



Universität Karlsruhe (TH)

Institut für Innovatives Rechnen und Programmstrukturen (IPD)

Informatik I WS 2002/03

Dozent: Prof. Dr.rer.nat. G. Goos

Übungsleiter: Markus L. Noga

<http://eins.info.uni-karlsruhe.de>

goos@ipd.info.uni-karlsruhe.de

noga@ipd.info.uni-karlsruhe.de

Übungsblatt 5

Ausgabe: 15.11.2001

Abgabe: 22.11.2002 14:00 Uhr

Einwurf im Keller des Informatik-Hauptbaus (Geb. 50.34)

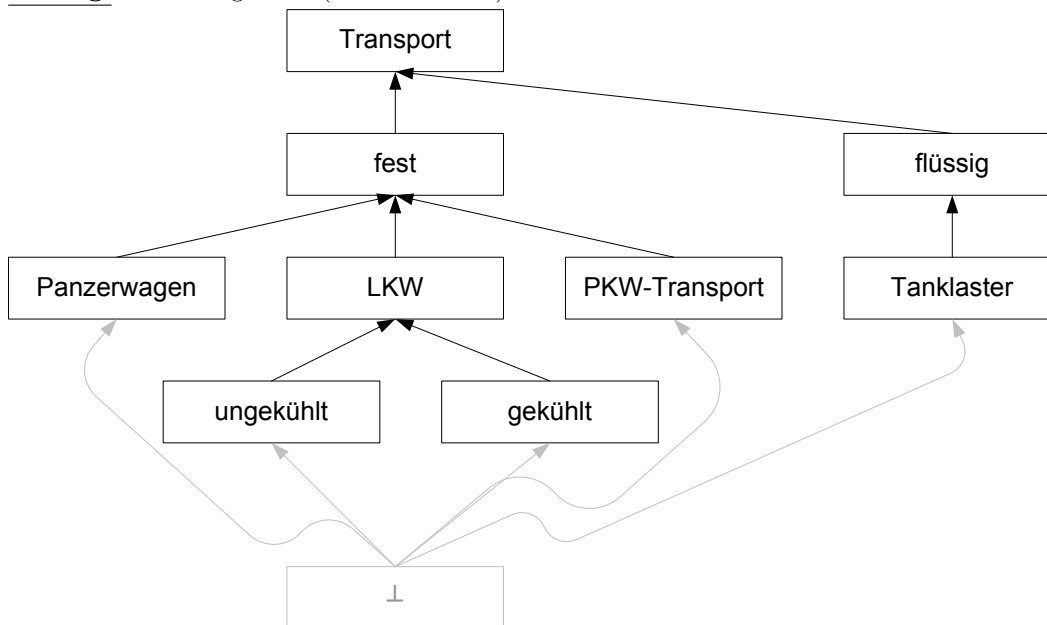
Aufgabe 1: Verbände (10 T- Punkte)

Je nach Eigenschaften der beförderten Güter muß die Spedition unterschiedliche Transportmittel einsetzen. Zur Verfügung stehen gekühlte und ungekühlte LKWs, Tankklaster, PKW-Transporter und gepanzerte Fahrzeuge für Werttransporte.

1.1 Halbverbände

Modellieren Sie die Eigenschaften der Güter als oberen Halbverband. Führen Sie dabei Oberbegriffe ein, die sich an Eigenschaften der Güter orientieren. Zeichnen Sie Ihr Ergebnis als Hasse-Diagramm und zeigen Sie die Erfüllung der Verbandseigenschaften.

Lösung: Siehe Diagramm (schwarzer Teil).



1.2 Verbände

Ergänzen Sie ihre Lösung zu einem Verband. Ist das Ergebnis ein vollständiger Verband? Begründen Sie.

Lösung: Siehe voriges Diagramm (schwarz + grau). Alle endlichen (oberen/unteren Halb-)Verbände sind vollständig.

Aufgabe 2: Endliche Automaten (20 T- Punkte)

In der Firma soll für die Fahrer ein Getränkeautomat angeschafft werden. Nur war dieser leider etwas zu teuer und so entschloss sich ein begabter Mitarbeiter diesen selber zu bauen.

