Αν η εφαρμογή συναντήσει τύπο δεδομένων που δεν αναγνωρίζει, δεν πρέπει να τον απορρίψει. Θα χρησιμοποιήσει την απλή τιμή «ως έχει», με άγνωστο τύπο. Φυσικά, όσο λιγότερους τύπους αναγνωρίζετε τόσο μικρότερη σημασιολογική ισχύ έχει η εφαρμογή σας.
- Μπορείτε να χρησιμοποιήσετε και άλλους τύπους δεδομένων πέρα από τους προκαθορισμένους από το σύνολο XSD. Θα πρέπει, βέβαια, να διαθέτετε και την εφαρμογή που τους αναγνωρίζει!

2.13 Ελέγξτε τις γνώσεις σας

1. Το πρότυπο RDF είναι

- γλώσσα προγραμματισμού του Σημασιολογικού Ιστού.
- μοντέλο οργάνωσης δεδομένων σε γράφους τριάδων.
- μορφότυπο αποθήκευσης τριάδων.

2. Σε έναν γράφο τριάδων RDF χρησιμοποιείται **δύο φορές** η ίδια τριάδα

`http://ex.com/τιμολόγιο1 http://voc.ab/subTotal "23"`

θέλοντας να εκφράσει δύο ξεχωριστά υποσύνολα στο ίδιο τιμολόγιο με την ίδια ακριβώς τιμή. Πιστεύετε ότι θα υπάρξει πρόβλημα λόγω του ότι ο γράφος RDF ορίζεται ως μαθηματικό σύνολο τριάδων;

- Θα υπάρξει πρόβλημα, γιατί στον γράφο μια τριάδα μπορεί να εμφανιστεί μόνο μια φορά.
- Δεν θα υπάρξει πρόβλημα. Ο γράφος θα περιέχει έναν ακριβώς κόμβο `τιμολόγιο1` και επίσης έναν μόνο κόμβο `23`, αλλά με δύο ακμές `subTotal` να τις ενώνουν.
- Δεν θα υπάρξει πρόβλημα. Οι δύο τριάδες, αν και φαίνονται όμοιες, δεν είναι η ίδια τριάδα: οι απλές τιμές (literals) ποτέ δεν συγχωνεύονται στον ίδιο κόμβο.

3. Θέλοντας να αποτυπώσει μια διαδρομή μεταξύ πόλεων, ένας γράφος περιέχει τις εξής τριάδες:

```
http://ex.com/διαδρομή55 http://voc.ab/hasNode "Ηγουμενίτσα"
http://ex.com/διαδρομή55 http://voc.ab/hasNode "Πρέβεζα"
http://ex.com/διαδρομή55 http://voc.ab/hasNode "Βόνιτσα"
http://ex.com/διαδρομή55 http://voc.ab/hasNode "Μύτικας"
http://ex.com/διαδρομή55 http://voc.ab/hasNode "Αστακός"
http://ex.com/διαδρομή55 http://voc.ab/hasNode "Μεσολόγγι"
```

Είναι δυνατή η έκφραση της σειράς των πόλεων της διαδρομής με τις παραπάνω τριάδες;

- i. Ναι, εφόσον φροντίσουμε να αποθηκεύσουμε τις τριάδες ακριβώς με τη σειρά που θέλουμε.
 - ii. Όχι. Οι τριάδες δεν έχουν αλληλουχία μέσα στον γράφο RDF. Απαιτούνται πρόσθετες τριάδες για να εκφράσουμε τη σειρά των πόλεων.
 - iii. Όχι πάντοτε. Μόνο αν τύχει τα ονόματα των πόλεων να ταξινομούνται λεξικογραφικά με τη σειρά της διαδρομής, μόνο τότε οι παραπάνω τριάδες μπορούν να εκφράσουν την επιθυμητή διάταξη.
4. Για να προσθέσετε μια λεζάντα σε έναν κόμβο του γράφου RDF, θα χρησιμοποιήσετε
- i. το κατηγορημα `rdf:label`.
 - ii. το κατηγορημα `rdfs:label`.
 - iii. το κατηγορημα `rdfs:comment`.
5. Αναπτύσσετε εφαρμογή επεξεργασίας σημασιολογικών δεδομένων, οργανωμένων κατά το πρότυπο RDF. Η εφαρμογή σας πρέπει να συγκρίνει εάν ισχύει `"12" > "9"`. Τι απαντάτε;
- i. Σφάλμα. Η σύγκριση δεν μπορεί να γίνει.
 - ii. Το αποτέλεσμα είναι αληθές.
 - iii. Το αποτέλεσμα είναι ψευδές.
6. Όπως και πριν, η ίδια εφαρμογή έχει να συγκρίνει τις απλές τιμές `"25.00"^^xsd:decimal` και `"25.0"^^xsd:decimal`. Ποιο από τα παρακάτω ισχύει;
- i. Οι δύο απλές τιμές είναι ίσες.
 - ii. Οι δύο απλές τιμές είναι ίδιες.
 - iii. Οι δύο απλές τιμές είναι ίσες και ίδιες.

7. Βρισκόμαστε ακόμα στην ίδια εφαρμογή. Συναντάτε απλές τιμές με άγνωστο σε εσάς τύπο δεδομένων (δεν υποστηρίζετε αυτόν τον τύπο). Ποια από τις παρακάτω ενέργειες υπαγορεύει η RDF;
- Θα πρέπει να απορρίψετε την τριάδα που περιέχει αυτή την απλή τιμή.
 - Θα δεχτείτε την τριάδα και θα απεικονίσετε την απλή τιμή ως έχει στον χρήστη.
 - Θα δεχτείτε την τριάδα, θα απεικονίσετε την απλή τιμή στον χρήστη και θα κάνετε συγκρίσεις μεταξύ τιμών αυτού του τύπου κατά λεξικογραφικό τρόπο (σύγκριση strings).

