Seminar 'Indizieren und Anfragen von Graphen in Datenbanken'

Silke Trißl

Humboldt-Universität zu Berlin, Institut für Informatik, 10099 Berlin
{trissl}@informatik.hu-berlin.de

Zusammenfassung In dem Seminar werden eine Reihe von Arbeiten zum Thema Graphen in Datenbanken besprochen. Wir betrachten sowohl die Art der Graphen, die in Datenbanken gespeichert wird als auch Indizierungs- und Anfragemöglichkeiten.
In Datenbanken werden sowohl Bäume, gerichtete azyklische Graphen als auch ungerichtete Graphen gespeichtert.
Diese Datenstrukturen werden zur effizienteren Anfragebeantwortung indiziert. Als Indizierungstechniken werden wir die transitive Hülle, verschiedene Nummerierungsschemata und Pfadindizierung kennen lernen. Auch sehen wir uns zum Schluss noch einige Anwendungen in der Biologie an.

Inhaltsverzeichnis

1 Einleitung und Latex

Das Seminar 'Indizieren und Anfragen von Graphen in Datenbanken' findet im Wintersemester 2007/08 als Blockseminar statt.

Der Termin für die Einführungsveranstaltung mit Vergabe der Themen ist **Dienstag, 23. Oktober** um **9 Uhr c.t.**. An diesem Termin wird auch der Termin für die Blockveranstaltung besprochen.

Die Themen für das Seminar sind in Abschnitt 2 dargestellt.

1.1 Scheinerwerb

Die Teilnehmer/innen des Seminars können am Ende einen Schein erhalten. Voraussetzungen dafür sind die aktive Teilnahme am Seminar, die Präsentation eines ausgewählten Themas, sowie die Anfertigung einer schriftlichen Ausarbeitung über dieses Thema (10-15 Seiten).

Die Anmeldung zum Seminar erfolgt ausschließlich über Goya.

Verpflichtend ist die Teilnahme an der Vorbesprechung sowie die Anwesenheit bei allen Vorträgen. Jeder Vortrag wird ca. 45 Minuten dauern und muss selbstständig erstellt sein. Der Vortrag kann sowohl auf Deutsch als auch auf Englisch gehalten werden.

Das Thema ist mit dem Betreuer bis zum **07. Dezember** zu *besprechen* und abzugrenzen. Die *Ausarbeitung* ist bis zum **21. Dezember 2007** abzugeben.

Am **08. Januar 2008** findet eine *Kurzvorstellung* (ca. 5 Minuten) der Themen durch die Teilnehmer statt, um einen Überblick über die Themen zu bekommen.

Die *Vorbesprechung* der Folien findet bis zum **08. Februar 2008** statt. Die voraussichtlichen Termine fr das *Blockseminar* sind der **12.** und **19. Februar 2008**

Termine für beide Vorbesprechungen (Thema & Folien) bitte rechtzeitig mit dem Betreuer abklären.

Die Ausarbeitung ist `ausschließlich` mit LateX zu erstellen.

Vorträge und Ausarbeitung können mit dem jeweiligen Betreuer der Arbeit nachbesprochen werden. Termine hierfür bitte individuell vereinbaren.

1.2 Kleine Tips

Für das Seminar sind Folien für den Vortrag zu erstellen. Dafür gibt es keine Vorgaben, unsere Empfehlung ist es allerdings, dies in Microsoft PowerPoint oder in OpenOffice zu tun. Für den Aufbau des Vortrags und auch der Ausarbeitung sind im Anschlußnoch einige Buchreferenzen für das Publizieren wissenschaftlicher Texte gegeben. Die Vorträge werden während des Blockseminars dann auf einem Windows-Rechner präsentiert.

Die Ausarbeitung ist in LaTeX zu machen. LaTeX ist eigentlich das einzige bekannte Satzsystem, mit dem man auch größere Texte und insbesondere mathematische Formeln bequem handhaben kann. In vielen Bereichen des wissenschaftlichen Publizierens wird diese System standardmäßig eingesetzt.

Unten ist ein on-line Tutorial angegeben, aber über Google findet man noch vieles mehr. Es gibt verschiedenste LaTeX-Bücher (auch in der Bibliothek) die da sicherlich auch weiterhelfen können.

