P4: Graphdaten-Verarbeitung

Prof. Dr. Ingo Claßen

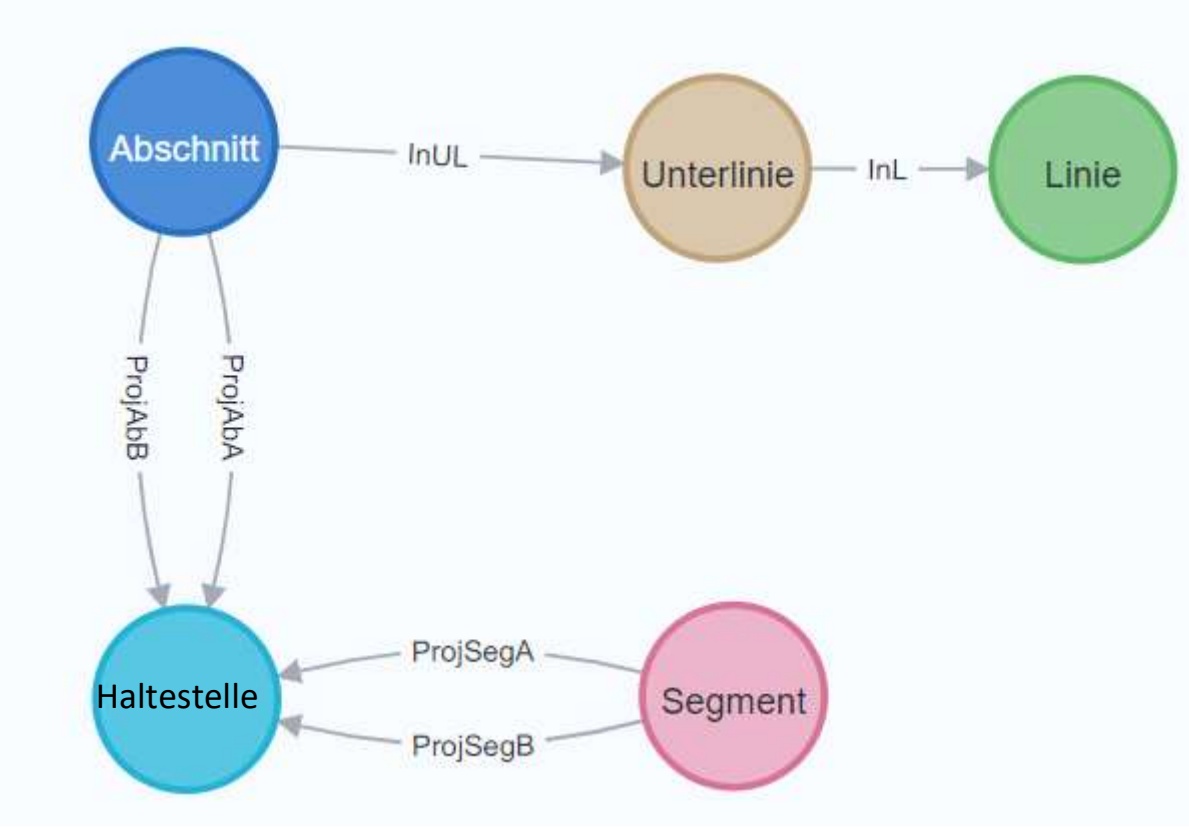
Mittwoch, 16. Februar 2022 12:01

P4-A1: Graph in Neo4j anlegen (1 Punkt)

Prof. Dr. Ingo Claßen

Mittwoch, 16. Februar 2022 12:01

Ziehen Sie relevante Daten aus der Postgres-Datenbank und legen Sie einen Graphen in Neo4j an, d.h. erzeugen Sie Knoten und Kanten zu folgendem Schema.



Schreiben Sie eine Cypher-Abfrage, die die Linien ermittelt die jeweils durch ein Segment gehen. Die Linien ergeben sich durch die Abschnitte, die durch das jeweilige Segment gehen. Diese sind über die Unterlinie mit der Linie verbunden.

"Start"	"Ende"	"Linien"
"Adenauerplatz"	"KonstanzerStr"	["U7"]
"Adenauerplatz"	"WilmersdorferStr"	["U7"]
"AfrikanischeStr"	"KurtSchumacherPlatz"	["U6"]
"AfrikanischeStr"	"Rehberge"	["U6"]
"Alexanderplatz"	"Jannowitzbrücke"	["U8"]

...

"Gleisdreieck"	"BülowStr"	["U2"]
"Gleisdreieck"	"KurfürstenStr"	["U1","U3"]
"Gleisdreieck"	"MendelssohnBartholdyPark"	["U2"]
"Gleisdreieck"	"Möckernbrücke"	["U1","U3"]
"Gneisenaustr"	"Mehringdamm"	["U7"]

Schreiben Sie eine Cypher-Abfrage, die die Unterlinienverläufe ermittelt. Dabei soll mit einem Plus- oder Minuszeichen gekennzeichnet werden, ob die Unterlinie in der jeweiligen Station hält oder nicht. Gehen Sie davon aus, dass in der ersten und letzten Haltestelle einer Unterline immer gehalten wird.