2.14 Αναπαράσταση γράφων RDF

Η παρουσίαση των γράφων και τριάδων (και τετράδων) RDF στις προηγούμενες ενότητες ακολουθούσε ένα αφηρημένο μοντέλο: την *αφηρημένη σύνταξη* (abstract syntax) της RDF. Στην πράξη, όμως, οι αφηρημένοι γράφοι τριάδων πρέπει με κάποιον τρόπο να *αναπαράσταθούν* κατά την ανταλλαγή τους μεταξύ των εφαρμογών. Ως αναπαράσταση εννοούμε στο σημείο αυτό τον τρόπο καταγραφής των γράφων σε αρχεία (RDF documents). Το πρότυπο RDF καθορίζει μια σειρά από μορφές αναπαράστασης των γράφων RDF, τα οποία ονομάζει *συγκεκριμένες συντάξεις* (concrete syntaxes) της RDF. Μια *συγκεκριμένη σύνταξη* περιγράφει τον τρόπο και τα σύμβολα που χρησιμοποιούνται για την καταγραφή σε αρχείο ενός γράφου (της αφηρημένης, δηλαδή, σύνταξης της RDF).

Όλες οι μορφές συγκεκριμένης σύνταξης της RDF είναι σε μορφή απλού κειμένου. Η διαφορά έγκειται στον τρόπο αναπαράστασης των γράφων. Ο τρόπος αυτός καθορίζει σε μεγάλο βαθμό και τη χρήση της κάθε μορφής:

- Οι πιο απλές μορφές (**N-Triples** και **N-Quads**) είναι ιδανικές για την ανταλλαγή πολύ μεγάλου όγκου τριάδων και τετράδων αντίστοιχα. Η απλή μορφή ευνοεί την εύκολη επεξεργασία των αρχείων και την άμεση ανάλυση των τριάδων. Δείτε τις μορφές αυτές ως αντίστοιχες των backup/restore dumps των βάσεων δεδομένων.
- Μετεξέλιξη των προηγούμενων, οι μορφές **Turtle** και **TriG** (για τριάδες και τετράδες RDF, αντίστοιχα) προσθέτουν λίγη «συντακτική ζάχαρη» (syntactic sugar) έτσι ώστε οι γράφοι RDF να μπορούν να γραφούν με πιο συνοπτικό τρόπο.⁴ Η ανάλυση των αρχείων είναι εδώ πιο σύνθετη διαδικασία.
- Μια σειρά από άλλες μορφές (**JSON-LD** και **RDFA**) έχουν έναν διαφορετικό σκοπό: μπορούν να φιλοξενήσουν τριάδες RDF σε ένα ευρύτερο πλαίσιο ή αναμειγνύουν τις τριάδες με άλλη πληροφορία (HTML) στο ίδιο αρχείο.

⁴Κάτι που καθιστά την Turtle αγαπητή για παραδείγματα, εφόσον χωράει καλύτερα στις σελίδες ενός βιβλίου!

- Τέλος, κάποιες μορφές έχουν πέσει σε «δυσμένεια»: Η **XML** από κύρια και υποχρεωτική μορφή ανταλλαγής δεδομένων RDF στην παλαιότερη RDF 1.0 έχει πλέον υποβιβαστεί σε «άλλη μία από τις διαθέσιμες μορφές».

Οι διάφορες μορφές συγκεκριμένης σύνταξης θα παρουσιαστούν στις επόμενες ενότητες περιληπτικά· σήμερα δεν υπάρχει λόγος να γραφούν τριάδες RDF «στο χέρι»! Υπάρχουν πάντα τα κατάλληλα εργαλεία για τη μετατροπή από τη μία μορφή στην άλλη, όπως για παράδειγμα το βοηθητικό πρόγραμμα `raper` (από τη βιβλιοθήκη ανοικτού λογισμικού [Raptor](#)):

```
raper -i ntriples -o rdfxml yourfile.nt > yourfile.rdf.xml
```

Επιπλέον, υπάρχουν τα εργαλεία για την κατασκευή και στη συνέχεια την αποθήκευση γράφων και τριάδων σε αρκετές γλώσσες προγραμματισμού. Όλα τα παραπάνω καθιστούν μη αναγκαία τη λεπτομερή γνώση των διάφορων μορφών συγκεκριμένης σύνταξης της RDF. Όμως θα πρέπει να είμαστε τουλάχιστον σε θέση να αναγνωρίζουμε τις διάφορες μορφές δεδομένων RDF!

Σημείωση: Η απαιτηλή σειρά της αποθήκευσης.

Θα θυμάστε, βέβαια, ότι ένας γράφος RDF είναι ένα σύνολο τριάδων *χωρίς αλληλουχία*. Δεν υπάρχει διάταξη των τριάδων, ή αλλιώς, δεν υπάρχει κάποια τριάδα που να είναι πρώτη, δεύτερη ή τελευταία.

Όταν, όμως, έρθει η ώρα της αναπαράστασης σε αρχείο, αναγκαστικά επιβάλλεται κάποια σειρά αποθήκευσης: σε όποια μορφή και να γραφούν οι τριάδες, στο αρχείο θα έχουν κάποια διάταξη –κάποια θα αποθηκευτεί πρώτη, κάποια δεύτερη κ.ο.κ.⁵ Η διάταξη αυτή είναι *εντελώς τεχνητή* και δεν πρέπει με κανέναν τρόπο να υποθέσουμε ότι η σειρά που λαμβάνουμε τις τριάδες μας έχει κάποια σημασία!

Δείτε το επόμενο παράδειγμα, το οποίο περιέχει ορισμένες τριάδες από το αρχείο RDF του BBC που περιγράφει μια [ομάδα επεισοδίων](#) της σειράς «Doctor Who», με την ακριβή σειρά εμφάνισης των τριάδων αυτών (το πρόθεμα `bbcprog:` ισούται με το `http://www.bbc.co.uk/programmes/` και το `dc:` με το `http://purl.org/dc/elements/1.1/`):

```
bbcprog:b03jprmh#programme – dc:title – "The Firemaker"
bbcprog:b03jprmf#programme – dc:title – "The Forest of Fear"
bbcprog:b03jprmc#programme – dc:title – "The Cave of Skulls"
bbcprog:b03jprm9#programme – dc:title – "An Unearthly Child"
```

⁵Γι' αυτό εξάλλου οι μορφές αποθήκευσης ονομάζονται και μορφές σειριοποίησης (serialization)!

