Aufgabe 1

Schreiben Sie für das folgende Problem Sattelpunktsuche ein Programm in der Programmiersprache PROG. (Solche Probleme treten bei der Multiplikation dünner Polynome auf).

Gegeben sei ein zweidimensionales Feld `b[0:m-1, 0:n-1]` von ganzen Zahlen und eine in `b` enthaltene ganze Zahl `x`. Es sei bekannt, daß die Zeilen und Spalten von `b` aufsteigend geordnet sind. Man schreibe ein Programm, das die Position von `x` in `b` angibt, d.h. Zahlen `i` und `j` mit `x=b[i,j]`. (Es genügt eine Fundstelle, wenn `x` mehrfach auftritt.) Versuchen Sie, die Anzahl der Vergleiche für den schlechtesten Fall zu minimieren.

(Freiwillig: Geben Sie Zusicherungen Q und R für Anfang und Ende des Programms an, die die Aufgabe des Programms spezifizieren. Beweisen Sie die partielle Korrektheit.) Hinweis: Man kann so beginnen:

```
i,p,q,j := 0,m-1,0,n-1;
do  ??? → i := i+1
□   ??? → p := p-1
□   ??? → q := q+1
□   ??? → j := j-1
od
```

und dann die *???* durch Schutzbedingungen ersetzen.

Lösung

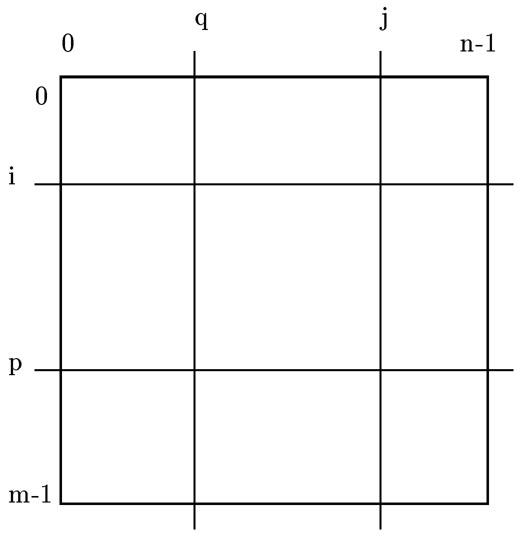
Die Zusicherungen am Anfang und am Ende des Programms sehen wie folgt aus:

$$\begin{array}{ll} Q: & x \in b[0:m-1, 0:n-1] \wedge \\ & \forall i \in \{0,\ldots,m-2\} \forall j \in \{0,\ldots,n-2\}: \\ & (b[i,j] \leq b[i+1,j] \wedge b[i,j] \leq b[i,j+1]). \end{array}$$

Q sichert also zu, daß x in dem Feld b enthalten ist und Zeilen und Spalten aufsteigend sortiert sind.

$$R: \quad 0 \leq i < m \wedge 0 \leq j < n \wedge b[i,j] = x$$

Beide Zusicherungen besagen, daß sich das gesuchte x in einem rechteckigen Bereich des Feldes b befindet. In Q umfaßt dieser Bereich noch das ganze Feld, in R besteht dieser Bereich nur noch aus einem einzigen Feldelement.



Eine Programminvariante, die ausdrückt, daß sich x in einem rechteckigen Bereich des Feldes befindet, sieht mit den Bezeichnungen aus dem obigen Bild so aus:

$$P: \quad 0 \leq i \leq p \leq m-1 \wedge 0 \leq q \leq j \leq n-1 \wedge x \in b[i:p, q:j].$$

Fortschritte im Programm werden dann einfach durch Verkleinerung des rechteckigen Bereichs gemacht. Die vier Möglichkeiten dazu sind:

$$i := i+1, \qquad p := p-1, \qquad q := q+1, \qquad j := j-1.$$

Das Programm sieht also bis jetzt so aus:

$i, p, q, j := 0, m-1, 0, n-1$
do $\;? \rightarrow \; i \; := \; i+1$
$\quad | \; ? \rightarrow \; p \; := \; p-1$
$\quad | \; ? \rightarrow \; q \; := \; q+1$
$\quad | \; ? \rightarrow \; j \; := \; j-1$
od

Wie sehen nun die Schutzbedingungen aus? Betrachte $i := i+1$. Die Ausführung dieser Zuweisung hält die Invariante P ein, sofern x sich nicht in Zeile i des Feldes befindet. Da Zeilen aber aufsteigend geordnet sind, kann das durch $b[i,j] < x$ überprüft werden, da dann ja auch alle Elemente in $b[i, 0:j]$ kleiner als x sein müssen. Die anderen Schutzbedingungen ergeben sich analog.

