

Informatik Abitur Bayern 2012 / IV - Lösung

Autor:
Wankert

1a i) lkk: Zustandsfolge $z_0 \rightarrow z_1 \rightarrow z_2 \rightarrow z_3 \rightarrow z_4$

z_4 ist kein Endzustand, also ist das Wort nicht zulässig.

ii) lklkkk: Zustandsfolge $z_0 \rightarrow z_1 \rightarrow z_2 \rightarrow z_3 \rightarrow z_1 \rightarrow z_2 \rightarrow z_3 \rightarrow z_4 \rightarrow z_5$

z_5 ist ein Endzustand, also ist das Wort zulässig.

3

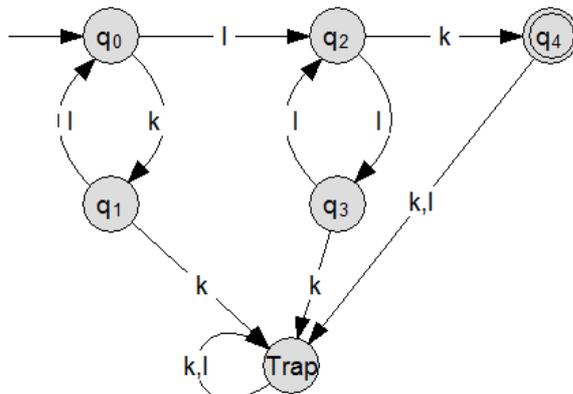
1b

```

public class AUTOMAT{
    private int zustand;
    private void zeichenVerarbeiten(char c) {
        switch (zustand) {
            ...
            case 3:
                switch (c) {
                    case 'l': zustand = 1; break;
                    case 'k': zustand = 4; break;
                    default: zustand = -1; //Fangzustand
                }
                break;
            ...
            case 5:
                zustand = -1; //Fangzustand
                break;
        }
    }
    boolean codeTesten(String wort) {
        zustand = 0;
        for (int i=0; i<wort.length(); i++){
            zeichenVerarbeiten(wort.charAt(i));
        }
        return (zustand == 5);
    }
}
    
```

9

1c



Angabe des Fangzustandes optional.

5

2 Vorteile:

- Je nachdem ob gerade mehr Speicherplatz für Daten oder für Programme benötigt wird, kann der Arbeitsspeicher flexibel aufgeteilt werden.
- Generieren von Programmcode zur Laufzeit.

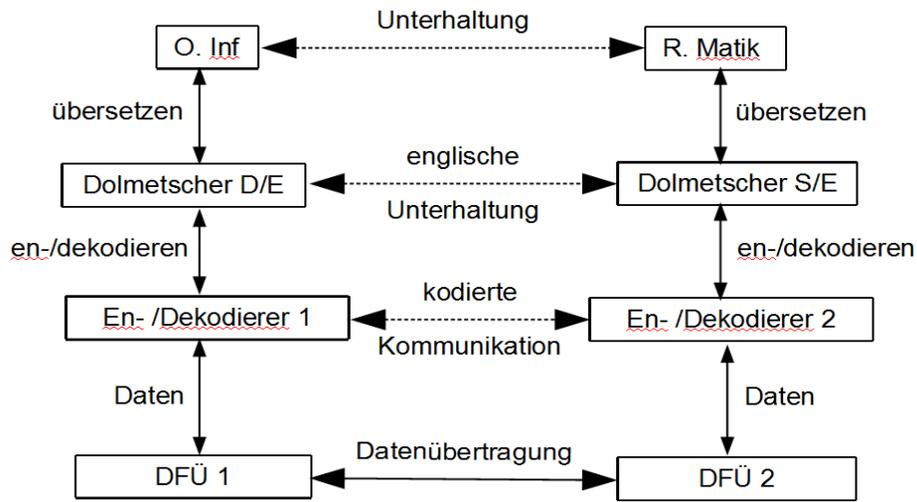
Nachteil:

- Von-Neumann'scher Flaschenhals: Auf den gemeinsamen Speicher muss über einen gemeinsamen Bus zugegriffen werden, dies kann zu Engpässen führen.
- Programm könnte sich zerstören, indem es zur Laufzeit seinen eigenen Code überschreibt.

4

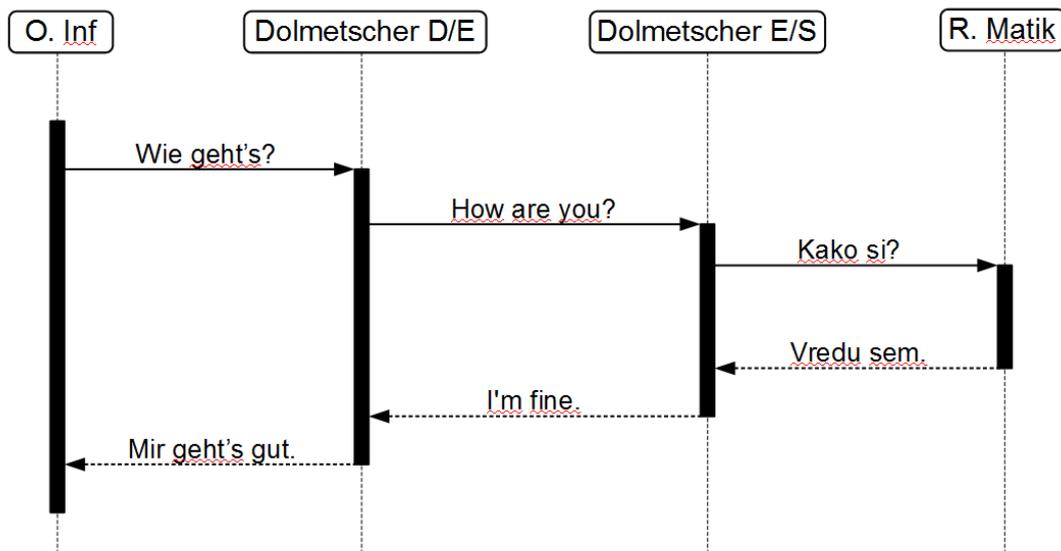
3a

6



3b

4



4a

3

a	14	6	6	4	2
b	8	8	2	2	2

4b Zu Beginn: a in 100, b in 101
Am Ende: a in 100, b in 101

6

```

start:load 100
      sub 101
      je ende
      jg sonst
      load 101
      sub 100
      store 101
      jmp start
sonst:store 100
      jmp start
ende: end
    
```

40