

# Informatik Abitur Bayern 2017 / III - Lösung

Autoren:  
Marek (1)/  
Schulz(2)/  
Janus(3, 4,  
5)

1a Gesucht ist die Grammatik G der Sprache L über dem Alphabet  $\Sigma$ .  
 $G = (\{M, L, B, Z, Z_1, Z_2\}, \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, x\}, P, \text{Startsymbol } M)$

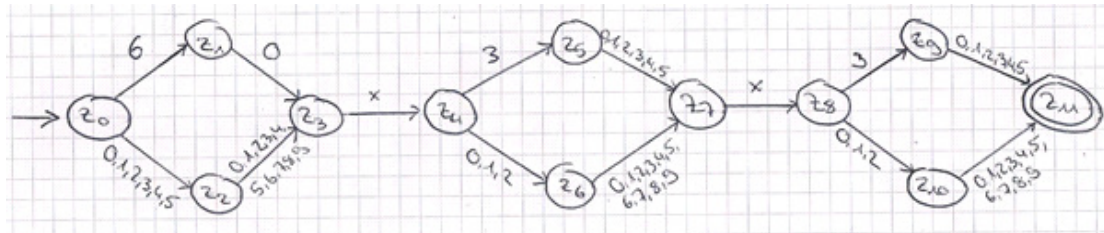
5

Dabei lauten die Produktionsregeln P wie folgt:

- $M = L \text{ "x" } B \text{ "x" } B;$
- $L = Z_2 Z \mid \text{"6" "0"};$
- $B = \text{"3" } Z_2 \mid Z_1 Z;$
- $Z = Z_2 \mid \text{"6" } \mid \text{"7" } \mid \text{"8" } \mid \text{"9"};$
- $Z_1 = \text{"0" } \mid \text{"1" } \mid \text{"2"};$
- $Z_2 = Z_1 \mid \text{"3" } \mid \text{"4" } \mid \text{"5"};$

Idee: Man wählt drei verschiedene Nichtterminale, um die unterschiedlichen Zifferntypen darzustellen. Dabei kann Z jede beliebige Ziffer zwischen 0 und 9 annehmen, während  $Z_1$  nur die Ziffern 0 - 2 und  $Z_2$  die Ziffern 0 - 5 annehmen können. Somit kann in der Grammatik gesichert werden, dass die maximale Größe eines Paketes nicht überschritten wird.

1b



4

Idee: Nach jeder neueingelesenen Ziffer wird in einen neuen Zustand gewechselt. Die oberen Zustände ( $z_1, z_5, z_9$ ) werden jeweils bei den jeweiligen Maximalgrößen erreicht, sonst gelangt man in die Zustände  $z_2, z_6$  und  $z_{10}$ . Wird die Maximalgröße eines Pakets überschritten, gelangt man in keinen weiteren Zustand und somit auch letztendlich nicht in den akzeptierenden Zustand  $z_{11}$ .

1c  $z_0 \rightarrow z_2 \rightarrow z_3 \rightarrow z_4 \rightarrow z_6 \rightarrow z_7 \rightarrow z_8 \rightarrow z_{10} \rightarrow z_{11}$

2

Da dies ein Endzustand ist, gehört die Größenangabe zur Sprache

1d Das Wort 47x00x11 ist ein Wort der Sprache L, da es aus der Grammatik ableitbar ist und somit der Syntax dieser Sprache genügt. Unter Semantik versteht man die Bedeutung von Zeichen, d.h. in diesem Fall die Bedeutung des Wortes, nämlich die Paketmaße 47 cm Länge, 0 cm Breite und 15 cm Höhe. Das Wort ist also syntaktisch korrekt, aber nicht semantisch korrekt, da es kein Paket mit Breite 0 geben kann.

