

Übung 6

Ziel:

- Topologisches Sortieren, z.B. make Tool zum Kompilieren.

Aufgabe:

Schreiben Sie ein Programm `make.c`, welches eine Reihenfolge für das Kompilieren und Linken von Dateien ermittelt.

Das Programm soll dazu Abhängigkeiten von einer Datei einlesen. Abhängigkeiten könnten folgendermaßen definiert sein, wobei das Element vor dem Doppelpunkt von den Elementen nach dem Doppelpunkt abhängt:

`libTrees.a : tree.o element.o`

`tree.o : tree.h tree.c`

`element.o : element.c`

`main.o : main.c`

`libParser.a : parser.o lexer.o`

`parser.o : parser.c parser.h`

`lexer.o : lexer.c lexer.h`

`parser.c : grammar.g`

`lexer.c : grammar.g`

`parser.h : grammar.g`

`lexer.h : grammar.g`

`treeApp : libTrees.a libParser.a main.o`

Jedes Element soll einen Knoten repräsentieren. Definieren Sie eine Knotenstruktur in der Sie den Namen des Elements, die Startzeit, Endezeit der Bearbeitung und den Status der Bearbeitung (weiß, grau, schwarz) eintragen können.

Erstellen Sie anschließend einen Graphen mit Hilfe einer Adjazenzmatrix, in die Sie die Abhängigkeiten eintragen.

Implementieren Sie anschließend eine Tiefensuche und führen Sie damit die topologische Sortierung durch.

Geben Sie zuletzt die Reihenfolge aus in welcher die einzelnen Dateien erstellt werden müssen.