

- 1a  $\Sigma = \{ \_ ; 0 ; 1 ; \dots ; 9 ; \_ ; A ; B ; \dots ; Z ; \_ ; \emptyset \}$   
 $V = \{ \text{Gesamt} ; \text{Ort} ; Z ; B \}$   
 $S = \text{Gesamt}$

6

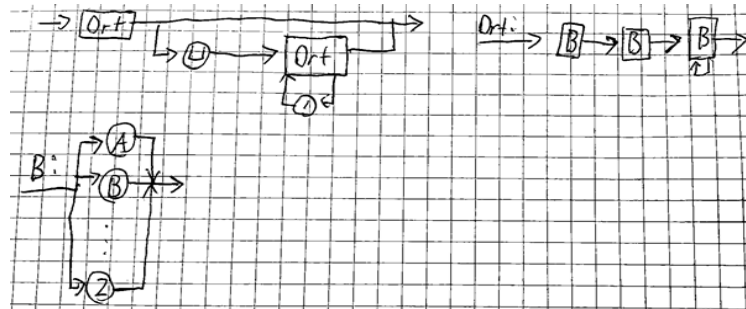
Regeln:

Gesamt = ZZ' : ZZ' BB[B]ZZZ' Ort' [Ort{ ' Ort' } Z[Z] [ 'V' | 'Ø' ] ;  
 Ort = BBB{B} ;  
 Z = '0' | '1' | ... | '9' ;  
 B = 'A' | 'B' | ... | 'Z' ;

- 1b Die Syntax beschreibt das pure Einhalten der Regeln, so ist zum Beispiel:  
 29:63 \_ICE1234 \_ULM \_5 syntaktisch richtig, da es aus den Produktionsregeln ableitbar ist.  
 Die Semantik bezeichnet darüber hinaus aber den Sinn. Die in unserem Beispiel gewählte  
 Uhrzeit existiert nicht, somit ist die Zeichenkette sinnlos, hat also keine Semantik.

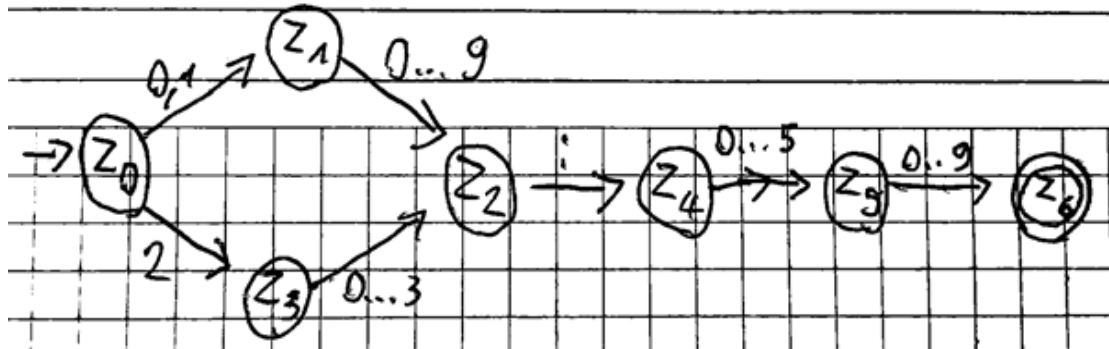
2

1c



4

1d



4

- |                      |             |             |   |
|----------------------|-------------|-------------|---|
| 2a Nutzerschicht     | Nutzer      | Nutzer      | 4 |
| Internetschicht      | Accesspoint | Accesspoint |   |
| Datentransferschicht | Mobilfunk   | Mobilfunk   |   |

- 2b Die Nutzer brauchen keine Kenntnisse über die Art und Weise der Datenübertragung, da sie die Nutzersicht nicht verlassen müssen.  
 Die einzelnen Ebenen sind disjunkt, veränderte Bedingungen müssen daher nur in der betroffenen Schicht beachtet werden.

2

```

3a start: load personen
      add betreten
      sub verlassen
      store personen
      loadl 0
      store betreten
      store verlassen
      loadl 200
      sub personen
      jle dann
      loadl 1
      jmp weiter
dann: loadl 0
weiter: store ampeln
      jmp start

```

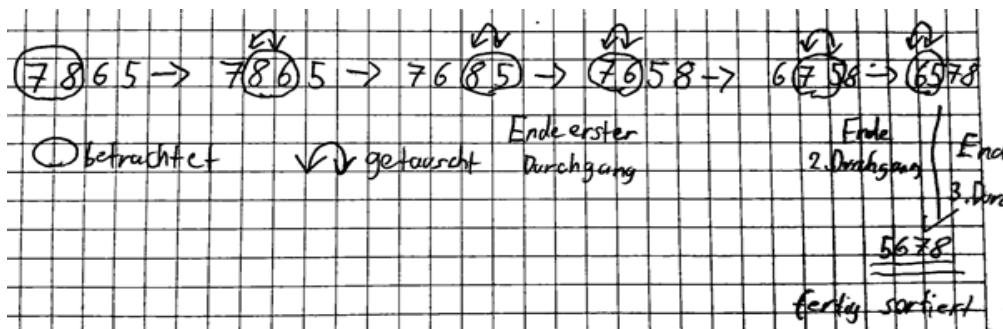
6

3b Da die Ampel nicht unmittelbar auf Personen reagiert, die die Lounge betreten, ist es möglich, dass zwei Personen direkt nacheinander hinein gehen, obwohl bereits 199 Personen in dieser sind. Wenn das Programm gerade die Befehle ab Zeile 7 abarbeitet ist dies möglich.

2

4a

6



Der Algorithmus vergleicht immer zwei Zahlen miteinander, beginnend am Listenanfang, dann jeweils eine Zahl später beginnend. Ist die erste Zahl größer, so werden die beiden Zahlen getauscht. Da bei jedem Durchgang die größte Zahl an das Ende des noch zu sortierenden Listenabschnitts rückt, muss mit jedem Durchgang eine Zahl weniger am Ende der Liste berücksichtigt werden. Zusätzlich wird geprüft, ob bei einem Durchgang ein Tausch stattfindet, ist dies nicht der Fall, so ist die Liste bereits fertig sortiert und der Algorithmus kann vorzeitig beendet werden. (Hinweis: Es handelt sich um den sog. Bubblesort-Algorithmus)

4b Günstigster Fall: vorsortierte Liste

4

Die äußere Wiederholung wird nur einmal ausgeführt, die innere dabei n-1 mal, bevor der Algorithmus endet, da kein Tausch stattfand. => Lineare Laufzeit mit n-1 Vergleichen

Ungünstigster Fall: absteigend sortierte Liste

Bei jedem Durchgang wandert nur eine Zahl vom Anfang der Liste an das Ende des noch zu sortierenden Teils der Liste. Daher wird die äußere Wiederholung die maximalen n-1 mal ausgeführt. Die innere Wiederholung wird dabei mit n-1 Wiederholungen beginnend, pro Durchlauf einmal weniger durchlaufen.

Es gibt daher  $(n-1) + (n-2) + \dots + [n-(n-1)]$  Vergleiche. Dies entspricht  $1 + 2 + 3 + \dots + (n-1)$

Vergleichen. Laut der gaußschen Summenformel sind dies  $\frac{(n-1) \cdot n}{2}$  also  $\frac{n^2 - n}{2}$

Vergleiche.

Das Laufzeitverhalten ist somit quadratisch.

40