Θα πρέπει να υποθέσουμε λοιπόν, βασιζόμενοι στη σειρά εμφάνισης των τριάδων, ότι τα επεισόδια παίχτηκαν με τη σειρά "The Firemaker", "The Forest of Fear", "The Cave of Skulls", "An Unearthly Child";

Στην πραγματικότητα συμβαίνει ακριβώς το αντίθετο! Δείτε το προηγούμενο με μερικές ακόμα τριάδες από το ίδιο αρχείο, πάντα κατά σειρά εμφάνισης των τριάδων (το πρόθεμα `po:` ισοδυναμεί με το `http://purl.org/ontology/po/`):

```
bbcprog:b03jprmh#programme - dc:title - "The Firemaker"
bbcprog:b03jprmh#programme - po:position - "4"^^xsd:int
bbcprog:b03jprmf#programme - dc:title - "The Forest of Fear"
bbcprog:b03jprmf#programme - po:position - "3"^^xsd:int
bbcprog:b03jprmc#programme - dc:title - "The Cave of Skulls"
bbcprog:b03jprmc#programme - po:position - "2"^^xsd:int
bbcprog:b03jprm9#programme - dc:title - "An Unearthly Child"
bbcprog:b03jprm9#programme - po:position - "1"^^xsd:int
```

Τα φαινόμενα (και η σειρά αποθήκευσης των τριάδων) απατούν!

2.15 N-Triples και N-Quads

Η σύνταξη N-Triples ξεκίνησε ως «η γλώσσα των παραδειγμάτων» της παλιάς (και πλέον ξεπερασμένης) RDF 1.0. Στο πρότυπο εκείνο, η μορφή N-Triples ήταν ενσωματωμένη στο έγγραφο με τα παραδείγματα (test cases) της RDF. Σήμερα είναι επίσημα η πιο απλή μορφή *συγκεκριμένης σύνταξης* (concrete syntax) της RDF 1.1 με δικό της έγγραφο, όπου ορίζεται η [προδιαγραφή της](#).

Η σύνταξη N-Triples είναι πάρα πολύ απλή και επιτρέπει τη γρήγορη επεξεργασία μεγάλου αριθμού τριάδων RDF. Το κύριο χαρακτηριστικό της είναι ότι *κάθε γραμμή κειμένου αναπαριστά ακριβώς μία τριάδα*. Δεν χρειάζεται να αναλύσετε άλλες γραμμές, ούτε να περιμένετε να ολοκληρωθεί η σάρωση του αρχείου για να μάθετε την τριάδα της τρέχουσας γραμμής! Επιπλέον, η συντακτική ανάλυση είναι εύκολη: δεν χρειάζεστε υποχρεωτικά ειδικές βιβλιοθήκες λογισμικού για να γράψετε ή να διαβάσετε τις τριάδες. Για τους λόγους αυτούς, η μορφή N-Triples χρησιμοποιείται για *μαζική ανταλλαγή* δεδομένων RDF.

- *Κωδικοποίηση χαρακτήρων*: επιτρέπονται όλοι οι Unicode χαρακτήρες σε κωδικοποίηση utf-8. Προσέξτε ότι στην προηγούμενη RDF 1.0 όλοι οι χαρακτήρες, εκτός από τους ASCII, έπρεπε να κωδικοποιηθούν με ειδικό τρόπο!
- *Κατάληξη αρχείων*: προτείνεται η κατάληξη `.nt`.

- *Τύπος δεδομένων* (media type) για να επιλέξετε μορφή κατά την αίτηση ή την απόκριση: τυπικά είναι application/n-triples, αλλά πρέπει να είστε έτοιμοι να δεχτείτε και άλλους τύπους, όπως text/plain.

Η πολύ απλή σύνταξη N-Triples συνοψίζεται στα εξής:

1. Κάθε γραμμή περιέχει ακριβώς μία τριάδα στη μορφή
`s p o .`
 δηλαδή υποκείμενο, κατηγορημα και αντικείμενο, με κενά (space ή tab) να ακολουθούν καθένα από τα τρία μέρη, μια τελεία και έναν χαρακτήρα νέας γραμμής (newline).
2. Τα αναγνωριστικά URI γράφονται πάντα σε *απόλυτη μορφή* (το πλήρες URI) χωρίς συντομογραφικά προθέματα και μεταξύ < και >, π.χ.
`<http://dbpedia.org/resource/Lodovico_Giustini>`
3. Οι απλές τιμές (literals) γράφονται πάντοτε μεταξύ " και "
`"Linked Data"`
 μπορούν προαιρετικά να έχουν ένδειξη γλώσσας με τη χρήση του @
`"Semantic Web"@en`
 ή ένδειξη τύπου δεδομένων (Προσοχή: ο τύπος δεδομένων γράφεται με απόλυτο URI!)
`"1.663E-4"^^<http://www.w3.org/2001/XMLSchema#double>`
4. Οι ανώνυμοι κόμβοι έχουν πρόθεμα το `_:` και ακολουθεί μια ετικέτα (label)
`_:b1234`
 (η τελεία δεν μπορεί να εμφανίζεται στην αρχή ή στο τέλος της ετικέτας και το μείον δεν μπορεί να βρίσκεται στην αρχή της ετικέτας)

Σε γενικές γραμμές, τα παραπάνω καθορίζουν πλήρως τη σύνταξη N-Triples. Ακολουθεί ένα μικρό παράδειγμα στη μορφή αυτή, αλλά *προσοχή!* Οι τριάδες έχουν διπλωθεί σε περισσότερες από μία γραμμές για να χωρέσουν στη σελίδα, ενώ κανονικά αυτό *δεν επιτρέπεται!*

```
_:genid1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  
↪ <http://purl.org/NET/c4dm/timeline.owl#Interval> .
```

```
_:genid1 <http://www.w3.org/2000/01/rdf-schema#label> "Time interval for  
↪ a version of \"Reptiles and Amphibians\""
```

```
_:genid1 <http://purl.org/NET/c4dm/timeline.owl#timeline>  
↪ <http://www.bbc.co.uk/programmes/b00nj6cl#timeline> .
```

```
_:genid1 <http://purl.org/NET/c4dm/timeline.owl#duration>
  ↪ "PT3600S"^^<http://www.w3.org/2001/XMLSchema#duration> .

<http://www.bbc.co.uk/programmes/b00nj6cl#programme>
  ↪ <http://purl.org/ontology/fo/time> _:genid1 .
```

Η σύνταξη **N-Quads** επεκτείνει τη μορφή N-Triples για να μπορέσουν να αποθηκευτούν επώνυμοι γράφοι (named graphs), επιτρέποντας ένα τέταρτο μέλος, πριν από την τελεία, σε κάθε γραμμή. Το τέταρτο μέλος είναι προαιρετικό· αν δεν υπάρχει, η τριάδα ανήκει στον εξ ορισμού (default) γράφο.