– Ebel H.F., Bliefert C.
Schreiben und Publizieren in den Naturwissenschaften.
Wiley-Vch 1998.

– Rossig W. E., Prätsch J.
Wissenschaftliche Arbeiten
Weyhe 2005.

– Universität Gießen
Kochbuch für Latex http://www.uni-giessen.de/hrz/tex/cookbook/cookbook.html

– Mittelbach F.
The LaTeX Companion, w. CD-ROM
Addison-Wesley Professional 2004.

1.3 Kurz zu Latex

In Latex kann man eigentlich auch alles machen, was mit Word oder OpenOffice funktioniert. Tabellen sieht man in Tabelle 1, Abbildungen in Abbildung 1 und Formeln bei Formel 1.

Was wesentlich einfacher ist, ist am Ende das Literaturverzeichnis zu erstellen. Einfach eine eigene *.bib-Datei anlegen, die Arbeiten mit den Schlüsseln im Text referenzieren und schon ist das Literaturverzeichnis fertig.

Um nun aus einer *.tex-Datei eine *.pdf-Datei zu machen sind folgende Schritte notwendig:

```
> latex datei.tex
...
LaTeX Warning: There were undefined references.
Output written on datei.dvi (9 pages, 30160 bytes).
Transcript written on datei.log.
>
> bibtex datei
The style file: unsrt.bst
Database file #1: bibdatei.bib
>
> latex datei.tex
Output written on datei.dvi (9 pages, 30160 bytes).
Transcript written on datei.log.
>
> dvips datei.bib
 ... -> datei.ps
> ps2pdf datei.ps
```

Tabellen sind dazu gedacht, um Inhalt übersichtlich darzustellen.

Vorname, Nachname	Straße	Ort	Alter
Max Mustermann	Musterstraße 23	Berlin	24
Miriam Musterfrau	Merweg 52	Königswusterhausen	18

Tabelle 1. Eine Beispieltabelle.

Abbildungen , wie bei Abbildung 1 sind auch ganz einfach mit dem Package `graphix` und dem Befehl `includegraphics` einzubinden.

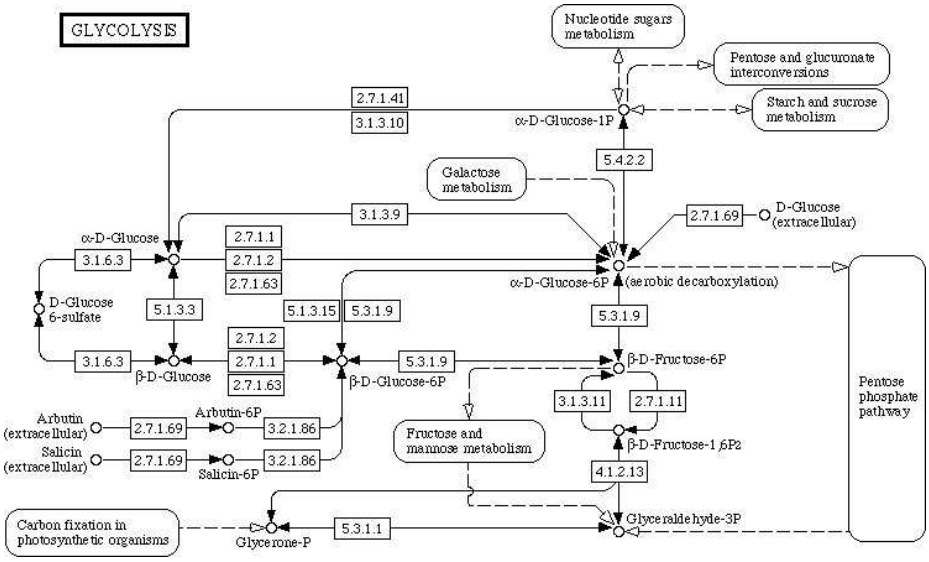

Abbildung 1. Die Glykolyse - der wichtigste Pathway beim Menschen.

