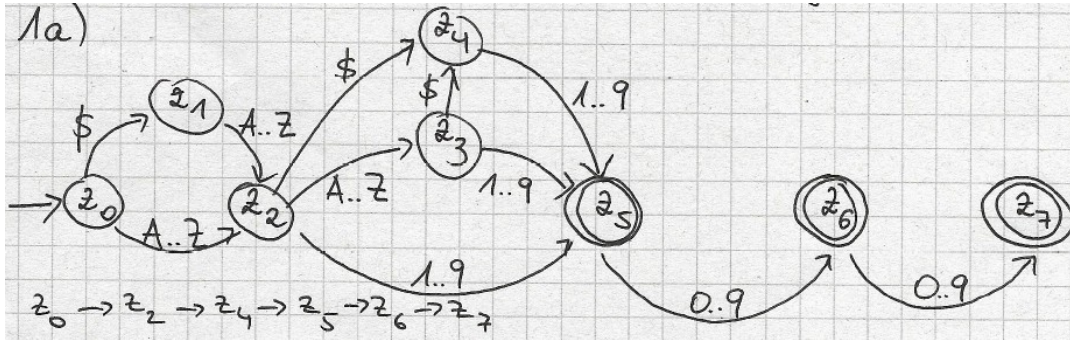


# Informatik Bayern 2021/III – Beispiellösung

Autor\*innen:  
 Anolick (1),  
 Kornhas (1),  
 Müller(ab 2),  
 Wiedemann  
 (1)