- *Κωδικοποίηση χαρακτήρων*: Επιτρέπονται όλοι οι Unicode χαρακτήρες, σε κωδικοποίηση utf-8.
- *Κατάληξη αρχείων*: Προτείνεται η κατάληξη .nq.
- *Τύπος δεδομένων* (media type): application/n-quads, πριν από την εμφάνιση του προτύπου είχε προταθεί το text/x-nquads.

2.16 Turtle και TriG

Η **Turtle** (Terse RDF Triple Language) είναι υπερσύνολο της σύνταξης N-Triples. Συνεπώς, ένα έγκυρο έγγραφο N-Triples είναι αυτόματα και ένα έγκυρο έγγραφο Turtle, χωρίς βεβαίως να ισχύει το αντίθετο.

- *Κωδικοποίηση χαρακτήρων*: Unicode χαρακτήρες, σε κωδικοποίηση utf-8.
- *Κατάληξη αρχείων*: Προτείνεται η κατάληξη .ttl.
- *Τύπος δεδομένων* (media type): text/turtle. Σύμφωνα με το πρότυπο, όταν στέλνεται περιεχόμενο που δεν είναι ASCII, πρέπει πάντα να υποδεικνύεται text/turtle; charset=UTF-8.

Στη συνέχεια θα δούμε τις βασικές συντακτικές «ευκολίες» που προσθέτει η Turtle στη βασική μορφή N-Triples:

1. Δεν υπάρχει πλέον ο περιορισμός της μίας γραμμής ανά τριάδα.
2. Μπορούμε να χρησιμοποιήσουμε συντομογραφικά προθέματα στα URIs με τη βοήθεια του @prefix. Δείτε πώς μπορούμε να γράψουμε τις τριάδες του προηγούμενου παραδείγματος από τη σειρά «Doctor Who»:

```

@prefix dc: <http://purl.org/dc/elements/1.1/> .

<http://www.bbc.co.uk/programmes/b03jprmh#programme>
  dc:title "The Firemaker" .

<http://www.bbc.co.uk/programmes/b03jprmf#programme>
  dc:title "The Forest of Fear" .

<http://www.bbc.co.uk/programmes/b03jprmc#programme>
  dc:title "The Cave of Skulls" .

<http://www.bbc.co.uk/programmes/b03jprm9#programme>
  dc:title "An Unearthly Child" .

```

Παρατηρήστε ότι:

- Υπάρχουν δύο γραφές για τα προθέματα, το @prefix απαιτεί τερματισμό με τελεία, ενώ το εναλλακτικό PREFIX όχι.
 - Εδώ είναι νόμιμο μια τριάδα να μοιραστεί σε δύο γραμμές.
 - Δεν μπορείτε να γράψετε bbcprog:b03jprmh#programme γιατί ό,τι ακολουθεί το : του προθέματος δεν μπορεί να περιέχει παρά μόνο γράμματα, αριθμούς και άλλα (:) (τα τελευταία όχι στην αρχή)!
3. Συντακτική ευκολία, όταν έχουμε τριάδες με κοινό υποκείμενο. Τότε μπορούμε να συνοψίσουμε τις τριάδες με τη βοήθεια του ερωτηματικού (;) ως εξής:

```

PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX po: <http://purl.org/ontology/po/>

<http://www.bbc.co.uk/programmes/b03jprmh#programme>
  dc:title "The Firemaker" ;
  po:position 4 .

```

Το προηγούμενο είναι ισοδύναμο με:

```

PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX po: <http://purl.org/ontology/po/>

<http://www.bbc.co.uk/programmes/b03jprmh#programme>
  dc:title "The Firemaker" .

<http://www.bbc.co.uk/programmes/b03jprmh#programme>
  po:position 4 .

```

Παρατηρήστε ότι η σύνταξη Turtle αναγνωρίζει ορισμένες απλές τιμές χωρίς εισαγωγικά:

- το 4 ισοδυναμεί με "4"^^<http://www.w3.org/2001/XMLSchema#integer> ή ακόμα και με "4"^^xsd:integer αν έχει δηλωθεί το πρόθεμα xsd:.
- το 3.14 ισοδυναμεί με "-3.14"^^xsd:decimal.
- το 1.663E-4 ισοδυναμεί με "1.663E-4"^^xsd:double.
- το true ισοδυναμεί με "true"^^xsd:boolean

4. Παρομοίως, μπορούμε με το κόμμα (,) να συνοψίσουμε τριάδες με κοινό υποκείμενο και κατηγορημα:

```
PREFIX po: <http://purl.org/ontology/po/>

<http://www.bbc.co.uk/programmes/p00jxl63#programme> po:episode
  <http://www.bbc.co.uk/programmes/b03jprmh#programme>,
  <http://www.bbc.co.uk/programmes/b03jprm9#programme> .
```

το οποίο είναι ισοδύναμο με το:

```
PREFIX po: <http://purl.org/ontology/po/>

<http://www.bbc.co.uk/programmes/p00jxl63#programme> po:episode
  <http://www.bbc.co.uk/programmes/b03jprmh#programme> .

<http://www.bbc.co.uk/programmes/p00jxl63#programme> po:episode
  <http://www.bbc.co.uk/programmes/b03jprm9#programme> .
```

5. Οι ανώνυμοι κόμβοι μπορούν επίσης να γραφούν με συνοπτικό τρόπο:

```
PREFIX voc: <http://my.voc.ab/>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>

<http://universiti.es/ionio>
  voc:location [ geo:lat 39.620944;
                  geo:lon 19.923716 ] .
```

Το προηγούμενο είναι ισοδύναμο με:

```
PREFIX voc: <http://my.voc.ab/>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>

<http://universiti.es/ionio> voc:location _:bnode1 .

_:bnode1 geo:lat 39.620944;
          geo:lon 19.923716 .
```

Σημείωση: Οι λίστες που δεν είχε ποτέ η RDF.

