

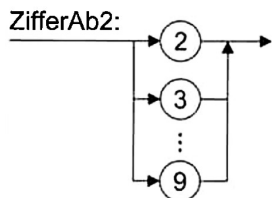
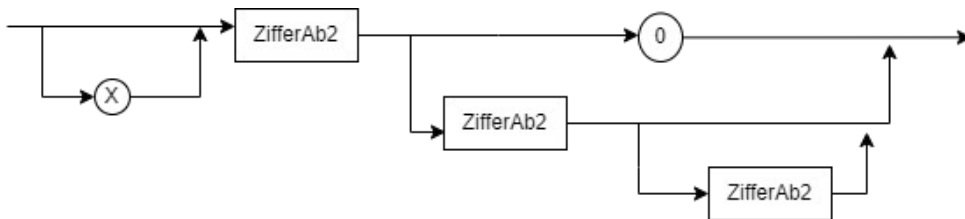
Informatik Abitur Bayern 2023 / IV - Lösung

Autoren:
 Unsin (1)
 Winter (2, 3)

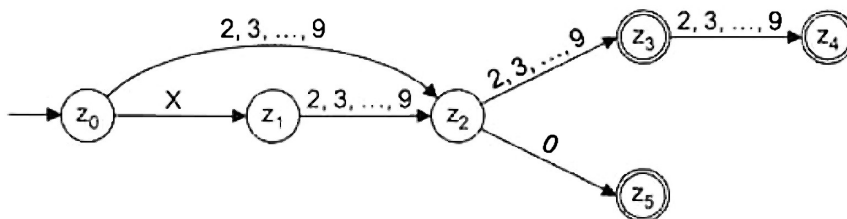
1a Beobachtung: Im Diagramm ist die 3. Ziffer in einer Schleife enthalten, es können also UN-Nummern beliebiger Länge erzeugt werden. 2
 z.B. 346856 kann mit dem Diagramm erzeugt werden, entspricht aber nicht der Beschreibung, weil UN-Nummern immer vierstellig sind

Produktionsregel	Erklärung
Gefahrtafel = Gefahrnummer"/"UN-Nummer;	Die Gefahrtafel besteht aus Gefahrnummer und UN-Nummer, getrennt durch einen Schrägstrich.
UN-Nummer = Ziffer Ziffer Ziffer Ziffer;	Die UN-Nummer ist immer vierstellig, wobei jede Ziffer einschließlich 0 vorkommen darf.
Ziffer = „0“, „1“, „2“, „3“, „4“, „5“, „6“, „7“, „8“, „9“;	Alle Zahlen von 0 bis 9
Gefahrnummer = [„X“] ZifferAb2 ((ZifferAb2 [ZifferAb2]) „0“) ;	Die Gefahrnummer kann mit dem Buchstaben X beginnen. Sie enthält zwei bis drei Ziffern ohne 0 und 1, d.h. Ziffern ab 2. Die zweite Ziffer kann eine 0 sein, in diesem Fall folgt keine weitere Ziffer.
ZifferAb2 = „2“, „3“, „4“, „5“, „6“, „7“, „8“, „9“;	Ziffer ohne 0 und 1, d.h. alle Ziffern ab 2.

Diagramme:
 Gefahrnummer:



1c Die Zustände werden mit z_0 bis z_5 benannt. 5



1d Die Syntax legt durch Regeln fest, wie Worte gebildet werden können. Anhand der Syntax kann entschieden werden, ob ein Wort zur Sprache gehört oder nicht. Die Semantik ist die Bedeutung der Sprache, z.B. welche Eigenschaften und Gemeinsamkeiten haben die Worte. Laut Angabe reagiert Benzin nicht auf gefährliche Weise mit Wasser. Syntaktisch wäre es jedoch möglich, die Gefahrnummer trotzdem mit einem „X“ zu versehen, d.h. X33/1203 wäre syntaktisch korrekt aber semantisch fehlerhaft. 3

2a Beispiel: Ist nur noch ein Set im Lager vorhanden und werden beide Roboter beauftragt, ein Set zusammenzustellen, sehen zu diesem Zeitpunkt beide Roboter, dass es möglich ist und starten den Auftrag. Nimmt nun der erste Roboter zufällig zwei Gegenstände und der andere Roboter zufällig die anderen beiden Gegenstände, kommt es zu einem Deadlock. Beide Roboter warten zyklisch darauf, dass die Gegenstände des anderen Roboters wieder zur Verfügung stehen, da die beiden Roboter aber auch für die Einsortierung zuständig sind, wird es niemals passieren, dass Gegenstände verfügbar werden. 3

2b Deadlocks können verhindert werden, wenn eine der vier Bedingungen nach Coffman et al. Nicht mehr erfüllt ist. HOLD AND WAIT könnte ausgeschlossen werden, indem ein Roboter, wenn er warten müsste, sofort alle seine Ressourcen wieder freigibt (seine Gegenstände wieder einsortiert). 2

Achtung: Livelock! No Preemption könnte verhindert werden, indem beispielsweise Roboter 1 immer Priorität besitzt und die Gegenstände von Roboter 2 übernehmen darf.