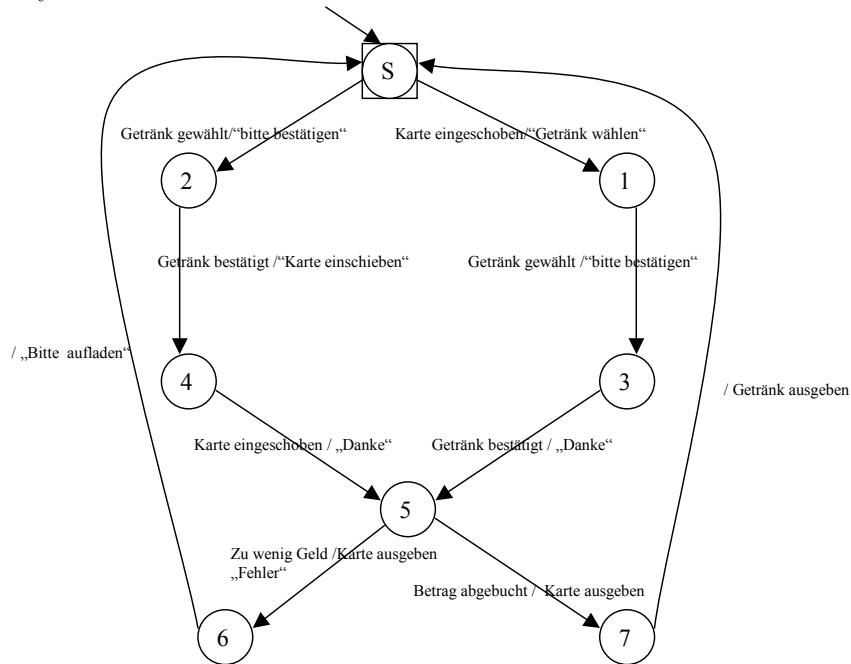
2.1

Erstellen Sie einen endlichen Automaten, der die Funktionsweise und Zustände des Getränkeautomaten darstellt. (Bezahlt wird wie in der Mensa mit Chipkey).

(Der Automat soll wie in der Mensa **alle sinnvollen** Eingaben beherrschen!)

Lösung:

Mealy-Automat



Ab dem 5. Zustand sind keine Benutzereingaben mehr erforderlich. Die nötigen Eingaben sind Automatenintern. Die Kanten zurück zum Startzustand benötigen gar keine Eingabe, sind also spontane Übergänge.

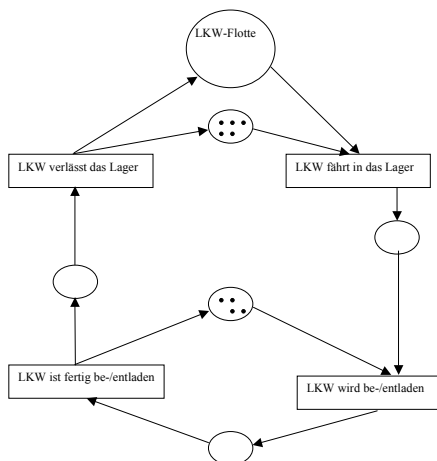
Aufgabe 3: Petrinetz (20 T- Punkte)

3.1

Durch die gute Auftragslage der Firma kann expandiert werden und es wurden auch gleich neue Zugmaschinen und Anhänger besorgt. Leider kann das Hauptlager maximal 5 Lastanhänger gleichzeitig aufnehmen. Entwerfen Sie ein Petrinetz für eine Ampelschaltung, welche dafür sorgt, das nie mehr als 5 LKW's im Lager sind. Zum Be-/Entladen sind nur 4 Plätze frei. Für die LKW-Flotte kann beliebig viel > 5 angenommen werden. Transitionen sind wie folgt zu verwenden:

1. LKW fährt ins Lager
2. LKW wird be-/entladen
3. LKW ist fertig be-/entladen
4. LKW verlässt Lager

Lösung:

