Gruppe $\boxed{\mathbf{3}}$
Formale Systeme und Automaten

Phil Pützstück, 377247
Benedikt Gerlach, 376944
Sebastian Hackenberg, 377550

Hausaufgabe 11

Aufgabe 5

Aufgabe 6

a)

Wir markieren folgende Nichtterminale in angegebener Reihenfolge: C, A, D, G
Wir können keine weitern Nichtterminale markieren, da sie eine zirkuläre Abhängigkeit haben:
Um S zu markieren, müssten wir B oder F markiert haben.
Um B zu markieren, müssten wir S oder F markiert haben.
Um F zu markieren, müssten wir B markiert haben.

b) Insbesondere ist die Sprache damit leer, da S kein terminierendes Nichtterminal ist.

Aufgabe 7

a	A,B,C	S	A	S,B	A	S,B
	b	B,C		A	S	A
		c	C	S	A	S
			a	A,B,C	S	A
				b	B,C	
					c	C

c	C			S	A
	c	C	S	A	S
		a	A,B,C	S	A
			c	C	
				c	C

Damit gilt $abcabc \in L(\mathcal{G})$ und $ccacc \notin L(\mathcal{G})$.

Aufgabe 8

Sei eine kontextfreie Grammatik $\mathcal{G} = (N, \Sigma, P, S)$ gegeben. Wenn $|L(\mathcal{G})| = \infty$, so muss ein Wort $z \in L(\mathcal{G})$ mit $|z| \geq 2^{|N|+1}$ existieren, sodass der Ableitungsbaum von z mindestens Höhe $h \geq |N|+1$ hat, also eins der endlich vielen Nichtterminale mehr als einmal in der Ableitung von z vorkommt (Idee Pumping-Lemma). Andererseits haben wir nun endlich viele Nichtterminale und damit endlich viele Ableitungen, in denen jedes Nichtterminal höchstens ein mal vorkommt. Weiter gilt auch, dass wenn ein $z \in L(\mathcal{G})$ mit $|z| \geq n = 2^{|N|+1}$ existiert (wobei n eben genau das n aus dem Pumping-Lemma ist), dass dann das Pumping Lemma uns einer Zerlegung liefert, welche wir insebsondere beliebig oft "pumpen" können, um unendlich viele verschiedene Wörter in $L(\mathcal{G})$ zu erhalten.
Insgesamt haben wir also gezeigt, dass:

$$\exists z \in L(\mathcal{G}) : |z| \geq 2^{|N|+1} \qquad \Longleftrightarrow \qquad |L(\mathcal{G})| = \infty$$

Nun können wir wie folgt entscheiden, ob $L(\mathcal{G})$ zu einer gegebenen Grammatik $\mathcal{G}$ unendlich ist: Es ist Σ endlich, also ist $\Sigma^n\Sigma^*$ ein regulärer Ausdruck, welcher alle Wörter über dem Eingabealphabeit mit mindestens Länge $n := 2^{|N|+1}$ erkennt. Dann können wir einen DFA $\mathcal{A}$ erstellen mit $L(\mathcal{A}) = L(\Sigma^n\Sigma^*)$. Weiter können wir aus der gegebenen Grammatik $\mathcal{G}$ einen PDA $\mathcal{B}$ mit $L(\mathcal{B}) = L(\mathcal{G})$ erstellen, welcher mit leerem Zustand akzeptiert. Dann können wir den PDA $\mathcal{C}$ erstellen, welcher die Produktkonstruktion der beiden darstellt. Also $L(\mathcal{C}) = L(\mathcal{A}) \cap L(\mathcal{B})$. Zu diesem können wir nun wieder eine kontextfreie Grammatik $\mathcal{G}'$ erstellen, sodass $L(\mathcal{G}') = L(\mathcal{C})$. Ferner ist also $L(\mathcal{G}') = \{z \mid z \in L(\mathcal{G}) \wedge |z| \geq n\}$. Wenn wir also mit dem Markierungsalgoritmus das Leerheitsproblem auf $\mathcal{G}'$ lösen, so wissen wir, ob die Sprache $L(\mathcal{G})$ ein Wort der Länge $\geq n$ enthält. Nach dem obigen Beweis ist dann $L(\mathcal{G}') \neq \varnothing \implies |L(\mathcal{G})| = \infty$.
Somit können wir mit dem angegebenen Algorithmus testen, ob die Sprache einer gegebenen kontextfreien Grammatik unendlich ist.