

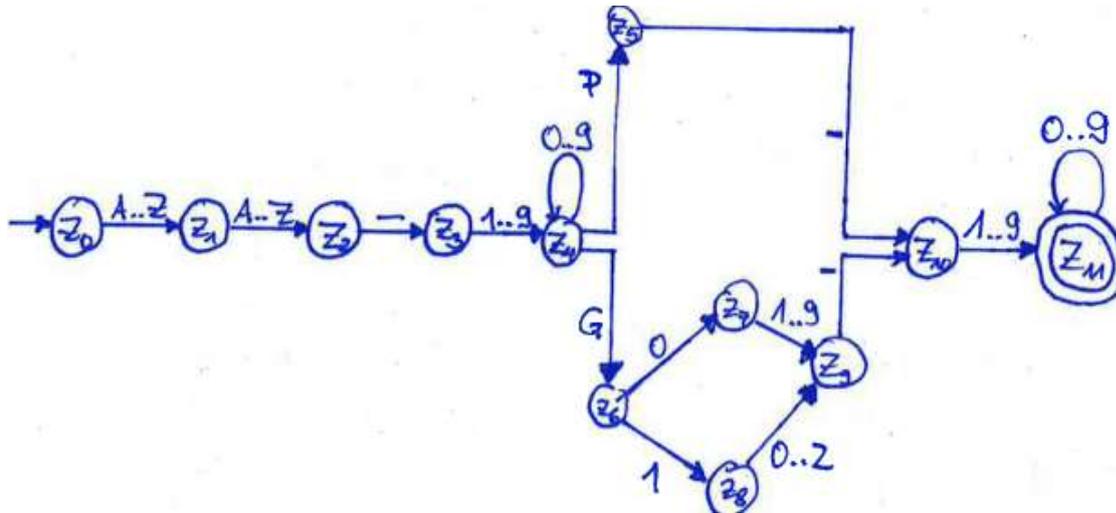
Informatik Abitur Bayern 2018 / IV - Lösung

Autor:
Reinold

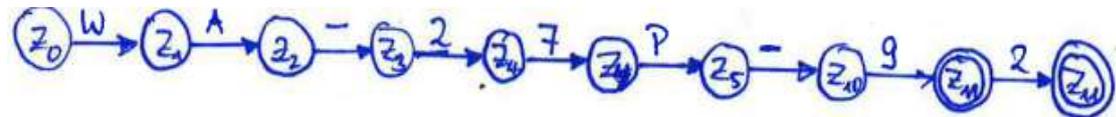
1a Eine vollständige Angabe der Grammatik ist hier nicht gefordert. Die Angabe der Regeln genügt! 5

- R1: Rechnungskennzahl = Initialen "-" Ziffernfolge Versicherung "-" Ziffernfolge.
- R2: Initialen = Buchstabe Buchstabe.
- R3: Buchstabe = "A"|"B"|...|"Z".
- R4: Versicherung = "P"|"G"("0" ZifferNichtNull|"1"("0"|"1"|"2")).
- R5: Ziffernfolge = ZifferNichtNull {Ziffer}.
- R6: ZifferNichtNull = "1"|"2"|...|"9".
- R7: Ziffer = "0"|ZifferNichtNull.

1b 6



1c 2



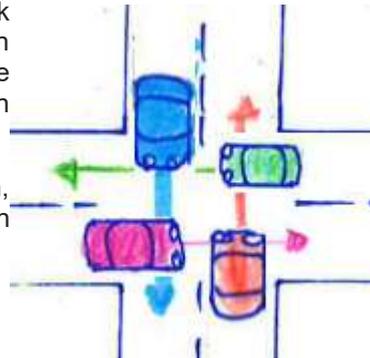
1d class KENNZAHLTESTER 6

```

class KENNZAHLTESTER
{
    private int zustand;
    public boolean kennzahlTesten(String kennzahl){
        zustand = 0;
        for(int i = 0; i<kennzahl.laenge(); i = i+1){
            zustandWechseln(kennzahl.zeichenAn(i));
        }
        if(zustand==11) return true else return false;
        //kurz: return zustand==11
    }

    private void zustandWechseln(char c){
        switch (zustand){
            ...
            case 6:
                switch(c){
                    case '0': zustand = 7; break;
                    case '1': zustand = 8; break;
                }
                break;
            ...
        }
    }
}
    
```

2a Unter einer **Verklemmung** versteht man in der Informatik einen Zustand, bei dem eine zyklische Wartesituation zwischen mehreren Prozessen auftritt, wobei jeder beteiligte Prozess auf die Freigabe von Betriebsmitteln wartet, die ein anderer beteiligter Prozess bereits exklusiv belegt hat.



4

Wenn alle vier Fahrzeuge in die Kreuzung einfahren, blockieren sie dadurch einen Kreuzungsbereich, den ein anderes Fahrzeug zum Verlassen der Kreuzung benötigt.

2b Eine Ampel verhindert das gleichzeitige Einfahren der vier Fahrzeuge. Zwei Fahrzeuge können die Ampel ungehindert passieren und die gemeinsam genutzte Ressource für die beiden anderen Fahrzeuge in deren Grünphase freimachen.

3a Fetch: Der Opcode zum Befehl sub, dessen Adresse im Befehlszähler steht, wird mit dem Argument 102 in das Befehlsregister geholt. Der Befehlszähler wird inkrementiert.

Decode: Der Befehl sub wird dekodiert (Schaltung der ALU)

Execute: Ausführen des Befehls (Akkumulator – Inhalt vor Speicherzelle 102); Schreiben des Ergebnisses in den Akkumulator

3b

5

| Befehlszähler | Akkumulator | 101 | 102 | 103 | 104 | nach Befehl |
|---------------|-------------|-----|-----|-----|-----|-------------|
| 1 | ? | 15 | 23 | 7 | ? | |
| 2 | 15 | 15 | 23 | 7 | ? | load 101 |
| 3 | -8 | 15 | 23 | 7 | ? | sub 102 |
| 10 | -8 | 15 | 23 | 7 | ? | jlt 10 |
| 11 | 23 | 15 | 23 | 7 | ? | load 102 |
| 12 | 16 | 15 | 23 | 7 | ? | sub 103 |
| 13 | 16 | 15 | 23 | 7 | ? | jlt 16 |
| 14 | 23 | 15 | 23 | 7 | ? | load 102 |
| 15 | 23 | 15 | 23 | 7 | 23 | store 104 |
| 18 | 23 | 15 | 23 | 7 | 23 | jmp 18 |
| 18 | 23 | 15 | 23 | 7 | 23 | end |

Eine Angabe der Zustandstabelle ist nicht gefordert. Die Werte des Befehlszählers können in Spalte 1 abgelesen werden.

In Speicherzelle 104 wird das Maximum der drei Zeilen abgelegt.

3c Der Algorithmus berechnet die Summe der Quadrate der Zahlen von 1 bis 10.

6

```
summe:      WORD 0 // Symbolische Adressierung der Speicherzellen
i:          WORD 100
eins:       WORD 1
// sollte im Kurs eine symbolische Adressierung der Speicherzellen nicht
// eingeführt worden sein, so können die Zahlen auch in numerisch
// adressierten Zellen abgelegt werden.

wdh:        dload i
            jle ende
            mul i
            add summe
            store summe
            dload i
            sub eins
            store i
            jmp wdh
ende:       end
```

Das Ergebnis steht in der Speicherzelle summe.