Γνωρίζουμε ότι η RDF δεν υποστηρίζει κανένα είδος διάταξης, και κατά συνέπεια ούτε λίστες. Τι σημαίνει λοιπόν η παρακάτω (έγκυρη) Turtle;

```
PREFIX ex: <http://ex.com/>
```

```
ex:s ex:p (ex:o1 ex:o2 ex:o3) .
```

Το προηγούμενο ισοδυναμεί με:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX ex: <http://ex.com/>
```

```
ex:s ex:p _:genid1 .
_:genid1 rdf:first ex:o1 .
_:genid1 rdf:rest _:genid2 .
_:genid2 rdf:first ex:o2 .
_:genid2 rdf:rest _:genid3 .
_:genid3 rdf:first ex:o3 .
_:genid3 rdf:rest rdf:nil .
```

Παρατηρούμε ότι η Turtle χρησιμοποιεί το λεξιλόγιο της RDF (`rdf:first`, `rdf:rest` και `rdf:nil`) για να φτιάξει ένα είδος *διασυνδεδεμένης λίστας*! Προσοχή: αυτή η μορφή χρησιμοποιείται σπάνια στον πραγματικό κόσμο, λόγω της δυσκολίας χειρισμού (αναζήτηση και ενημέρωση) τέτοιων δομών.⁶

Η μορφή **TriG** επεκτείνει την Turtle για τον σχηματισμό επώνυμων γράφων. Η επέκταση επιτρέπει να συνδέσουμε μια ομάδα τριάδων με ένα URI γράφου:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
```

```
PREFIX po: <http://purl.org/ontology/po/>
```

```
PREFIX voc: <http://my.voc.ab/>
```

```
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
```

```
<http://example.com/graph1> #graph URI
{
  <http://www.bbc.co.uk/programmes/b03jprmh#programme>
    dc:title "The Firemaker" ;
```

⁶Επίσης, πολύ σπάνια χρησιμοποιούνται τα παρόμοια `rdf:Seq`, `rdf:Bag`, `rdf:Alt` που δεν καλύπτονται εδώ.

```

    po:position 4 .
}

GRAPH <http://example.com/graph2> # εναλλακτική μορφή
{

    <http://universiti.es/ionio>
        voc:location [ geo:lat 39.620944;
                        geo:lon 19.923716 ] .
}

```

Όπως φαίνεται στο προηγούμενο παράδειγμα, προηγείται το URI του γράφου και στη συνέχεια οι τριάδες που ανήκουν σε αυτόν, μέσα σε { }. Η δεσμευμένη λέξη GRAPH είναι προαιρετική και μπορεί να γραφεί με πεζά γράμματα. Αν θέλετε οι τριάδες να ανήκουν στον εξ ορισμού γράφο, παραλείψτε το URI του γράφου ή ακόμα και τα { } για τις τριάδες του.

- *Κωδικοποίηση χαρακτήρων*: Unicode χαρακτήρες, σε κωδικοποίηση utf-8.
- *Κατάληξη αρχείων*: Προτείνεται η κατάληξη .trig.
- *Τύπος δεδομένων* (media type): application/trig.




2.17 Άλλες μορφές αναπαράστασης RDF

Οι δύο μορφές αναπαράστασης δεδομένων RDF που παρουσιάζονται στην ενότητα αυτή διαφέρουν από τις προηγούμενες μορφές στο ότι δεν αναπτύχθηκαν μέσα στο πρότυπο RDF, ούτε εξυπηρετούν ως απλοί «τρόποι γραφής» της RDF. Αντιθέτως, η RDFa και η JSON-LD έχουν ευρύτερους σκοπούς.

Η μορφή **RDFa** στοχεύει στην αναπαράσταση σημασιολογικών δεδομένων, αναγνώσιμων από τις εφαρμογές, μέσα στο ίδιο έγγραφο HTML που απευθύνεται στον άνθρωπο. Με άλλα λόγια, το έγγραφο περιέχει και τα δύο είδη πληροφορίας. Μια εφαρμογή που αναγνωρίζει τη μορφή RDFa μπορεί να «διυλίσει» το μεικτό έγγραφο και να εξαγάγει τις τριάδες RDF. Ταυτόχρονας, ο χρήστης βλέπει μόνο το περιεχόμενο που προορίζεται γι' αυτόν.

Η RDFa υποστηρίζεται ενεργά από τις μεγάλες μηχανές αναζήτησης, τα on-line καταστήματα και τις εφαρμογές κοινωνικής δικτύωσης και βασίζεται στα λεξιλόγια που παρέχει το schema.org. Ως μορφή, βέβαια, μπορεί να υποστηρίξει οποιαδήποτε εφαρμογή, αρκεί να υπάρχει το κατάλληλο λεξιλόγιο.

Για παράδειγμα, στο σύστημα βιβλιοθήκης [Koha](http://koha.org), ενώ ο χρήστης βλέπει τα διαθέσιμα αντίτυπα ενός βιβλίου στην υποθετική διεύθυνση <http://op.ac/891> (σχήμα 2.8), μια εφαρμογή που μπορεί να φιλτράρει την πληροφορία RDFa («RDFa distiller») θα εξαγάγει τις παρακάτω τριάδες (δείχνουμε την πληροφορία για το πρώτο αντίτυπο μόνο):

Item type	Current location	Call number	Copy number	Status
 book	Κεντρικό Παράρτημα	938.301 ΘΕΟ (Browse shelf)	1	Available
 book	Κεντρικό Παράρτημα	938.301 ΘΕΟ (Browse shelf)	1	Available
 book	Κεντρικό Παράρτημα	938.301 ΘΕΟ (Browse shelf)	1	Available

Σχήμα 2.8: Εφαρμογή βιβλιοθήκης, τι βλέπει ο χρήστης

```
@prefix schema: <http://schema.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix gr: <http://purl.org/goodrelations/v1#> .
```

```
_:1 rdf:type schema:Offer;
    schema:seller "Κεντρικό Παράρτημα";
    schema:itemOffered <http://op.ac/891#record>;
    schema:businessFunction gr:LeaseOut;
    schema:sku "938.301 ΘΕΟ";
    schema:availability schema:InStock .
```

Με άλλα λόγια, υπάρχει μια «προσφορά» (ο ανώνυμος κόμβος `_:1`) του βιβλίου της σελίδας αυτής. Η προσφορά είναι διαθέσιμη, δανείζεται και ανήκει στην κεντρική βιβλιοθήκη με αριθμό "938.301 ΘΕΟ". Όχι και άσχημα από σημασιολογική άποψη –αρκεί να γνωρίζει η εφαρμογή σας τα λεξιλόγια `schema:` και `gr:`!

