

1a

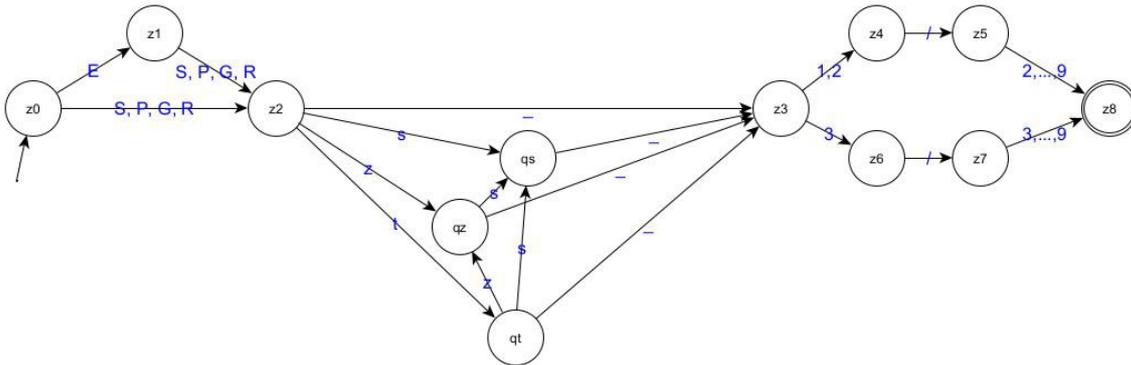
4

EBNF

bezeichnung = ['E'] zugtyp zusatzinfos '_' achsen ;
 zugtyp = 'S' | 'P' | 'G' | 'R' ;
 zusatzinfos = ['t'] ['z'] ['s'] ;
 achsen = ('3' | ' zahlen) | (('1' | '2') | ' ('2' | zahlen)) ;
 zahlen = '3' | '4' | '5' | '6' | '7' | '8' | '9' ;

b Erkennender Automat:

2



2a Möglichkeit 1:

2

Anzahl der Verbindungen:

$$99+98+\dots+2+1=(99+0)+(98+1)+\dots+(50+49)=50 \cdot 99 = 4950$$

Gesamtlänge Kabel:

$$4950 \cdot 150 \text{ m} = 742\,500 \text{ m}$$

Möglichkeit 2:

$$\binom{100}{2} \cdot 150 \text{ m} = 742500 \text{ m}$$

b Durch die gefährliche Umgebung können leicht Kabel und Sensoren beschädigt werden. Durch die Topologie wird sichergestellt, dass die Überwachung selbst bei Teilausfällen fortgeführt werden kann.

2

3 Ein Deadlock ergibt sich, wenn beide Durchreiche belegt sind und alle Roboter darauf warten etwas in die Durchreiche stellen zu können.

4

Kann durch verschiedenes verhindert werden:

1. Mglk: Küchenroboter priorisieren abgeräumtes Geschirr
2. Mglk: Servierroboter priorisieren zubereitete Speisen
3. Mglk: Einer der Küchenroboter spezialisiert sich auf abgeräumtes Geschirr.
4. Mglk: Einer der Servierroboter spezialisiert sich auf zubereitete Speisen.

4a

4

Durchläufe der Wiederholung	ug	og	erg
0	1	11	6
1	1	6	3
2	3	6	4
3 (=Ablauf)	3	4	3

b

7

```

loadi 1    //ug=1
store ug
loadi x    //og=x
store og
loadi ug   //erg=(ug+og)/2
add og
divi 2
store erg
wdh:      load erg
          sub ug    //ug<erg
          jle return
          load erg  //erg*erg<=x
          mul erg
          sub x
          jle dann
          load erg  //falsch
          store og
          jmp wdh
dann:     load erg  //wahr
          store ug
          jmp Wdh
return:   load erg
          hold

```

c

4

Direkte Übertragung	Vereinfachung
<pre> int a2(int x){ erg = 0; while(true){ tmp = erg + 1; if(tmp * tmp > x){ break; } erg++; } return erg; } </pre>	<pre> int a2(int x){ erg = 0; for(; (erg + 1) * (erg + 1) <= x; erg++){ } return erg; } </pre>

d Die Algorithmen liefern die auf eine ganze Zahl abgerundete Wurzel von x

2

e Lösung:

4

a2: doppelte Anzahl der Wiederholungen

a1: zwei Wiederholungen mehr

Ich würde a1 bei großen Werten von x vorziehen, da eine Verdopplung der Wiederholungen in diesem Fall den Laufzeit sehr viel mehr verlängert als zwei Wiederholungen mehr.

Erklärung (nicht verlangt):

Ist erg das Ergebnis des Algorithmus a2, so entspricht die Anzahl der Wiederholungen genau dem Wert von erg.

Wegen $\sqrt{4 \cdot x} = 2 \cdot \sqrt{x}$ verdoppelt sich so die Anzahl der Wiederholungen.

In Algorithmus a1 wird der Suchraum, in dem nach der gesuchten Zahl gesucht wird, mit jeder Wiederholung halbiert. Zu Beginn hat der Suchraum die Größe x. Wegen $\frac{4 \cdot x}{2 \cdot 2} = x$ erhöht sich die Anzahl der Wiederholungen um zwei.

Die Aufgabenstellung gibt nicht an, aufgrund welcher Kriterien man sich für einen Algorithmus entscheiden soll. Unter den Kriterien Lesbarkeit, Wartbarkeit und Programmieraufwand schneidet a2 natürlich deutlich besser ab als a1.

Weiterhin ist nicht spezifiziert, ob eine Beurteilung verlangt ist, für den Fall, dass x groß ist und dann noch einmal vervierfacht wird oder für den Fall, dass x einfach nur groß ist.

