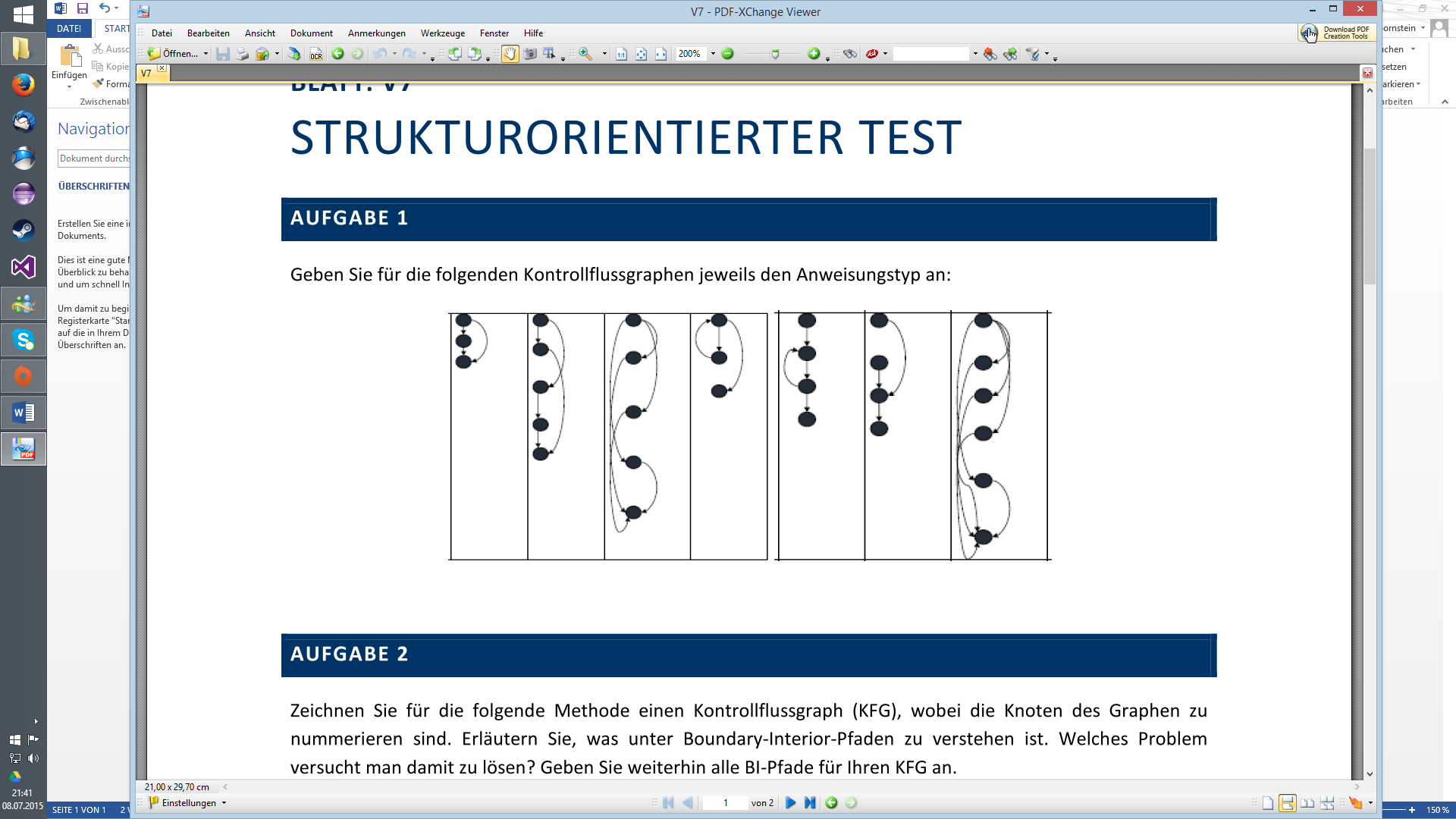
Aufgabe 1



1. n1: IF-Anweisung, n2: THENn3: Anweisung nach dem IF-Block
2. n1: IF-Anweisung, n2: THEN, n3: ELSEif, n4: ELSEif, n5: Anweisung nach IF-Block
3. IF, ELSEif-a, ELSEif-b, ELSEif-c, Anweisung nach IF
4. WHILE, Zykluskörperanweisung der WHILE-Schleife, Anweisung nach WHILE-Block
5. Beliebige Anweisung, DO-WHILE Schleifenstartanweisung, DO-WHILE-Testanweisung, Folgeanweisung
6. Kein Kontrollflussgraph, weil Knoten 2 nicht vom Start aus erreichbar
7. IF, ELSEif, ELSEif, ELSEif, ELSE, beliebige Anweisung nach dem IF-ELSE-Block

Aufgabe 2

*Kontrollflussgraph*

|  |  |
| --- | --- |
|  | **BI-Pfade:**  **n0n1n2n7**  **n0n1n2n3n4n6n2n7**  **n0n1n2n3n5n6n2n7**  **n0n1n2n3n4n6n2n3n4n6n2n7**  **n0n1n2n3n5n6n2n3n5n6n2n7**  **n0n1n2n3n4n6n2n3n5n6n2n7**  **n0n1n2n3n5n6n2n3n4n6n2n7** |

Beim *Boundary-interior-Pfadtest* wird eine Schleife getestet, einerseits werden die Grenzwerte (boundaries) betrachtet: Also die Schleife wird nicht betreten (1) und der Zykluskörper wird einmal ausgeführt (2), andererseits soll das Schleifeninnere (interior) ebenfalls getestet werden: Schleife wird einmal wiederholt (3).

Der Test hat den Sinn:

* den Programmablauf ohne Schleifendurchgang zu testen
* zu testen, was passiert, wenn die Schleife durchlaufen wird
* die Schleife zu testen, ob sie tut, was sie soll

Aufgabe 2

*C0 - Anweisungsüberdeckung*

Wir zählen nur die Knoten n1-n7:

Pfad ist **n1n2n3n5n6n2n7**

6 besuchte Knoten / 7 Knoten insgesamt =

*C1 – Zweiüberdeckung*

6 Zweige werden abgelaufen. Wir haben insgesamt 9 Zweige, also einen Zweigüberdeckungsgrad von .

*BI-Pfadüberdeckung*

Die While-Schleife wird nur einmal ausgeführt und einmal übersprungen. Damit ist ein Grenzfall abgedeckt, aber für die Gültigkeit der einmaligen Ausführung fehlt noch der Test des if-Zweiges.

Resultat

*C2-Einfache Bedinungsübedeckung*

¼ (if…)

2/2 (while…)

¼ (if …)

4/10=