Formeln - nicht erschrecken, Formel 1 müssen Sie nicht verstehen.

$$\int_{-1}^{1} \frac{\mathrm{d}x}{(a-x)\sqrt{1-x^2}} = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{\mathrm{d}t}{a-\sin t} = \frac{1}{2}\int_{-\pi}^{\pi} \frac{\mathrm{d}t}{a-\sin t} \tag{1}$$

Natürlich geht es auch etwas einfacher, dafür aber ohne Beschriftung:
$\frac{a+b}{a*b} * a^4 \leq c$

2 Themen

2.1 GraphDB, Complex Networks

Das Papier über GraphDB [19] und das Papier über Network biology [1] ist verbindliche Literatur für alle Studenten.

Erreichbarkeit

2.2 Transitive Hlle in der Datenbank

Die Berechnung der transitiven Hülle im Hauptspeicher ist bekannt. Lu stellt in [9] einige Möglichkeiten vor, die transitive Hülle in einer reltationalen Datanbank zu berechnen. Der logarithmische Ansatz ist dabei von besonderem Interesse. Ioannidis stellt dabei die optimale Strategie in [11] vor.

2.3 Nummerierungsschema

Grust und Kollegen [22] indizieren ein XML Dokument, indem sie Pre- und Postorderwerte verwenden. Diese Methode funktioniert nur auf Bäumen, daher stellt Agrawal und Kollegen [18] wird eine Möglichkeit vor, um DAGs und Graphen zu durch dieses Nummerierungsschema zu indizieren.

2.4 Dual Labeling

Die tranisitive Hülle für Graphen kann sehr groß werden. Daher zeigen Wang und Kollegen [8], dass sie mit Hilfe der Pre- und Postorderwerte die Größe der transitiven Hülle verkleinern können.

2.5 Transitive Hülle mit geometrischem Ansatz

Cohen et al. [5] haben beschrieben, wie man mit dem 2-Hop-Cover die transitive Hülle abbilden kann. Allerdings ist die Berechnung für das optimale Cover NP-complete. Sie haben schon eine Möglichkeit beschrieben, wie man das Cover möglichst gut berechnet. Cheng und Kollegen stellen in [13] verwenden einen anderen Ansatz, um ein sehr gutes Cover – jedoch nicht das optimale – zu berechnen.

Distanzen und Pfade

2.6 Distanz-Orakel

Distanzen auf ungerichteten Graphen mit gewichteten Kanten kann man entweder zur Anfragezeit berechnen oder durch einen Index bewantworten. Thorup & Zwick [15] stellen einen Index vor, der schnell berechnet werden kann und zur Anfragezeit schnell eine approximative Distanz liefert, die maximal $\delta * dist$ von der wahren Distanz abweicht.

2.7 Index für Distanzen mit Updates

Geerts und Kollegen [7] stellen einen kleinen Index für die exakte Beantwortung von Distanzanfragen. Sobald sich allerdings der Graph ändert, d.h. Knoten und Kanten hinzugefügt werden sollen, muss der Index geupdatet werden. Auch diesen Aspekt beleuchten sie in der Arbeit.

2.8 Straßennetzwerke I und II

Natürlich kann man auch Straßennetzwerke als Graphen ansehen. Knoten sind Kreuzungen, während Kanten Verbindungen zwischen den Kreuzungen darstellen. Die Kanten in einem solchen Netzwerk sind natürlich gelabelt, entweder mit Reisezeiten oder mit Entfernungen. Es gibt 2 verschiedene Ansätze dieses Netzwerk zu indizieren. Der erste Ansatz beruht auf einer Hierarchie von Straßen und wurde durch Sanders & Schultes [17] veröffentlicht. Der zweite Ansatz sieht sich die Topologie des Netzwerkes an und berechnet für eine Menge von Knoten alle Knoten, über die ein kürzester Weg führt. Dies wurde von Bast und Kollegen [2] beschrieben. In [3] wurden beide Methoden miteinander verglichen.

Anfrageoptimierung

2.9 Twig-Query Processing auf Bäumen

Diese Arbeite stellt Methoden vor, um XQueries in relationalen Datenbanksystemen zu beantworten. Bruno et al. [16] beschränkt sich dabei auf XML Dokumente, die Bäume darstellen.