Noch trivialere wäre es, die Nebenläufigkeit zu verhindern und einen Roboter dauerhaft auszuschalten.

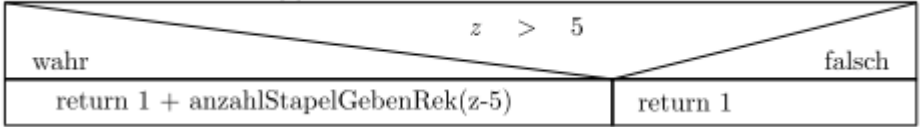
Vermutlich allerdings nicht im Sinne der Aufgabe.

Hinweis: Die Coffman-Bedingungen sind nicht im G8-Lehrplan enthalten, müssen hier also nicht explizit angesprochen werden. Im neuen Lehrplan werden sie explizit gefordert. Da die Aufgaben sicher auch als Übungsmaterial für das G9+ dienen, werden sie hier schon verwendet.

3a Angenommen Zuweisungen, Vergleiche, Additionen und Rückgabe sind in $\Theta(1)$ berechenbar. In diesem Fall wird die Wiederholung $\lceil n/5 \rceil$ -Mal durchlaufen. Dies entspricht einer linearen Laufzeit, also $\Theta(n)$. Da in jedem Durchlauf tatsächlich Berechnungen stattfinden, die in der Laufzeit $\Theta(1)$ berechnet werden, liegt die Laufzeit des Algorithmus in $\Theta(n)$. 2

Hinweis: Die Θ -Notation ist erst im Lehrplan des G9+ gefordert und wird deshalb hier schon einmal verwendet. Im Sinne des alten Lehrplans müsste man von konstantem Laufzeitverhalten bei $\Theta(1)$ und von linearem Laufzeitverhalten bei $\Theta(n)$ sprechen.

3b  3



3c Beweis durch Induktion (Es wird gezeigt, dass für die Zahlen 1-4 das Ergebnis gleich sich bei einer Erhöhung um 5 die Laufzeit in beiden Algorithmen gleich ändert.): 3

Für die Werte 1,2,3 und 4 liefert $\text{anzahlStapelGeben}(z)$ den Wert 1, $\text{anzahlStapelGeben2}(z)$ liefert $\lceil n/5 \rceil + 1$, also ebenfalls den Wert 1. Achtung: Für $n=0$ liefern die Algorithmen verschiedene Werte!

Für den Wert 5 liefert $\text{anzahlStapelGeben}(z)$ den Wert 1 (Die Wiederholung wird nicht durchlaufen). $\text{anzahlStapelGeben2}(z)$ liefert $z/5 = 1$, also ebenfalls den Wert 1.

Induktionsschritt $z \rightarrow z + 5$:

Für $\text{anzahlStapelGeben}(z)$ wird die Wiederholung ein zusätzliches mal durchlaufen. Das Ergebnis erhöht sich um 1. Für $\text{anzahlStapelGeben2}(z)$ wird entweder $(z + 5)/5 = (z/5) + 1$ oder $(z + 5)/5 + 1 = (z/5) + 2$ zurückgegeben. In beiden Fällen erhöht sich ebenfalls der Rückgabewert um 1.

Die Algorithmen liefern also für alle positiven ganzzahligen Werte das gleiche Ergebnis.

Die Laufzeit von $\text{anzahlStapelGeben2}(z)$ ändert sich nicht mit der Eingabelänge und unter der Voraussetzung, dass Zuweisungen, Vergleiche, Additionen und Rückgabe in $\Theta(1)$ liegen, liegt auch die Laufzeit von $\text{anzahlStapelGeben2}(z)$ in $\Theta(1)$, ist also konstant.

Hinweis: Beweise durch vollständige Induktion gehen über die Anforderungen der Schulinformatik hinaus. Es genügt eine anschauliche Begründung: Ist z durch 5 teilbar, sind alle Stapel voll. Ansonsten wird ein zusätzlicher Stapel begonnen. Beide Methoden liefern also die Anzahl an Stapeln.

3d minResthöhe wird zu Beginn auf den dritten Parameter, also 200 gesetzt. Die Wiederholung wird zweimal durchlaufen, dabei nimmt minResthöhe die Werte 35 und anschließend 10 an. 2

3e Die drei Übergabeparameter liegen in den Speicherzellen 100, 101 und 102. Die Variable minResthöhe wird in Zelle 103 gespeichert. Die Zählvariable k wird in Zelle 104 gespeichert. Die Variable resthöhe wird in 105 gespeichert. Der Rückgabewert wird in Zelle 200 8

gespeichert.

Hinweis: Eine symbolische Adressierung der Speicherzellen ist möglich.

```
        LOAD 102
        STORE 103
        LOADI 0
        STORE 104
wiederhole : LOAD 102
            DIV 100
            SUB 104
            JLT endezaehle
            LOAD 104
            MUL 100
            SUB 102
            MULI -1
            MOD 101
            STORE 105
            SUB 103
            JGE endewenn
            LOAD 105
            STORE 103
endewenn :  LOADI 1
            ADD 104
            STORE 104
            JMP wiederhole
endezaehle : LOAD 102
            SUB 103
            STORE 200
            HOLD
```