Das Programm mit Schutzbedingungen sieht dann also so aus:

$i, p, q, j := 0, m-1, 0, n-1;$
do $b[i,j] < x \;\; \rightarrow i := i+1$
$\quad | \; b[p,q] > x \;\; \rightarrow p := p-1$
$\quad | \; b[p,q] < x \;\; \rightarrow q := q+1$
$\quad | \; b[i,j] > x \;\; \rightarrow j := j-1$
od

Um die Korrektheit zu beweisen, sind die beiden mittleren Anweisungen nicht erforderlich. Das endgültige Programm wäre dann:

$i, j := 0, n-1;$
do $b[i,j] < x \;\; \rightarrow i := i+1$
$\quad | \; b[i,j] > x \;\; \rightarrow j := j-1$
od

Das Programm terminiert, da i den Wert $m-1$ nicht überschreiten und j den Wert 0 nicht unterschreiten kann.

Befindet sich x – wie zugesichert – in dem Feld, so gilt bei der Termination:

$$\neg b[i,j] < x \text{ und } \neg b[i,j] > x$$

also bleibt nicht anderes als $b[i,j] = x$.

Es werden insgesamt höchstens $m+n$ Vergleiche angestellt. Diese Anzahl läßt sich auch nicht entscheidend verbessen!

Nehmen wir einmal an, daß unser Feld quadratisch wäre, also $m = n$. Die Elemente der Nebendiagonalen $b[0, m-1], b[1, m-2], \ldots, b[m-1, 0]$ bilden eine ungeordnete Liste. Befindet sich das gesuchte x nun gerade auf der Nebendiagonalen, so sind im ungünstigsten Fall gerade m Vergleiche erforderlich.

Aufgabe 2

Gegeben sei folgende Präzedenzmatrix:

$x\backslash y$	A	B	C	D	E	F	G	H
A		$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$
B			$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	
C				$\lessdot$	$\lessdot$		$\lessdot$	
D					$\lessdot$		$\lessdot$	
E								
F							$\lessdot$	
G								
H							$\lessdot$	

Es soll gelten: x ist präzedent zu y, z.B. H ist präzedent zu G.

1. Erstellen Sie für diese Präzedenzmatrix:
   (a) den Präzedenzgraphen
   (b) den Konnektivitätsgraphen und
   (c) das Kausalnetz.
2. Wodurch unterscheiden sich Kausalnetze von B/E-Netzen?
3. Was bedeuten die Relationen li und co?
   (a) Was ist eine Klique, was ein Bezirk?
   (b) Geben Sie für das oben gesuchte Kausalnetz einige Linien und Schnitte an.
   (c) Zeigen Sie, daß in diesem Fall die Relationen li und co nicht transitiv sind.
   (d) Unter welchen Bedingungen sind diese Relationen transitiv?

Lösung

1. Der Präzendenzgraph, der Konnektivitätsgraph und das zugeordnete Kausalnetz finden sich in den Abbildungen 1 bis 3.

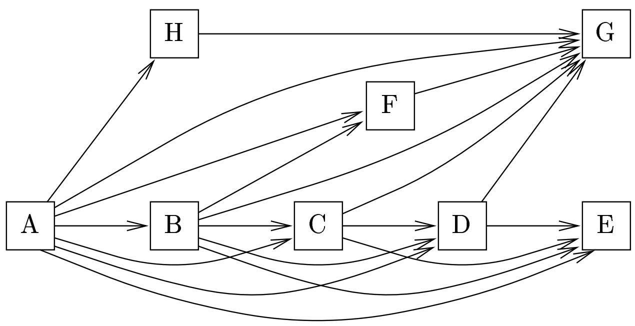


Abbildung 1: Der Präzendenzgraph

2. Ein Kausalnetz (S, T, F) (Def. 1.2.4) ist ein zyklenfreies, stellenunverzweigtes Netz (Def. 1.1.2). Ein B/E-Netz (S, T, F, m_0) (Def. 1.3.3) ist ein spezielles S/T-Netz (S, T, F, K, W, m_0) (Def. 1.3.2). Ein B/E-Netz ist also formal kein Netz, sondern besteht aus einem Netz mit Kapazitätzfunktion, Kantenbewertung und Anfangsmarkierung. Das unterliegende Netz (S, T, F) ist nicht notwendigerweise ein Kausalnetz, d.h. es kann Zyklen und verzweigte Stellen enthalten.