"Linie"	"Unterlinie"	"Verlauf"
"U1"	1017001	["+WarschauerStr","+SchlesischesTor","+GörlitzerBahnhof","+KottbusserTor","+PrinzenStr","+HalleschesTor","+Möckernbrücke","+Gleisdreieck","+KurfürstenStr","+Nollendorfplatz","+Wittenbergplatz","+Kurfürstendamm","+UhlandStr"]
"U1"	1017002	["+UhlandStr","+Kurfürstendamm","+Wittenbergplatz","+Nollendorfplatz","+KurfürstenStr","+Gleisdreieck","+Möckernbrücke","+HalleschesTor","+PrinzenStr","+KottbusserTor","+GörlitzerBahnhof","+SchlesischesTor","+WarschauerStr"]
"U1"	1017003	["+KottbusserTor","-PrinzenStr","+HalleschesTor","+Möckernbrücke","+Gleisdreieck","+KurfürstenStr","+Nollendorfplatz","+Wittenbergplatz","+Kurfürstendamm"]
"U1"	1017004	["+Kurfürstendamm","+Wittenbergplatz","+Nollendorfplatz","+KurfürstenStr","+Gleisdreieck","+Möckernbrücke","+HalleschesTor","-PrinzenStr","+KottbusserTor"]
"U2"	1018001	["+Pankow","+VinetaStr","+SchönhauserAllee","+EberswalderStr","+Senefelderplatz","+RosaLuxemburgPlatz","+Alexanderplatz","+KlosterStr","+MärkischesMuseum","+Spittelmarkt","+Hausvogteiplatz","+Stadtmitte","+MohrenStr","+PotsdamerPlatz","+MendelssohnBartholdyPark","+Gleisdreieck","+BülowStr","+Nollendorfplatz","+Wittenbergplatz","+ZoologischerGarten","+ErnstReuterPlatz","+DeutscheOper","+BismarckStr","+SophieCharlottePlatz","+Kaiserdamm","+TheodorHeussPlatz","+NewWestend","+OlympiaStadion","+Pub

P4-A4: Graphprojektion anlegen (1 Punkt)

Prof. Dr. Ingo Claßen

Mittwoch, 16. Februar 2022 12:10

Die folgenden Algorithmen setzen voraus, dass ein in-memory Abbild des Graphen aus der Datenbank heraus erzeugt wurde. Dieses Abbild wird aus dem tatsächlichen Graphen so herausprojiziert, dass die Algorithmen damit arbeiten können. Erstellen Sie eine passende Graph-Projektion.

Ermitteln Sie den Netzzusammenhang im Graphen.

"Haltestelle"	"Komponente"
"BerlinHauptbahnhof"	2
"BrandenburgerTor"	2
"Bundestag"	2
"TheodorHeussPlatz"	0
"Alexanderplatz"	0
"Bundesplatz"	0
"FrankfurterAllee"	0
"FriedrichStr"	0
"Gesundbrunnen"	0
"HeidelbergerPlatz"	0

Es sollen die kürzeste Pfade zwischen zwei Haltestellen ermittelt werden. Hier sollen zwei Varianten betrachtet werden: Anzahl Haltestellen und Gesamtlänge der Strecke. Als Beispiel soll die Verbindung zwischen HeidelbergerPlatz und KottbusserTor genommen werden.

Anzahl Stationen

"Verlauf"
["HeidelbergerPlatz","FehrbellinerPlatz","BlisseStr","BerlinerStr","Ba yerischerPlatz","EisenacherStr","Kleistpark","YorckStr","Möckernbrücke ","HalleschesTor","PrinzenStr","KottbusserTor"]

Länge

"Verlauf"	"KumulierteLaenge"
["HeidelbergerPlatz","FehrbellinerP latz","Hohenzollernplatz","Spichern Str","AugsburgerStr","Wittenbergpla tz","Nollendorfplatz","KurfürstenSt r","Gleisdreieck","Möckernbrücke"," HalleschesTor","PrinzenStr","Kottbu sserTor"]	[0.0,1211.0,1974.0,2515.0,3124.0,35 53.0,4366.0,4980.0,5757.0,6370.0,69 62.0,7967.0,8734.0]

P4-A7: Closeness Centrality (4 Punkte)

Prof. Dr. Ingo Claßen

Mittwoch, 16. Februar 2022 12:10

Ermitteln sie die Closeness Centrality im Netz in Bezug auf die Länge der Segmente. D.h., berechnen sie für jede Haltestelle die Summe der Längen der kürzesten Pfade zu allen anderen Haltestellen. Sortieren sie das Ergebnis aufsteigend nach den Summen.

Vergleichen sie dieses Ergebnis mit der Standard Closeness Centrality.

P4-A8: Betweeness Centrality (4 Punkte)

Prof. Dr. Ingo Claßen

Mittwoch, 16. Februar 2022 12:10

Ermitteln sie die Betweeness Centrality im Netz in Bezug auf die Länge der Segmente. D.h., berechnen sie für jede Haltstelle die Anzahl der kürzesten Pfade, die durch diese Haltestellen gehen. Sortieren sie das Ergebnis absteigend nach den Anzahlen.

Vergleichen sie dieses Ergebnis mit der Standard Betweeness Centrality.

P4-A9: Umsteigevorgänge (5 Punkte)

Prof. Dr. Ingo Claßen

Mittwoch, 16. Februar 2022 12:10

Da es im Verkehrsnetz der U-Bahn Linien gibt, kann es notwendig sein, dass bei einem Pfad von Haltestelle A nach Haltstelle B Umsteigevorgänge notwendig sind.
Schreiben Sie in Python eine Funktion die für einen gegeben Pfad die Umsteigepunkte ermittelt.