



UNIVERSITÀ DEGLI STUDI ROMA TRE

Facoltà di Ingegneria  
Corso di Laurea in Ingegneria Informatica

Tesi Di Laurea

Questo è il titolo della tesi

Laureando

**Marco Faretra**

Matricola 460573

Relatore

**Prof. Paolo Atzeni**

Correlatore

**Ing. Luigi Bellomarini**

Anno Accademico 2016/2017

*Questa è la dedica*

# Ringraziamenti

Grazie a tutti

# Introduzione

Negli ultimi tempi, la mole di dati all'interno delle aziende cresce quotidianamente, per questo motivo molte compagnie desiderano mantenere i propri dati in knowledge graph, che per l'utilizzo e la gestione di tale strumento si necessita di un Knowledge Graph Management System (KGMS).

Fin dagli anni 70, l'importanza della conoscenza è stata evidente, e l'idea di salvare conoscenza e di elaborarla per trarre nuova conoscenza esisteva già da allora. Il collo di bottiglia era la tecnologia di quei tempi, gli hardware erano troppo lenti, la memoria principale troppo piccola; i DBMS erano troppo lenti e rigidi; Non era presente un web dove un sistema esperto poteva acquisire dati; il machine learning e le reti neurali furono ridicolizzate e non riuscite.

Con il passare degli anni, l'avvento tecnologico ha subito una crescita radicale, l'hardware si è evoluto, le tecnologie dei database sono migliorate notevolmente, è presente un web con dati aperti, le aziende possono partecipare sui social networks. La ricerca ha portato ad una comprensione migliore di molti aspetti nell'elaborazione della conoscenza e reasoning su grandi quantità di dati.

A causa di tutto ciò, migliaia di grandi e medie aziende, desiderano gestire i propri knowledge graph e cercano adeguati KGMS. Inizialmente soltanto le grandi aziende ad esempio Google (che utilizza il proprio knowledge graph per il proprio motore di ricerca), Amazon, Facebook, ecc... ne possedevano uno, ma con il passare degli anni molte aziende medio/basse desiderano avere un knowledge graph aziendale privato, che contiene molti dati in forma di fatti, come ad esempio conoscenza su clienti, prodotti, prezzi, concorrenti, piuttosto che conoscenza di tutto il mondo da Wikipedia o altre fonti simili.

Un KGMS completo deve svolgere compiti complessi di reasoning, ed allo stesso tempo ottenere performance efficienti e scalabili sui Big Data con una complessità computazionale accettabile. Inoltre necessita di interfacce con i database aziendali, il web e il machine learning. Il core di un KGMS deve fornire un linguaggio per rappresentare la conoscenza e il reasoning.

Vadalog rappresenta un sistema KGMS, che offre un motore centrale di reasoning principale ed un linguaggio per la gestione e l'utilizzo.

Il linguaggio Vadalog appartiene alla Famiglia Datalog $\pm$ , che estende Datalog con quantificatori esistenziali nelle teste delle regole, nonché da altre caratteristiche ed allo stesso tempo limita la sua sintassi in modo da ottenere decidibilità e tracciabilità dei dati.

Datalog [Wika] è un linguaggio di interrogazione per basi di dati che ha riscosso un notevole interesse dalla metà degli anni ottanta, è basato su regole di deduzione.

Il core logico di Vadalog è in grado di processare tale linguaggio ed è in grado di eseguire task di reasoning ontologici e risulta computazionalmente efficiente, tale da soddisfare i requisiti già citati (Big Data, Web, Machine Learning, ...), esso ha accesso ad un repository di regole. Per dare un esempio consente l'aggregazione attraverso la somma, il prodotto, il massimo, ecc... anche in presenza di ricorsioni.

Esso fornisce anche degli strumenti che permettono il data analytics, l'iniezione di codice procedurale, l'integrazione con diverse tipologie di input (ad esempio database relazionali, file csv, ecc...).

L'obiettivo della mia Tesi è stato l'ampliamento di Vadalog con nuove features.

Di seguito un accenno delle funzionalità di cui mi sono principalmente occupato, che verrà descritto in maniera più approfondita nei prossimi capitoli:

- Implementazione di nuovi tipi di dato.
- Riscritture per ottimizzare i tempi di calcolo (ad esempio "Push Selection Down").
- Creazione di benchmark per effettuare test sulle performance.
- Supporto di nuove funzionalità (ad esempio, supporto ai csv, supporto alle funzioni arbitrarie, ecc...).

- Integrazione di codice procedurale all'interno del linguaggio Vadalog.

L'attività di Tesi è stata svolta presso il Laboratorio Basi di Dati, dell'Università degli Studi Roma Tre, in collaborazione con L'Università di Oxford.

