



**Aufgabe 17 (Transformation kontextfreier Grammatiken [klausurähnlich], 3+3+2 Punkte)**

Gegeben sei jeweils folgende Grammatik ( $\{\text{Start}, A, B, C\}$ ,  $\{a, b\}$ , Regeln, Start) mit den Regeln

- a)  $\text{Start} \rightarrow aAB$      $A \rightarrow ABBA \mid B$      $B \rightarrow b \mid \varepsilon$

Geben Sie eine sprachäquivalente Grammatik (Regelmenge) ohne  $\varepsilon$ -Regeln an.

- b)  $\text{Start} \rightarrow AB$      $A \rightarrow ABBA \mid B$      $B \rightarrow b \mid A$

Geben Sie eine sprachäquivalente Grammatik (Regelmenge) ohne Ketten-Regeln an.

- c)  $\text{Start} \rightarrow AB$      $A \rightarrow ABBA \mid a$      $B \rightarrow b \mid AAA$

Geben Sie eine sprachäquivalente Grammatik (Regelmenge) in Chomsky-Normalform an.

**Aufgabe 18 (Transformation kontextfreier Grammatiken [klausurähnlich], 3+3+2 Punkte)**

Gegeben sei jeweils folgende Grammatik ( $\{\text{Start}, A, B, C\}$ ,  $\{a, b\}$ , Regeln, Start) mit den Regeln

- a)  $\text{Start} \rightarrow ABC$      $A \rightarrow AAB \mid \varepsilon$      $B \rightarrow b \mid \varepsilon$      $C \rightarrow CC \mid c$

Geben Sie eine sprachäquivalente Grammatik (Regelmenge) ohne  $\varepsilon$ -Regeln an.

- b)  $\text{Start} \rightarrow ABC$      $A \rightarrow BB$      $B \rightarrow b \mid A$      $C \rightarrow c \mid A$

Geben Sie eine sprachäquivalente Grammatik (Regelmenge) ohne Ketten-Regeln an.

- c)  $\text{Start} \rightarrow ABC$      $A \rightarrow ABC \mid a$      $B \rightarrow b \mid ABC$      $C \rightarrow ABC \mid c$

Geben Sie eine sprachäquivalente Grammatik (Regelmenge) in Chomsky-Normalform an.

**Aufgabe 19 (Grammatiktransformationen)**

- Geben Sie Ihre Lösung aus 17 a) in die Datei KFGOhneEps.kfg ein und prüfen Sie mit test.kontextfreieGrammatik.KFGOhneEpsTest.java ob Ihre Lösung richtig sein könnte.
- Geben Sie Ihre Lösung aus 17 b) in die Datei KFGOhneKette.kfg ein und prüfen Sie mit test.kontextfreieGrammatik.KFGOhneKetteTest.java ob Ihre Lösung richtig sein könnte.
- Geben Sie Ihre Lösung aus 17 c) in die Datei KFGInChomsky.kfg ein und prüfen Sie mit test.kontextfreieGrammatik.KFGInChomskyTest.java ob Ihre Lösung richtig sein könnte.
- Geben Sie Ihre Lösung aus 18 a) in die Datei KFGOhneEps2.kfg ein und prüfen Sie mit test.kontextfreieGrammatik.KFGOhneEps2Test.java ob Ihre Lösung richtig sein könnte.
- Geben Sie Ihre Lösung aus 18 b) in die Datei KFGOhneKette2.kfg ein und prüfen Sie mit test.kontextfreieGrammatik.KFGOhneKette2Test.java ob Ihre Lösung richtig sein könnte.
- Geben Sie Ihre Lösung aus 18 c) in die Datei KFGInChomsky2.kfg ein und prüfen Sie mit test.kontextfreieGrammatik.KFGInChomsky2Test.java ob Ihre Lösung richtig sein könnte.
- Formen Sie folgende Grammatiken schrittweise in die Chomsky-Normalform um. Schreiben Sie Ihr Ergebnis in die angegebene Datei und nutzen Sie die angegebene Testklasse um zu prüfen, ob Ihr Ergebnis richtig sein kann.