Πώς, όμως, η εφαρμογή έφτασε στις προηγούμενες τριάδες; Ας δούμε τον σχετικό κώδικα HTML (αφαιρώντας τα κομμάτια που δεν μας ενδιαφέρουν):

```
<tr vocab="http://schema.org/" typeof="Offer">
  <td property="seller">
    <link property="itemOffered" href="#record" />
    <link property="businessFunction"
      href="http://purl.org/goodrelations/v1#LeaseOut">
      Κεντρικό Παράρτημα
    </td>
  <td property="sku">
    938.301 ΘΕΟ
  </td>
```



```

<td>
  <link property="availability"
    href="http://schema.org/InStock" />
  Available
</td>

</tr>

```

Η πρώτη γραμμή δηλώνει το εξ ορισμού λεξιλόγιο `http://schema.org/` που θα χρησιμοποιηθεί στη συνέχεια σε οποιαδήποτε ιδιότητα μέσα στο `<tr>`, με τη βοήθεια του vocab:

```
<tr vocab="http://schema.org/" typeof="Offer">
```

Επίσης χρησιμοποιεί την ιδιότητα `typeof` της RDFa για να ορίσει ένα υποκείμενο χωρίς όνομα (άρα, έναν ανώνυμο κόμβο) του τύπου `Offer` (για την ακρίβεια, λόγω του εξ ορισμού λεξιλογίου, είναι το `schema:Offer`). Στις τριάδες που είδαμε προηγουμένως εμφανίζεται ως ανώνυμος κόμβος `_:1`.

Στη συνέχεια, οποιοδήποτε `property` μέσα στο περιεχόμενο του `<tr> ... </tr>` λειτουργεί ως κατηγορήμα του ανώνυμου κόμβου `_:1`. Το κατηγορήμα, αν δεν ορίζεται διαφορετικά, ανήκει στο εξ ορισμού λεξιλόγιο. Για παράδειγμα, οι επόμενες γραμμές

```

<td property="sku">
  938.301 ΘΕ0
</td>

```

παράγουν την τριάδα:

```
@prefix schema: <http://schema.org/> .
```

```
_:1 schema:sku "938.301 ΘΕ0" .
```

Στην προηγούμενη τριάδα, στη θέση αντικειμένου έχουμε την απλή τιμή (literal) `938.301 ΘΕ0`. Στο επόμενο παράδειγμα, όμως, τα πράγματα είναι διαφορετικά:

```

<td>
  <link property="availability"
    href="http://schema.org/InStock" />
  Available
</td>

```

Εδώ δεν χρησιμοποιείται το Available ως αντικείμενο· αντιθέτως, το τελευταίο παρέχεται από την ιδιότητα href (παρατηρήστε την εφαρμογή του λεξιλογίου schema: και στο αντικείμενο!):

```
@prefix schema: <http://schema.org/> .
```

```
_:1 schema:availability schema:InStock .
```

Κλείνοντας, θα πρέπει να σημειώσουμε ότι η προδιαγραφή της RDFa περιγράφει πολλά πρόσθετα χαρακτηριστικά που δεν θα παρουσιαστούν εδώ, όπως η χρήση του resource για τη δημιουργία κόμβων με URI, του datatype για τον ορισμό του τύπου των δεδομένων ή το prefix για τη χρήση πολλαπλών λεξιλογίων. Αν θέλετε να εμβαθύνετε στο θέμα, δείτε τα δύο μέρη του προτύπου, [RDFa Lite 1.1](#) και [RDFa Core 1.1](#).

Ενώ η RDFa στοχεύει στην ενσωμάτωση τριάδων RDF μέσα στις ιστοσελίδες, η μορφή **JSON-LD** εισάγει σημασιολογική πληροφορία στα Web APIs (Application Programming Interfaces). Τα τελευταία χρησιμοποιούν σήμερα σχεδόν αποκλειστικά τη μορφή **JSON** για την ανταλλαγή πληροφορίας. Η JSON έχει πολύ απλή μορφή και η ανάλυσή της γίνεται με μικρό κόστος επεξεργασίας. Από την άλλη πλευρά, όμως, δεν μεταφέρει μαζί με τα δεδομένα κανενός είδους σημασιολογία. Το κενό αυτό έρχεται να καλύψει η μορφή **JSON-LD**. Η JSON-LD δεν φτιάχτηκε ειδικά για τα δεδομένα RDF, καθώς μπορεί να μεταφέρει πληροφορία που δεν χωράει στο πρότυπο της RDF.

Στο επόμενο παράδειγμα, θα δούμε πάλι πώς περιγράφει το BBC τη σειρά «Doctor Who» σε μορφή JSON-LD⁷. Ζητώντας τη σχετική σελίδα σε μορφή JSON-LD, π.χ.

```
curl -s -i -H 'Accept: application/ld+json'
'http://www.bbc.co.uk/things/cbeab979-c95b-432e-a3bb-2b1d502f4db5#id'
```

θα λάβετε την εξής πληροφορία:

```
{
  "results": [
    {
      "@id":
↪ "http://www.bbc.co.uk/things/cbeab979-c95b-432e-a3bb-2b1d502f4db5#id",
      "@type": [
        "rdfs:Resource",
        "tagging:TagConcept",
        "tag:TagConcept",
        "ManagedThing",
```

