

Visualisierung von Argumentkarten - Inkrementelle Layouts

Diplomarbeit
von

Stefan Altmayer, Lukas Barth, Eric Braun,
David Hopf, Frederike Neuber

An der Fakultät für Informatik
Institute of Theoretical Computer Science

Betreuende Mitarbeiter: Ignaz Rutter

Bearbeitungszeit: 1st October 2014 – 31th March 2015

Inhaltsverzeichnis

1. Content Chapters	1
1.1. Einführung	1
1.2. Heuristische Ansätze	1
1.3. Ein Integer Linear Program	1
1.3.1. Implementation	1
1.3.1.1. User Interface	1
Literaturverzeichnis	5
Anhang	5
A. Appendix Section 1	5

1. Content Chapters

The content chapters of your thesis should of course be renamed. How many chapters you need to write depends on your thesis and cannot be said in general.

1.1. Einführung

Algorithmen beschreiben, Datenstrukturen, etc. ...

1.2. Heuristische Ansätze

...

1.3. Ein Integer Linear Program

...

1.3.1. Implementation

1.3.1.1. User Interface

In diesem Kapitel wird die grafische Oberfläche, welche Teil des Prototypen ist, mit ihren Funktionalitäten vorgestellt. Außerdem wird die Bedeutung für einen eigenen Editor im Rahmen des Seminars erläutert.

Beschreibung

Die GUI ist eine Webanwendung, welche mit Javascript programmiert wurde. Hierzu wurden die Javascript-Bibliotheken Cytoscape.js[?] und Angular.js[?] benutzt. Cytoscape.js bietet bereits Funktionalitäten, um einen Graphen darzustellen und zu manipulieren. Angular.js wurde als Javascript Framework benutzt, um die Interaktion mit Html und Javascript zu erleichtern. Ansonsten wurde reines Javascript benutzt, um die weiteren Features zu implementieren.

Da diese Arbeit sich auf das inkrementelle Einfügen von Knoten fokussiert, wurde der Editor mit einer Funktion ausgestattet, welche es erlaubt, einen Knoten lediglich mit ein paar Mausklicks einzufügen und die Größe festlegen zu können. Es wird verhindert, dass ein neu eingefügter Knoten mit einem bestehenden Knoten überlappt, indem für den neuen Knoten genug Platz geschaffen wird. Der Algorithmus um Platz zu schaffen unterstützt die vier folgenden Modi:

- es wird Platz gemacht, indem alle Knoten in eine der vier orthogonalen Richtungen verschoben werden, damit unmittelbar links, rechts, oberhalb und unterhalb von dem neuen Knoten kein anderer Knoten sich befindet. Alle Knoten, die in die gleiche Richtung verschoben werden, werden um den gleichen Betrag verschoben. Dieser Betrag wird so bestimmt, dass alle Knoten einen minimalen Abstand von dem eingefügten Knoten einhalten.
- der erste Modus aber es wird nur links und rechts Platz gemacht.
- der erste Modus aber es wird nur oben und unten Platz gemacht.
- die Knoten werden um den gleichen Betrag nach außen verschoben, damit kein anderer Knoten mit dem Neuen überlappt

In jedem Modus wird ein Mindestabstand zwischen dem neu eingefügten Knoten und allen anderen eingehalten. Somit kommt es zu keinen Überlappungen der Knoten, die zu nah beieinander sind. Abhängig vom Modus der gerade aktiv ist zeigt die GUI ein entsprechendes Overlay um anzuzeigen, wo überall Platz gemacht wird. Nachdem ein Knoten mit einem der vier Modi eingefügt wurde, kann dessen Größe festgelegt werden. Anschließend hat der Benutzer die Möglichkeit über einen Button das Layout des Graphen anzupassen. Somit wird der Platz, der vorher gemacht wurde, wieder minimiert, wobei versucht wird, gewisse Layoutbedingungen nicht zu verletzen (die wird näher in FIXME beschrieben). Des Weiteren zeigt jeder Knoten einen Titel und eine Beschreibung an. Somit können Informationen von Thesen und Argumenten in den Knoten angezeigt werden. Dies ist eine grundlegende Anforderung an eine Argumentkarte.

Außerdem können Knoten horizontal sowie vertikal auf einer Linie angeordnet werden. Dies hilft dem Benutzer, den Graphen nach seinen Bedürfnissen zu gestalten. Hier können verschiedene Funktionen genutzt werden, um neue Position der Knoten zu berechnen. In dem Prototypen wird der Mittelwert der x - oder y - Koordinate berechnet und als neue Position der auszurichtenden Knoten benutzt. Eine weitere Funktion könnte die Knoten an dem zuletzt markierten Knoten ausrichten oder an einem Knoten ausrichten, den man explizit auswählt. Da es sich nur um einen Prototypen handelt wurden diese Funktionen jedoch nicht implementiert.

Die letzte wichtige Funktion ist das Importieren von *.graphml* Dateien. Somit können Argumentkarten aus dem Argunet-Editor importiert werden. Dies erlaubt ein schnelles Testen der Algorithmen. Es werden Position und Größe der Knoten, Kanten zwischen den Knoten und die Beschriftung der Knoten importiert. Falls man ein orthogonales Kantenrouting implementiert hätte, könnte man hier auch noch die Position der Knicke auf den Kanten importieren, falls diese in der *.graphml*-Datei vorhanden sind.

Bedeutung für das Seminar

Die Anforderung der Philosophiestudenten, war, dass der Graph sich nur wenig ändern sollte, wenn ein Knoten eingefügt wird. Die verschiedenen Modi dienen dazu, dass der Benutzer entscheiden kann, wieviel sich im Graphen ändert. Hier muss angemerkt werden, dass nur der erste Modus garantieren kann, dass sich orthogonale Kanten nicht überlappen würden. Da diese Arbeit sich jedoch ausschließlich auf die Position der Knoten bezieht und die Kanten ausklammert, wird dieses Problem nicht weiter diskutiert.

Der anschließende Layoutalgorithmus versucht dann so weit es geht das ursprüngliche Layout wieder herzustellen. Wie ähnlich der Graph nach dem Einfügen des Knoten aussieht, hängt stark von der Position und der Größe des neuen Knoten ab. Die zusätzliche Ausrichtfunktion dient dazu, dass der Benutzer in das Layout eingreifen kann, und noch mehr mitbestimmen kann wie das endgültige Layout aussieht.

Diese beiden Funktionen sind sinnvoll, da der Benutzer, welcher eine Argumentkarte erstellt, meistens bereits weiß, wie die Knoten angeordnet werden sollen. Insbesondere weiß der Benutzer auch wie ein neu eingefügter Knoten positioniert werden soll. Der Benutzer ist jedoch nicht darauf angewiesen diese Layoutfunktionen zu benutzen. Ohne eine Anpassung vom Benutzer versucht der ILP-Algorithmus eine optimale (und somit in den meisten Fällen eine für den Benutzer wünschenswerte) Lösung zu finden. Die GUI bietet dem Benutzer Funktionen um schnell einen Graphen inkrementell aufzubauen und zugleich hat er die Freiheit selbst mit zu entscheiden, wie er den Graphen gestaltet.

Anhang

A. Appendix Section 1

ein Bild

Abbildung A.1.: A figure