1a

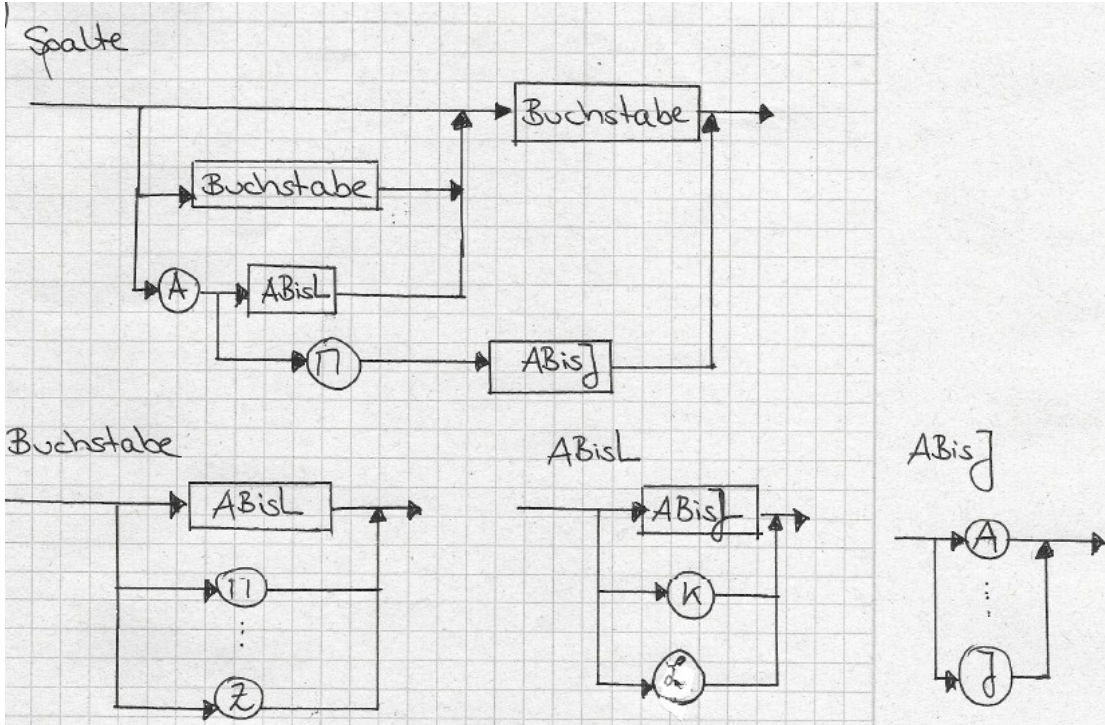


6

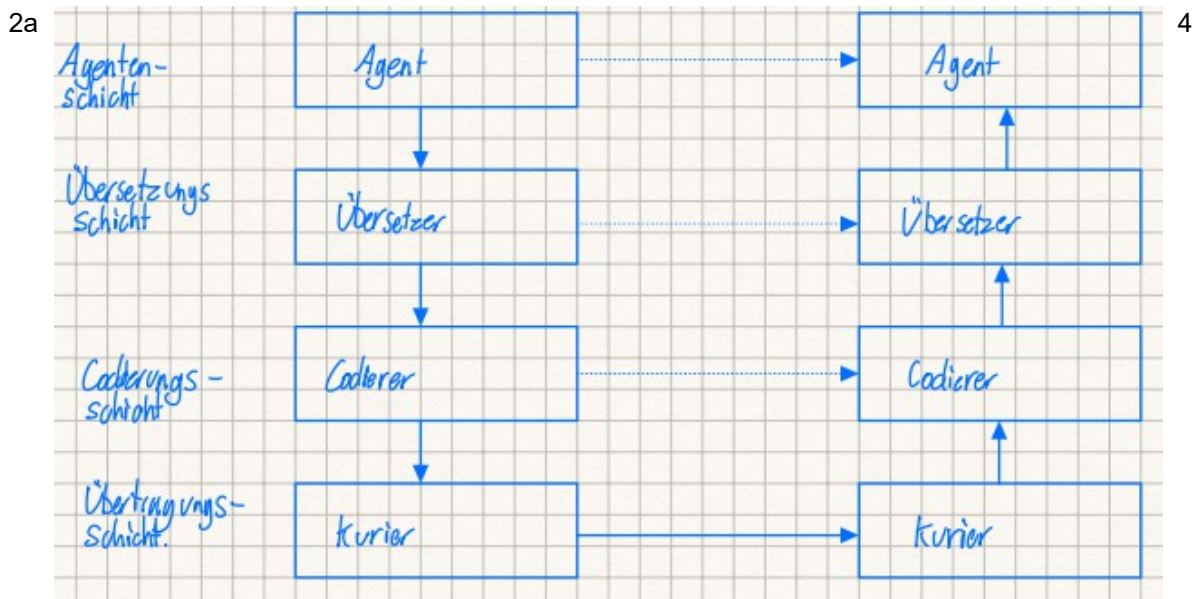
- b Zellbezug = ["\$"] Spalte ["\$"] Zeile;  
 Spalte = Buchstabe [Buchstabe];  
 Zeile = ZifferOhneNull [Ziffer][Ziffer];  
 Buchstabe = "A" | "B" | "C" | ... | "Z";  
 ZifferOhneNull = "1" | "2" | "3" | ... | "9";  
 Ziffer = "0" | ZifferOhneNull;

4

c



5



- b Protokolle beschreiben den Ablauf von Kommunikation und Datenaustausch auf einer Schicht. 4

Wenn die Kuriere ein anderes Übergabeverfahren vereinbaren, beeinflusst dies nicht die Arbeit der Codierer, Übersetzer und Agenten.

- c  $100^x = 2 \cdot 3600 \cdot 500\,000 \cdot 1000 = 7,3 \cdot 10^{14}$  5

$$\frac{0,1}{100} * 100^x \geq 24 * 3600 * 5.000.000 \Rightarrow x \geq 7,3$$

Die Mindestlänge der Zeichenkette beträgt somit **8 Zeichen**.

- 3a Eine Verklemmung ist eine zyklische Wartesituation, in der Prozesse Ressourcen reserviert haben und auf die Freigabe anderer Ressourcen warten, um in ihrer Arbeit fortzufahren. . 4

Im Beispiel entsprechen die Ressourcen den Bereichen der Fahrbahn, auf denen die Fahrzeuge A1, A2, B1 und B2 stehen. Die Fahrzeuge entsprechen den Prozessen, die auf die Freigabe eines Fahrbahnabschnitts warten, der von einem anderen Auto exklusiv belegt wurde..

- b Eine mögliche Regel wäre, dass Fahrzeuge nur in den Kreuzungsbereich einfahren dürfen, wenn der Bereich davor frei ist. Eine analoge Strategie könnte sein, dass ein Prozess, der auf die Freigabe weiterer Betriebsmittel wartet, seine reservierten Betriebsmittel wieder freigibt. 5

4a

3

		Akku	100	101	102
(Anfangszustand)				7	0
Load i	101	101			
Store	100		101		
Load	(100)	7			
Load	100	101			
add i	1	102			
Store	100		102		
Load	(100)	0			

4b

```

start:      loadi 0
            store 98
            loadi 101
            store 100
            load (100)
            jeq end
            sub 99
            jeq found
            load 100
            addi 1
            store 100
            jmp start
found:     loadi 1
            store 98
end:      hold

```

40