2.10 XML Query Optimierung

XML Dokumente können in relationalen Datenbanksystemen gespeichert und indiziert werden. Werden nund XQuery Anfragen gestellt, die auch Joins enthalten, so möchte man natürlich diese Anfragen auch optimieren. Wu et al. [25] haben nun untersucht, ob die Methoden, die für relationale Datenbanksysteme entwickelt wurden auch für die Optimierung von XML Anfragen verwendet werden können. Natürlich benötigt man auch Informationen über die Größe der (Zwischen-)Ergebnisse, die sie in [26] untersuchten.

Suche nach Graphen

2.11 Closure-Trees

Gegeben viele Graphen in einer Datenbank, die sich ähnlich sind und einen Anfragegraphen. Um die Frage zu beantworten, welche Graphen in der Datenbank den Graphen enthalten, müssen die Graphen in der Datenbank indiziert werden. He & Singh stellen in [10] vor, wie man ähnliche Graphen in der Datenbank zusammenfassen kann für die Indexerstellung.

2.12 Häufige Subgraphen I und II

Im Gegensatz zu den Closure-Trees versuchen Yan et al. [24] und Cheng et al. [12] häufige Subgraphen in der Datenbank zu finden. Ihre Ansätze unterscheiden sich, daher werden auch beide Arbeiten im Seminar vorgestellt.

Graphen und Biologie

2.13 Graphanfragen - Sprachen

Wenn man Graphen anfragen will reicht SQL nicht mehr aus, um die Anfragen auszudrücken. Daher wurden verschiedene Möglichkeiten entwickelt, um Graphen in Datenbanken anzufragen. Leser [23] hat die Anfragesprache PQL vorgeschlagen, während Eckman & Brown [4] für die Graphanfragen das Datenbanksystem DB2 verwenden. Sohler et al. [6] stellen eine XML-basierte Methode vor.

2.14 QPath – Suche nach Pfaden

In [21] stellen die Autoren eine Möglichkeit vor, wie man nach Pfaden in einem Graphen suchen kann. Natürlich möchte man nicht unbedingt Pfade, die dem Anfragepfad genau entsprechen, sondern auch solche, die 'ungefähr' so aussehen, wie der Anfragepfad.

2.15 Biologische Netzwerke vergleichen

Jeder Organismus hat sein eigenes metabolisches und Protein-Protein Interaktionsnetzwerk. Hat man nun mehrere solcher Netzwerke möchte man diese auch miteinander vergleichen. Koyutürk und Kollegen [14] beschreiben, wie man gemeinsame Subgraphen in unterschiedlichen Netzwerken findet. Singh und Kollegen [20] gehen sogar noch einen Schritt weiter und versuchen die Netzwerke global miteinander zu alignieren.

Literatur

1. Albert-Ląszlǫ Barabąsi and Zoltąn N Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, Feb 2004.
2. Bast, Holger and Funke, Stefan and Matijevic, Domagoj. TRANSIT: Ultrafast Shortest-Path Queries with Linear-Time Preprocessing. In *Proceedings of the 9th DIMACS Implementation Challenge — Shortest Path*. DIMACS, 2006.
3. Bast, Holger and Funke, Stefan and Matijevic, Domagoj and Sanders, Peter and Schultes, Dominik. In Transit to Constant Time Shortest-Path Queries in Road Networks. In *Proceedings of the 9th Workshop on Algorithm Enginneering and Experiments (ALENEX)*, pages 46–59. SIAM, 2007.
4. Eckman, Barbara A. and Brown, P. G. Graph data management for molecular and cell biology. *IBM J. Res & Dev.*, 50(6):545 – 560, November 2006.
5. Edith Cohen and Eran Halperin and Haim Kaplan and Uri Zwick. Reachability and Distance Queries via 2-Hop Labels. *SIAM J. Comput.*, 32(5):1338–1355, 2003.
6. Florian Sohler and Daniel Hanisch and Ralf Zimmer. New methods for joint analysis of biological networks and expression data. *Bioinformatics*, 20(10):1517–1521, Jul 2004.
7. Floris Geerts and Peter Z. Revesz and Jan Van den Bussche. On-line Maintenance of Simplified Weighted Graphs for Efficient Distance Queries. In *Proceedings of the 14th ACM International Symposium on Geographic Information Systems (ACM-GIS 2006)*, pages 203–210. ACM Press, 2006.