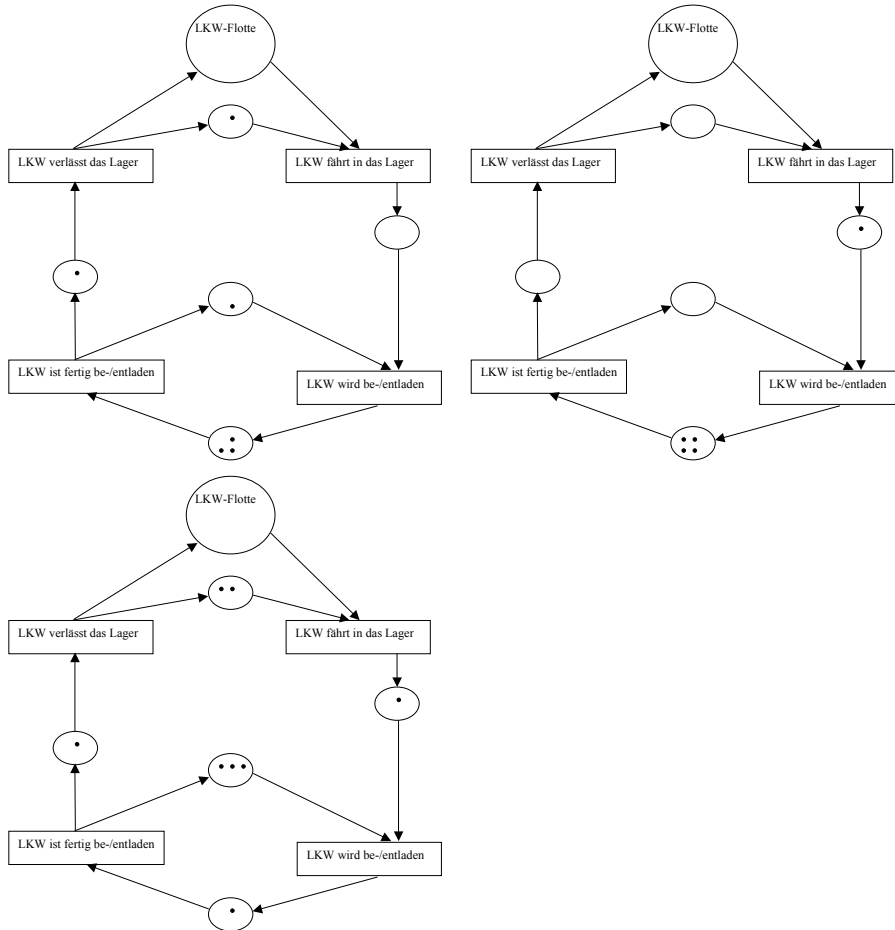


3.2

Zeichnen Sie die das oben erstellte Petrinetz in folgenden Zuständen:

1. 3 LKW werden be-/entladen, 1 ist fertig mit be-/entladen.
2. Das Lager ist voll und alle Ladebuchten sind belegt. 1 LKW muss warten.
3. Im Lager sind 3 LKW's. 1 wird abgefertigt, einer ist fertig und der 3. ist grad gekommen.

Lösung:



Aufgabe 4: Haskell Einstieg (10 P- Punkte)

Machen Sie sich mit der Haskell-Programmierungsumgebung vertraut und erstellen Sie zu den nachfolgenden Aufgabenstellungen Haskell-Programme. Testen Sie die Programme mit mindestens 3 Testeingaben.

4.1 Berechnung der grössten von zwei Zahlen

Lösung:

```
maxim :: Int -> Int -> Int
maxim a b = if a > b then a else b
```

4.2 Berechnung der Fakultätsfunktion

Lösung:

```
fact :: Int -> Int
fact 0 = 1
fact x = x * fact (x-1)
```

4.3 Berechnung des grössten gemeinsamen Teilers

Lösung:

```
ggT :: Int -> Int -> Int
ggT p q = if mod p q == 0 then q else ggT q (mod p q)
```

4.4 Berechnung der kleinsten Zahl aus einer Liste von 4 Elementen

Lösung:

```
minlist :: [Int] -> Int
minlist [x] = x
minlist (x:xs) | x < minlist xs = x
                | otherwise      = minlist xs
```