2

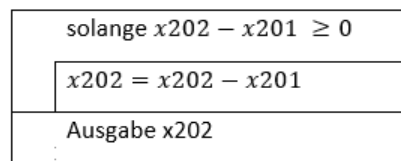
2a

	Akkumulator	201	202
Vorbelegung		6	17
load 202	17	6	17
sub 201	11	6	17
store 202	11	6	11
sub 201	5	6	11
store 202	5	6	5
sub 201	-1	6	5
load 202	5	6	5

3

2b Das Programm berechnet allgemein den Rest der ganzzahligen Division der in Speicherzelle 202 abgelegten Zahl durch die Zahl aus Speicherzelle 201. Im angegebenen Beispiel beträgt die Menge der möglichen Prüzziffern {0, 1, 2, 3, 4, 5}. 2

2c Als Struktogramm sieht die gegebene Methode wie folgt aus: 4



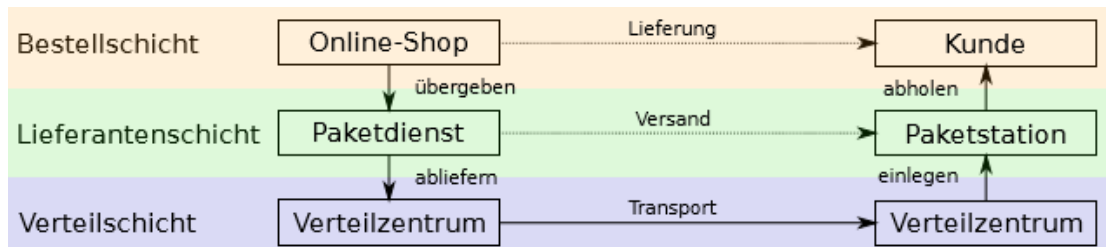
2d Ergänzt wird das Programm vor Zeile 10 durch folgende Änderungen: 3

```

8: load 201
9: jlt 17
...
17: dload -1
9: end
  
```

Die (potenziell) negative Zahl aus 201 wird in den Akkumulator geladen, geprüft und die Programmausführung in Zeile 17 fortgesetzt, wenn ein negativer Wert enthalten ist.

3a 5



**Ergänzende Hinweise:**  
 Auf der Bestellschicht agieren Online-Shop und Kunde direkt miteinander. Auf abstrakter Ebene gelangt die Lieferung also direkt vom Online-Shop zum Kunden (Pfeil "Lieferung"). Tatsächlich muss die Lieferung jedoch zuerst in ein für die Lieferantenschicht passendes Format verpackt werden (hier z.B. ein Postpaket mit Versandetikett) und an den Paketdienst übergeben werden. Betrachtet man den Systemschnitt auf dieser Schicht, so erfolgt scheinbar eine Beförderung der Lieferung direkt vom annehmenden Paketdienst zur Empfänger-Paketstation (Pfeil "Versand"). Der Paketdienst führt aber nicht die komplette Zustellung der Lieferung selbst durch, sondern erzeugt abermals ein für die unterhalb liegende Verteilschicht passendes Format (z.B. ein Container mit Paketen), und übergibt diese an das Verteilsystem. Als unterste Schicht wird hier nun der tatsächliche Transport durchgeführt, bevor der Schichtenstapel in umgekehrter Reihenfolge durchlaufen wird (z. B. Entladen des Containers auf der Lieferantenschicht, Entpacken des Pakets durch den Kunden auf der Bestellschicht).

3b z. B. eines der folgenden Argumente 2

- Ein wichtiger Vorteil der Schichtenarchitektur ist die Austauschbarkeit einzelner Schichten, so kann innerhalb der Lieferkette beispielsweise der Verteilnetzbetreiber ausgetauscht werden, ohne, dass dies den Online-Shop oder Kunden tangiert.
- Ebenso können auch andere Online-Shops die Lieferinfrastruktur nutzen, sofern sie geeignete "Schnittstellen" bereitstellen.
- Davon abgesehen liegt ein Vorteil in der Kapselung von Abläufen (vgl. Objektorientierung). Ein potentiell hochkomplexes System kann so beherrschbar gestaltet werden, indem Funktionalitäten klar den verschiedenen Schichten zugeordnet werden und durch eindeutige Schnittstellendefinitionen mit den Nachbarschichten verbunden werden. Im gegebenen Szenario muss sich der Betreiber des Online-Shops beispielsweise nicht um die Routenplanung der Lieferfahrzeuge kümmern, während umgekehrt der Paketdienst nichts über die aktuellen Entwicklungen am Buchmarkt wissen muss.