⁷Όχι μόνο γιατί πρόκειται για σειρά-ορόσημο, αλλά και γιατί το BBC είναι από τους πρωτοπόρους στην τομέα των Συνδεδεμένων Δεδομένων (βλ. [2])!

```

        "http://purl.org/ontology/fo/Brand",
        "owl:Thing",
        "Thing"
    ],
    "disambiguationHint": "Long-running BBC programme",
    "label": "Doctor Who",
    "preferredLabel": "Doctor Who",
    "sameAs": "http://www.bbc.co.uk/programmes/b006q2x0#programme"
}
],
"@context": {
    "ManagedThing": {
        "@id": "http://www.bbc.co.uk/ontologies/cms/ManagedThing"
    },
    "Thing": {
        "@id": "http://www.bbc.co.uk/ontologies/coreconcepts/Thing"
    },
    "disambiguationHint": {
        "@id":
↪ "http://www.bbc.co.uk/ontologies/coreconcepts/disambiguationHint"
    },
    "label": {
        "@id": "http://www.w3.org/2000/01/rdf-schema#label"
    },
    "owl:Thing": {
        "@id": "http://www.w3.org/2002/07/owl#Thing"
    },
    "preferredLabel": {
        "@id":
↪ "http://www.bbc.co.uk/ontologies/coreconcepts/preferredLabel"
    },
    "rdfs:Resource": {
        "@id": "http://www.w3.org/2000/01/rdf-schema#Resource"
    },
    "results": {
        "@id": "@graph"
    },
    "sameAs": {
        "@id": "http://www.bbc.co.uk/ontologies/coreconcepts/sameAs",
        "@type": "@id"
    },
    "tag:TagConcept": {
        "@id": "http://www.bbc.co.uk/ontologies/tag/TagConcept"
    },
    "tagging:TagConcept": {
        "@id": "http://www.bbc.co.uk/ontologies/tagging/TagConcept"
    }
}

```

```

},
"type": {
  "@id": "http://www.w3.org/1999/02/22-rdf-syntax-ns#type",
  "@type": "@vocab"
}
}

```

Πρόκειται για ένα αντικείμενο με δύο ιδιότητες (results και @context). Το πρώτο μέρος results περιέχει τη ζητούμενη πληροφορία, έναν πίνακα με ένα μόνο στοιχείο:

```

{
  "@id":
  ↪ "http://www.bbc.co.uk/things/cbeab979-c95b-432e-a3bb-2b1d502f4db5#id",
  "@type": [
    "rdfs:Resource",
    "tagging:TagConcept",
    "tag:TagConcept",
    "ManagedThing",
    "http://purl.org/ontology/po/Brand",
    "owl:Thing",
    "Thing"
  ],
  "disambiguationHint": "Long-running BBC programme",
  "label": "Doctor Who",
  "preferredLabel": "Doctor Who",
  "sameAs": "http://www.bbc.co.uk/programmes/b006q2x0#programme"
}

```

- Η ιδιότητα @id μάς ειδοποιεί πως ό,τι ακολουθεί είναι URI. Εδώ πρόκειται για τον κόμβο που θα χρησιμοποιηθεί ως υποκείμενο των επόμενων τριάδων.
- Η ιδιότητα @type δηλώνει το είδος (κλάση) του κόμβου:

- rdfs:Resource,
- tagging:TagConcept,
- tag:TagConcept,
- ManagedThing,
- http://purl.org/ontology/po/Brand,
- owl:Thing και
- Thing.

Παρατηρήστε ότι η οντότητα που συμβολίζει το URI μπορεί να ανήκει σε πολλά είδη ταυτόχρονα!

- Στη συνέχεια, ακολουθούν οι υπόλοιπες ιδιότητες του κόμβου με τις αντίστοιχες τιμές τους, ορίζοντας ανά ζεύγος το κατηγορήμα και το αντικείμενο των τριάδων:

```
{
  "disambiguationHint": "Long-running BBC programme",
  "label": "Doctor Who",
  "preferredLabel": "Doctor Who",
  "sameAs": "http://www.bbc.co.uk/programmes/b006q2x0#programme"
}
```

Μισό λεπτό! Ακόμα κι αν δεχτούμε ότι η εφαρμογή μας μπορεί να καταλάβει τη γενική δομή του αντικειμένου, υπάρχουν κάποια θέματα:

- Εκτός από το `http://purl.org/ontology/ro/Brand`, πώς ξέρουμε ποιο URI έχουν οι υπόλοιποι τύποι; Τι είναι, για παράδειγμα, το `Thing`;
- Το ίδιο ισχύει για τις ιδιότητες του κόμβου: τι σημαίνουν (δηλαδή, ποιο URI έχουν) τα `disambiguationHint`, `label`, `preferredLabel` ή `sameAs`; Προφανώς ανήκουν σε κάποιο λεξιλόγιο, αλλά σε ποιο;
- Το αντικείμενο του `sameAs` είναι καταφανώς URI. Γιατί δεν χρησιμοποιήσαμε το `@id`;

Η λύση σε όλα τα προηγούμενα ερωτήματα βρίσκεται στο δεύτερο μέρος του εγγράφου: το `@context`. Το μέρος αυτό περιλαμβάνει όλη τη σχετική σημασιολογική πληροφορία. Η JSON-LD σχεδιάστηκε έχοντας το `@context` ξεχωριστό: μπορεί να προστεθεί εκ των υστέρων σε ένα απλό έγγραφο JSON, αναβαθμίζοντάς το αυτόματα σε σημασιολογικό έγγραφο JSON-LD. Έτσι οι παλαιότερες μη σημασιολογικές εφαρμογές μπορούν να συνεχίσουν να λειτουργούν, ενώ οι νεότερες εφαρμογές μπορούν να επωφεληθούν από τη ρητή σημασιολογία που παρέχεται.

Επιστρέφοντας στο παράδειγμά μας, το `@context` μάς λέει ότι ο τύπος `Thing` αντιστοιχεί στο URI

`http://www.bbc.co.uk/ontologies/coreconcepts/Thing`

με τον εξής τρόπο:

```
{
  "Thing": {
    "@id": "http://www.bbc.co.uk/ontologies/coreconcepts/Thing"
  },
}
```

Στη συνέχεια, δηλώνεται ότι το κατηγορημα `sameAs` αντιστοιχεί στο URI

`http://www.bbc.co.uk/ontologies/coreconcepts/sameAs`

και ότι επιπλέον το αντικείμενο του `sameAs` είναι επίσης URI:

```
{
  "sameAs": {
    "@id": "http://www.bbc.co.uk/ontologies/coreconcepts/sameAs",
    "@type": "@id"
  },
}
```

Το `results` είναι ο εξ ορισμού γράφος (ό,τι υπάρχει μέσα σε αυτόν μοιράζεται το κοινό `@context`):

```
{
  "results": {
    "@id": "@graph"
  },
}
```

Όπως και στην περίπτωση της RDFa, δεν θα καλύψουμε εδώ όλο το πρότυπο JSON-LD. Μπορείτε να ανατρέξετε στην [προδιαγραφή](#) του για περισσότερες πληροφορίες.