	Grammatik, Start ist X	Datei	Test im Paket test.kontextfreieGrammatik
I.	$X \rightarrow A \mid B$ $A \rightarrow AAA \mid aBBB$ $B \rightarrow b \mid \varepsilon$	KFGTrafo1.kfg	KFGTrafo1Test.java
II.	$X \rightarrow aXa \mid A$ $A \rightarrow bAb \mid B$ $B \rightarrow \varepsilon$	KFGTrafo2.kfg	KFGTrafo2Test.java
III.	$X \rightarrow aaX \mid A$ $A \rightarrow bbA \mid B$ $B \rightarrow ccX \mid \varepsilon$	KFGTrafo3.kfg	KFGTrafo3Test.java



**Aufgabe 20 (👉 programmatische Konstruktion kontextfreier Grammatiken; Ansatz wird auf Folgeblättern genutzt)**

Kontextfreie Grammatiken können in der gegebenen Bibliothek auch rein aus Objekten schrittweise konstruiert werden. Lesen Sie dazu Kapitel 4.2 der theoriesammlung-Doku.

- Schreiben Sie eine Grammatik für die Sprache  $\{a^n b^{n+2} \mid n \geq 0\}$  ausschließlich aus Objekten bestehend in Java, also ohne Textdatei oder String-Variable mit der Grammatik als Text in einem Beispielprogramm und lassen Sie sich einen zu Ihrer Grammatik gehörenden Ableitungsbaum für das Wort aabbbb in eine JPG-Datei generieren.
- Die in der Vorlesung vorgestellte Grammatik zu einer Programmiersprache steht in der Datei „Programmiersprache.kfg“. Schreiben Sie ein Programm in dieser Sprache mit dem die Werte in den Variablen x1 und x2 (vom Typ 2) vertauscht werden. Zeigen Sie mit der Java-Bibliothek, dass Ihr Programm syntaktisch korrekt ist und lassen Sie sich ein JPG mit einem Ableitungsbaum generieren. Die Grammatik können Sie natürlich aus der Datei einlesen.

**Aufgabe 21 (👉 eigene Umsetzung von Transformationsalgorithmen; aufwändig)**

In der gegebenen Bibliothek sind bereits die Transformationsalgorithmen aus der Vorlesung umgesetzt. Dabei wurden die Algorithmen nur prototypisch ohne jedwede Optimierung programmiert. Lassen Sie eine Klasse von KontextfreieGrammatik.java erben und überschreiben Sie die folgenden oder einen der folgenden Algorithmen.

**public boolean** ohneLeeresWortRegeln(): transformiert die Grammatik sprachäquivalent so, dass es keine Regeln mit dem leeren Wort auf der rechten Seite gibt; das Ergebnis gibt an, ob das leere Wort von der Grammatik erzeugt werden kann

**public KontextfreieGrammatik** entferneKettenregeln(): transformiert die Grammatik sprachäquivalent so, dass es keine Kettenregeln mehr gibt; das Ergebnis ist this

**public KontextfreieGrammatik** inChomskyNormalform(): transformiert die Grammatik sprachäquivalent in eine Grammatik in Chomsky-Normalform; das Ergebnis ist this

Tragen Sie Ihre Klasse in test.kontextfreieGrammatik.Optimierungstest.java in Zeile 22 ein und lassen Sie das Programm laufen. Es wird versucht, die Korrektheit zu prüfen und an Beispielen grob die Performance analysiert.

Sie können zur Bearbeitung auch die Klasse grammatik.OptimierteGrammatik.java nutzen, die einfach eine Kopie der existierenden Algorithmen enthält, die überarbeitet oder völlig neu konzipiert werden können. Anregungen für ein neues Konzept können z. B. dem Skript von Karsten Morisse entnommen werden, da es dort meist leicht abgewandelte Ansätze gibt.

Schwankungen bis mindestens 3% können durch die Laufzeitumgebung auftreten.