

Exercise for the Lecture  
**Graph Theory**  
Summer 2022  
Sheet 3

Please upload exactly one PDF file in Ilias. The submissions will **not** be graded. They serve only as feedback for the tutor.

**Exercise 1: Erreichbarkeit/Reachability** (5 Points)

In einem Digraph  $G$  ist ein Knoten  $v$  von einem Knoten  $u$  erreichbar, wenn  $u = v$  ist oder wenn es einen Pfad von  $u$  nach  $v$  gibt. *In a directed graph  $G$ , a node  $v$  is reachable from a node  $u$  if  $u = v$  or there exists a path from  $u$  to  $v$ .*

- Verändern Sie den Breitensuch-Algorithmus (BFS) aus der Vorlesung, so dass er auf Eingabe  $G$  und  $u, v \in V(G)$  entscheidet, ob  $v$  von  $u$  erreichbar ist. *Modify the Breadth-First Search (BFS) algorithm from the lecture to decide, given input  $G$  and  $u, v \in V(G)$ , whether  $v$  is reachable from  $u$ .*
- Verändern Sie entsprechend den Tiefensuch-Algorithmus (DFS). *Modify the Depth-First Search (DFS) algorithm accordingly.*

**Exercise 2: Erreichbarkeit in unendlichen Graphen/Reachability in infinite graphs**

In der künstlichen Intelligenz spielt das Erreichbarkeitsproblem in unendlichen Graphen eine große Rolle. Abweichend von der Vorlesung betrachten wir hier solche Graphen. *In artificial intelligence, the problem of reachability in infinite graphs is of great importance. Deviating from the lecture, we are considering such graphs.*

- Verwenden Sie ihren abgewandelten Tiefensuch- und Breitensuch-Algorithmus, um das Erreichbarkeitsproblem von  $v_0$  nach  $v_1$  in  $G$ :

$$V(G) := \{v_i | i \in \mathbb{N}_0\}$$
$$E(G) := \{(v_2, v_0)\} \cup \{(v_1, v_i) | i \in \mathbb{N}_0 \setminus \{1, 0\}\}$$

zu lösen. Terminieren Ihr Programme? Wenn nein, warum nicht?

*Use your modified depth-first search and breadth-first search algorithms to solve the reachability problem from  $v_0$  to  $v_1$  in  $G$ . Do your programs terminate? If not, why?*

- Bearbeiten Sie die gleiche Aufgabe für Erreichbarkeitsproblem von  $v_2$  nach  $v_3$  in  $F$ :

$$V(F) := \{v_i | i \in \mathbb{N}_0 \setminus \{0, 1\}\}$$
$$E(F) := \{(v_i, v_{i+2}) | i \in \mathbb{N}_0 \setminus \{0, 1\}\} \cup \{(v_i, v_{i+1}) | i \in \mathbb{N}_0 \setminus \{0, 1\}\}$$

*Perform the same task for the reachability problem from  $v_2$  to  $v_3$  in  $F$ .*

### Exercise 3: Kontrollflussgraph/Control Flow Graph

(5 Points)

In der Vorlesung wurde vorausgesetzt, dass jede Programmzeile eines Kontrollflussgraphens von der ersten Zeile  $r$  erreichbar ist. *In the lecture, it was assumed that every line of code in a control flow graph is reachable from the first line  $r$ .*

- a) Erweitern Sie die Definition von  $\text{dom}$  und  $\text{idom}$  für Programm Quellcode mit (evtl.) nicht erreichbaren Programmzeilen. *Extend the definition of  $\text{dom}$  and  $\text{idom}$  for program code with (possibly) unreachable lines of code.*
- b) Erstellen Sie ein Codebeispiel mit mindestens einer unerreichbaren Zeile. Erstellen Sie den zugehörigen Kontrollflussgraphen. *Create an code example where at least one line is unreachable. Create the corresponding control flow graph.*
- c) Verwenden sie den unveränderten Algorithmus aus der Vorlesung zum Berechnen von Dominatoren auf ihrem Beispielcode. Was wird für den unerreichbaren Teil des Programmes berechnet? Wie erklären Sie sich dieses Verhalten? *Use the unaltered algorithm from the lecture to calculate dominators on your sample code. What will be computed for the unreachable part of the program? How do you explain this behavior?*