

# Informatik Abitur Bayern 2014 / III - Beispiellösung

Autor:  
Reinold

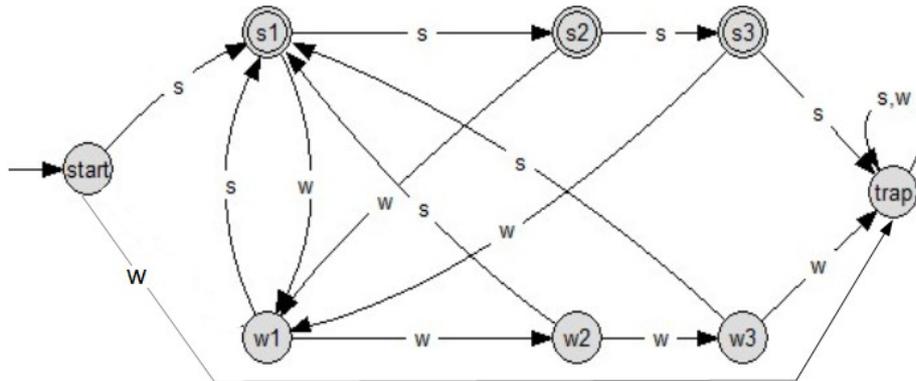
1a  $\langle S \rangle \rightarrow \langle \text{Paar} \rangle \quad \langle \text{Paar} \rangle \quad \langle \text{Paar} \rangle \quad \langle \text{Schwarz} \rangle$   
 $\rightarrow \langle \text{Schwarz} \rangle \langle \text{Weiß} \rangle \langle \text{Schwarz} \rangle \langle \text{Weiß} \rangle \langle \text{Schwarz} \rangle \langle \text{Weiß} \rangle \langle \text{Schwarz} \rangle$   
 $\rightarrow \text{ss} \quad \text{w} \quad \text{sss} \quad \text{ww} \quad \text{s} \quad \text{www} \quad \text{s}$

2

1b Jeder Strichcode besteht aus maximal 3 Strichpaaren und einem schwarzen Strich, also aus sieben Strichen der maximalen Breite 3. Die maximale Breite beträgt 21. Der kürzestmögliche Strichcode hat die Länge 3 (sws). Deshalb sind Strichcodes aus einem einzigen schwarzen Streifen nicht möglich.  
 Anpassung von Regel 1:  
 $\langle S \rangle \rightarrow \{ \langle \text{Paar} \rangle \} \langle \text{Schwarz} \rangle$ .

4

1c

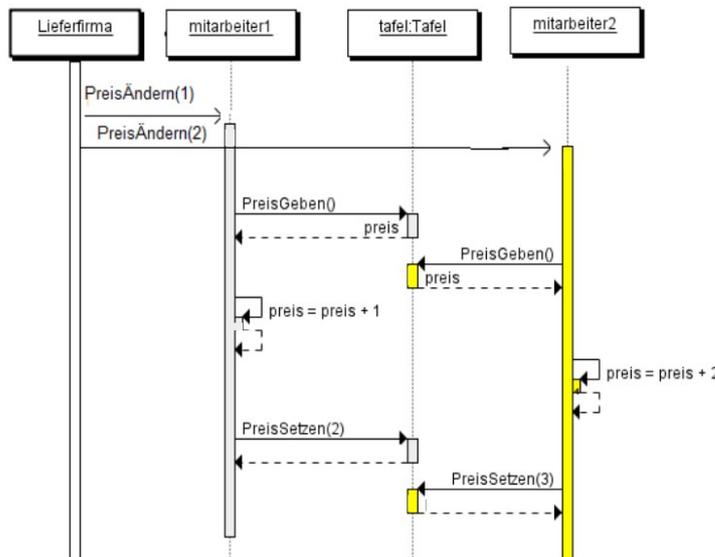


6

2a Das Betriebsmittel Preistafel kann von mehreren Prozessen parallel gelesen und geschrieben werden. Da der Zugriff koordiniert werden muss sind die Prozesse nebenläufig. Sonst könnte z. B. der zweite den veralteten Preis lesen, ehe der erste eine neue Preistafel ausgehängt hat.

3

2b



5

2c Durch Monitore lassen sich Programmabschnitte definieren, die immer nur von einem Prozess gleichzeitig ausgeführt werden dürfen. Einen solchen Monitor benötigt die Methode PreisÄndern.

4

3a  $t = 0,3 \cdot 10^{20-6} / 10000 \text{ s} = 0,3 \cdot 10^{10} \text{ s} = 95 \text{ a}$

3

3b Momentan ist der RSA-Algorithmus sicher für geeignet hohe Zahlen. Bei entsprechend wachsender Rechengeschwindigkeit ( $\rightarrow$  Mooresches Gesetz) müsste dies durch Erhöhung der Zahlen ausgeglichen werden.

3

4a  $w$  ist die größte Ganzzahl unterhalb der Wurzel von  $r$ .  $r$  wird nicht verändert.  
 $w = 3$ ;  $r = 11$ .

2

4b w in 102, r in 101

8

```
    loadi 1
    store 102
wdh:
    load 102
    mul 102
    sub 101
    jmpnn endeWdh
    load 102
    addi 1
    store 102
    jmp wdh
endeWdh:
    load 102
    mul 102
    sub 101
    sub 102
    jmpnp ende
    load 102
    subi 1
    store 102
ende: hold
```