- 4 Damit es zu der beschriebenen Fehlfunktion kommen kann, muss zunächst davon ausgegangen werden, dass der Server die eingehenden Reservierungsanfragen (pseudo-)parallel abarbeitet. Ist dies gegeben, kann es passieren, dass, nachdem die Reservierung für das letzte noch freie Ablagefach angenommen wurde (Zeile 2 im Algorithmus), ein weiterer Kunde ein Ablagefach erfolgreich reserviert, bevor durch die Verringerung der Zahl der freien Ablagefächer (Zeile 3) weitere Reservierungen verhindert werden.

Ein solcher Ablauf ist beispielsweise:

- Kunde A - Zeile 1: Es ist noch genau ein Ablagefach frei.
- Kunde A - Zeile 2: Die Reservierung wird angenommen.
- Kunde B - Zeile 1: Es ist noch genau ein Ablagefach frei.
- Kunde B - Zeile 2: Die Reservierung wird angenommen.
- Kunde B - Zeile 3: Die Zahl der freien Ablagefächer wird verringert auf 0.
- Kunde B - Zeile 4: Kunde B bekommt das letzte freie Fach zugewiesen.
- Kunde A - Zeile 3: Die Zahl der freien Ablagefächer wird auf -1 verringert
- Kunde A - Zeile 4: Es wird ein Ablagefach zugewiesen, obwohl keines mehr verfügbar ist.

Schützt man den Algorithmus durch einen Monitor, so kann er nur von einem Prozess vollständig ausgeführt werden, ehe der nächste Prozess mit der Ausführung beginnt.

- 5 Bei einer Passwortlänge von fünf Ziffern gibt es 4

$$10 \times 10 \times 10 \times 10 \times 10 = 10^5 = 100000$$

verschiedene Passwörter, welche (im Worst-Case) alle durchprobiert werden müssen, bevor das letzte ausprobierte Passwort zum Erfolg führt. Bei 0,2 ms pro individuellem Anmeldeversuch ergibt sich ein Gesamtaufwand von

$$100000 \times 0,0002 \text{ Sekunden} = 20 \text{ Sekunden.}$$

Im Durchschnitt (Average Case) ist jedoch davon auszugehen, dass das gesuchte Passwort bereits nach der Hälfte der Zeit erraten wird.

Die Sicherheit der Passwörter selbst könnte gesteigert werden, indem

- a) das Zeichenalphabet vergrößert wird (z.B. Ziffern  $\cup$  Buchstaben  $\cup$  Sonderzeichen)
- b) die Passwortlänge vergrößert wird. (*Hinweis: Variante (b) ist dabei effektiver, da die Anzahl möglicher Passwörter fester Länge mit jedem zusätzlichen Zeichen im Alphabet schwächer wächst, als dies bei Verlängerung des Passwortes um ein Zeichen (bei unverändertem Alphabet) der Fall ist.*)

Weiterhin kann auch bei unveränderter Passwortsammensetzung durch technische Maßnahmen die Sicherheit erhöht werden. Wird z.B. die Dauer eines Anmeldeversuches künstlich verlängert oder die Anzahl der maximal erlaubten Fehlversuche beschränkt (vgl. Pineingabe am Geldautomaten), kann ein erfolgreicher Brute-Force-Angriff ebenfalls effektiv verhindert oder zumindest sehr unwahrscheinlich gemacht werden.