8. Haixun Wang and Hao He and Jun Yang and Philip S. Yu and Jeffrey Xu Yu. Dual Labeling: Answering Graph Reachability Queries in Constant Time. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE)*, page 75. IEEE Computer Society, 2006.
9. Hongjun Lu. New Strategies for Computing the Transitive Closure of a Database Relation. In *Proceedings of the 13th International Conference on Very Large Data Bases (VLDB)*, pages 267–274. Morgan Kaufmann, 1987.
10. Huahai He and Ambuj K. Singh. Closure-Tree: An Index Structure for Graph Queries. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE)*, page 38. IEEE Computer Society, 2006.
11. Ioannidis, Yannis E. On the Computation of the Transitive Closure of Relational Operators. In *Proceedings of the 12th International Conference on Very Large Data Bases (VLDB)*, pages 403–411. Morgan Kaufmann, 1986.
12. James Cheng and Yiping Ke and Wilfred Ng and An Lu. FG-Index: Towards Verification-Free Query Processing on Graph Databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 857–872. ACM Press, 2007.
13. Jiefeng Cheng and Jeffrey Xu Yu and Xuemin Lin and Haixun Wang and Philip S. Yu. Fast Computation of Reachability Labeling for Large Graphs. In *Proceedings of the 10th International Conference on Extending Database Technology (EDBT)*, volume 3896 of *Lecture Notes in Computer Science*, pages 961–979. Springer, 2006.
14. Mehmet Koyutrk and Ananth Grama and Wojciech Szpankowski. An efficient algorithm for detecting frequent subgraphs in biological networks. *Bioinformatics*, 20 Suppl 1:i200–i207, Aug 2004.
15. Mikkel Thorup and Uri Zwick. Approximate Distance Oracles. *J. ACM*, 52(1):1–24, 2005.
16. Nicolas Bruno and Nick Koudas and Divesh Srivastava. Holistic Twig Joins: Optimal XML Pattern Matching. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 310–321. ACM Press, 2002.
17. Peter Sanders and Dominik Schultes. Highway Hierarchies Hasten Exact Shortest Path Queries. In *Proceedings of the 13th Annual European Symposium on Algorithms (ESA)*, volume 3669 of *Lecture Notes in Computer Science*, pages 568–579. Springer, 2005.
18. Rakesh Agrawal and Alexander Borgida and H. V. Jagadish. Efficient Management of Transitive Relationships in Large Data and Knowledge Bases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 253–262. ACM Press, 1989.
19. Ralf Hartmut Güting. GraphDB: Modeling and Querying Graphs in Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*, pages 297–308. Morgan Kaufmann, 1994.
20. Rohit Singh and Jinbo Xu and Bonnie Berger. Pairwise Global Alignment of Protein Interaction Networks by Matching Neighborhood Topology. In *Proceedings of the 11th Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, volume 4453 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2007.
21. Shlomi, Tomer and Segal, Daniel and Ruppin, Eytan and Sharan, Roded. QPath: a method for querying pathways in a protein-protein interaction network. *BMC Bioinformatics*, 7:199, 2006.
22. Torsten Grust and Maurice van Keulen and Jens Teubner. Accelerating XPath evaluation in any RDBMS. *ACM Trans. Database Syst.*, 29:91–131, 2004.
23. Ulf Leser. A query language for biological networks. *Bioinformatics*, 21 Suppl 2:ii33–ii39, Sep 2005.
24. Xifeng Yan and Philip S. Yu and Jiawei Han. Graph indexing based on discriminative frequent structure analysis. *ACM Trans. Database Syst.*, 30(4):960–993, 2005.

25. Yuqing Wu and Jignesh M. Patel and H. V. Jagadish. Structural Join Order Selection for XML Query Optimization. In *Proceedings of the 19th International Conference on Data Engineering (ICDE)*, pages 443–454. IEEE Computer Society, 2003.
26. Yuqing Wu and Jignesh M. Patel and H. V. Jagadish. Using histograms to estimate answer sizes for XML queries. *Inf. Syst.*, 28(1-2):33–59, 2003.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die Seminararbeit selbstständig und nur unter Zuhilfenahme der angegebenen Quellen angefertigt habe.

Berlin, 10. Oktober 2007