- *Κωδικοποίηση χαρακτήρων*: Unicode χαρακτήρες, σε κωδικοποίηση utf-8.
- *Κατάληξη αρχείων*: Προτείνεται η κατάληξη `.jsonld`.
- *Τύπος δεδομένων* (media type): `application/ld+json`.

Σημείωση: Τι συνέβη με την XML;

Την εποχή που οριστικοποιήθηκε η πρώτη έκδοση της RDF (1.0), η XML μεσουρανούσε ως το απόλυτο πρότυπο ανταλλαγής δεδομένων. Δεν είναι περίεργο λοιπόν που η τότε RDF απαιτούσε την παροχή των τριάδων σε μορφή XML! Η μορφή αυτή της RDF είναι υπερβολικά φλύαρη και πολύπλοκη. Δείτε κι εσείς:

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf      = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:po       = "http://purl.org/ontology/po/"
```

```

xmlns:dc      = "http://purl.org/dc/elements/1.1/">

<po:Series rdf:about="/programmes/p00jxl63#programme">
  <po:episode>
    <po:Episode rdf:about="/programmes/b03jprmh#programme">
      <dc:title>The Firemaker</dc:title>
      <po:position rdf:datatype="http://www.w3.org/2001/XMLSchema#int">4
    </po:position>
    </po:Episode>
  </po:episode>
  <po:episode>
    <po:Episode rdf:about="/programmes/b03jprmf#programme">
      <dc:title>The Forest of Fear</dc:title>
      <po:position rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3
    </po:position>
    </po:Episode>
  </po:episode>
  <po:episode>
    <po:Episode rdf:about="/programmes/b03jprmc#programme">
      <dc:title>The Cave of Skulls</dc:title>
      <po:position rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2
    </po:position>
    </po:Episode>
  </po:episode>
  <po:episode>
    <po:Episode rdf:about="/programmes/b03jprm9#programme">
      <dc:title>An Unearthly Child</dc:title>
      <po:position rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1
    </po:position>
    </po:Episode>
  </po:episode>
</po:Series>
</rdf:RDF>

```

Κάπου το έχουμε **ξαναδεί** αυτό το παράδειγμα, το θυμάστε; Η γενική ιδέα στη μορφή XML είναι ότι έχουμε μια σειρά από στοιχεία το ένα μέσα στο άλλο, σε εναλλασσόμενη μορφή: οντότητα – κατηγορημα – οντότητα – ..

Η μορφή XML της RDF έχει χάσει τη δημοτικότητά της με την πάροδο του χρόνου. Αν μη τι άλλο, η επεξεργασία ενός τέτοιου εγγράφου απαιτεί πολύ περισσότερους υπολογιστικούς πόρους από άλλες, πιο «ελαφριές» μορφές της RDF. Θα πρέπει, όμως, να είμαστε πάντα σε θέση να αναλύσουμε ένα τέτοιο έγγραφο: υπάρχουν ακόμα «εκεί έξω» αρχαίες εφαρμογές που μιλάνε μόνο αυτή τη διάλεκτο!

2.18 Σύνοψη

Στο κεφάλαιο αυτό γνωρίσαμε τις τριάδες και τους γράφους RDF: τη βασική μορφή πληροφορίας, πάνω στην οποία βασίζονται ο Σημασιολογικός Ιστός και τα Συνδεδεμένα Δεδομένα. Μάθαμε τι μπορεί να χρησιμοποιηθεί στα διάφορα μέρη της τριάδας (υποκείμενο, κατηγορημα, αντικείμενο): ισχυρά αναγνωριστικά URIs με παγκόσμια εμβέλεια, ανώνυμοι κόμβοι για τοπική χρήση και απλές τιμές (literals) που περιέχουν την τελική πληροφορία. Αναγνωρίσαμε –έστω και εισαγωγικά– τη μεγάλη σημασία των ευρέως γνωστών λεξιλογίων (vocabularies) για την αύξηση της σημασιολογικής αξίας των δεδομένων μας. Στη συνέχεια, ασχοληθήκαμε πιο τυπικά με το πρότυπο RDF, τον τρόπο οργάνωσης σε επώνυμους γράφους και τον χειρισμό των τύπων δεδομένων της RDF. Τέλος, γνωρίσαμε τις διάφορες μορφές αναπαράστασης δεδομένων RDF, από τις πιο απλές μορφές (N-Triples) μέχρι τις πιο σύνθετες RDFa και JSON-LD.

Οι τριάδες, όπως τις είδαμε στο κεφάλαιο, αυτό δεν διαθέτουν κανένα είδος σημασιολογικής δομής. Μπορείτε να δηλώσετε οτιδήποτε, χωρίς κανέναν περιορισμό. Αυτό μπορεί να οδηγήσει σε τριάδες χωρίς νόημα:

```
PREFIX ex: <http://ex.com/resource/>
PREFIX voc: <http://ex.com/vocab#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

ex:SemanticWeb  voc:teaches  ex:ComputerArchitecture .

ex:ClassRoom2  voc:startsAt  "12:00:00.0"^^xsd:time .
```

Στα επόμενα κεφάλαια θα εξερευνήσουμε τρόπους για να δηλώσουμε το *σχήμα* και να εκφράσουμε την *οργάνωση* των σημασιολογικών δεδομένων.

2.19 Βιβλιογραφικές αναφορές

[1] *RDF 1.1 Concepts and Abstract Syntax*, W3C Recommendation, 25 February 2014, διαθέσιμο από: <http://www.w3.org/TR/rdf11-concepts/>, ημερομηνία πρόσβασης: 25.6.2015.

[2] Bartlett Oliver. *Linked Data: Connecting together the BBC's Online Content*, 19 February 2013, διαθέσιμο από: <http://www.bbc.co.uk/blogs/internet/entries/af6b613e-6935-3165-93ca-9319e1887858>, ημερομηνία πρόσβασης: 25.6.2015.