# Indice

<b>Introduzione</b>	<b>iv</b>
<b>Indice</b>	<b>vii</b>
<b>Elenco delle figure</b>	<b>viii</b>
<b>1 Vadalog Engine</b>	<b>1</b>
1.1 Questa è una Sezione . . . . .	1
1.1.1 Questa è una Sottosezione . . . . .	1
<b>2 Il linguaggio Vadalog</b>	<b>2</b>
2.1 Questa è una Sezione . . . . .	2
2.1.1 Questa è una Sottosezione . . . . .	2
<b>3 Algoritmi</b>	<b>4</b>
<b>4 Prove sperimentali</b>	<b>5</b>
<b>5 Related Work</b>	<b>6</b>
<b>Conclusioni e sviluppi futuri</b>	<b>7</b>
<b>Bibliografia</b>	<b>8</b>

# Elenco delle figure

2.1	SPQR-tree di un grafo. (a) L'albero di allocazione della faccia esterna. (b)	
	Il cammino notevole di cui si parla tanto nella Sezione 2.1. . . . .	3



# Capitolo 1

## Vadalog Engine

### 1.1 Questa è una Sezione

Prova di testo di capitolo. Anche l'opera di [Lar05] andrebbe riscritta interamente.

#### 1.1.1 Questa è una Sottosezione

Ancora del testo. Come si afferma in [JS96] molto lavoro deve ancora essere fatto.

## Capitolo 2

# Il linguaggio Vadalog

### 2.1 Questa è una Sezione

Prova di testo di capitolo. Vorrei citare qui tutta l'opera omnia di [oEE90, Wikb, Box97, AHPZ96].

#### 2.1.1 Questa è una Sottosezione

Ancora del testo

Come si evince dalle Figure 2.1.a e 2.1.b non si capisce molto.

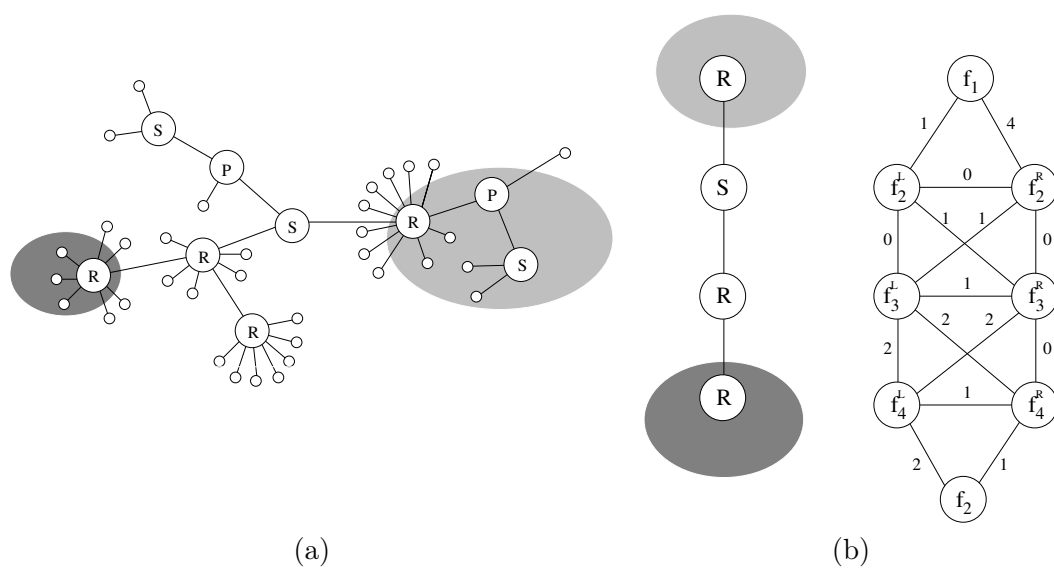


Figura 2.1: SPQR-tree di un grafo. (a) L'albero di allocazione della faccia esterna. (b) Il cammino notevole di cui si parla tanto nella Sezione 2.1.

## Capitolo 3

# Algoritmi

## Capitolo 4

# Prove sperimentali

## Capitolo 5

## Related Work

# Conclusioni e sviluppi futuri

La tesi è finita

# Bibliografia

- [AHPZ96] Eric Andonoff, Gilles Hubert, Annig Le Parc, and Gilles Zurfluh. Integrating versions in the omt models. In *ER '96: Proceedings of the 15th International Conference on Conceptual Modeling*, pages 472–487, London, UK, 1996. Springer-Verlag.
- [Box97] D. Box. *Essential COM*. Addison Wesley Professional, 1997.
- [JS96] Trevor H. Jones and Il-Yeol Song. Analysis of binary/ternary cardinality combinations in entity-relationship modeling. *Data Knowledge Engineering*, 19(1):39–64, 1996.
- [Lar05] C. Larman. *Applicare UML e i pattern - Analisi e progettazione orientata agli oggetti - 3a Edizione*. Prentice Hall, 2005.
- [oEE90] Institute of Electrical and Electronics Engineers. Ieee standard computer dictionary: A compilation of ieee standard computer glossaries, 1990.
- [Wika] Wikipedia. <https://it.wikipedia.org/wiki/Datalog>.
- [Wikb] Wikipedia. <http://en.wikipedia.org/wiki/Interoperability>.