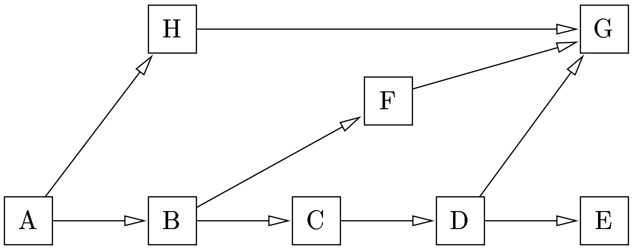


Abbildung 2: Der Konnektivitätsgraph

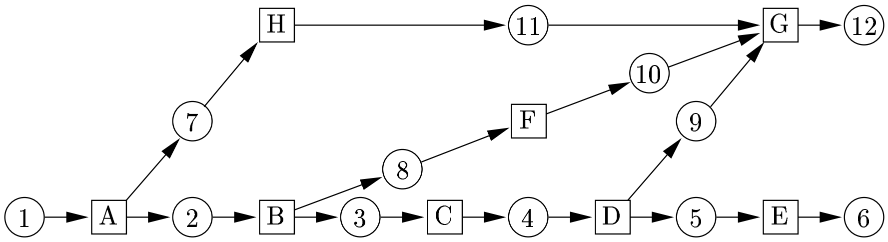


Abbildung 3: Das Kausalnetz

3. *co* und *li* werden üblicherweise in Kontext einer Halbordnung $<$ auf einer Menge X definiert: $li := < \cup <^{-1} \cup id_X$ drückt Vergleichbarkeit oder Identität von zwei Elementen aus, während $co := (X \times X - li) \cup id_X$ zwei unvergleichbare oder identische Elemente in Beziehung setzt. Beide Relationen sind reflexiv und symmetrisch.

   Da ein Kausalnetz (S, T, F) eine Kausalordnung $< := F^+$ auf $X := S \cup T$ definiert, sind *co* und *li* insbesondere für Kausalnetze definiert. Meist werden Kausalnetze (bzw. deren Kausalordnungen) als Darstellungen kausaler Präzedenzen interpretiert. Unter dieser Interpretation läßt sich *li* als (ungerichtete) kausale Abhängigkeit und *co* als kausale Unabhängigkeit deuten.

   (a) Für eine reflexive, symmetrische Relation $R \subseteq X \times X$ (wie z.B. *li* und *co*) ist eine Klique von R eine Teilmenge $A \subseteq X$ mit $A \times A \subseteq R$. Jedes Paar aus $A \times A$ ist also in der Relation R. Ein Ken von R ist eine maximale Klique, d.h. eine Klique, die nicht zu einer gößeren (bzgl. Mengeninklusion $\subset$) Klique erweitert werden kann.

   (b) Linien sind maximale Kliquen von *li* (Def. 1.2.10), also im obigen Kausalnetz beispielsweise die Mengen $\{1, 2, 3, 4, 5, 6, A, B, C, D, E\}$, $\{12, G, 10, F, 8, B, 2, A, 1\}$ und $\{1, A, 7, H, 11, G, 12\}$. Man beachte, daß es sich um Mengen handelt, die Reihenfolge der Elemente also keine Rolle spielt. Schnitte sind maximale Kliquen von *co* (Def. 1.2.10), also beispielsweise $\{11, 10, 9, 5\}$, $\{1\}$, $\{H, 8, 9, E\}$, $\{H, B\}$, $\{H, F, D\}$. Die ersten beiden Schnitte sind S-Schnitte, während es sind es sich bei den letzten beiden um T-Schnitte handelt.

   (c) Eine Relation R is transitiv, gdw. $(x, y) \in R$ und $(y, z) \in R$ auch $(x, z) \in R$ impliziert. *li* ist hier nicht transitiv, weil (beispielsweise) $(C, 1) \in li$ und $(1, 7) \in li$ aber $(C, 7) \notin li$. *co* ist hier ebenfalls nicht transitiv, weil $(H, 3) \in co$ und $(3, 11) \in co$ aber $(H, 11) \notin co$.

   (d) *co* is genau dann transitiv, wenn es keine zwei verschiedenen Elemente $(x, z) \in li$ gibt, die in *co* zu einem dritten Element y stehen, d.h. $(x, y) \in co$ und $(y, z) \in co$. Durch Vertauschen von *co* und *li* erhält man die duale Aussage für *li*. Hinreichend (aber nicht notwendig) für die Transitivität von *co* bzw. *li* sind die trivialen Fälle $li = id_X$ bzw. $co = id_X$. Diese Fälle liegen bei Kausalnetzen ohne Präzedenzen bzw. bei sequentiallen Kausalnetzen vor.

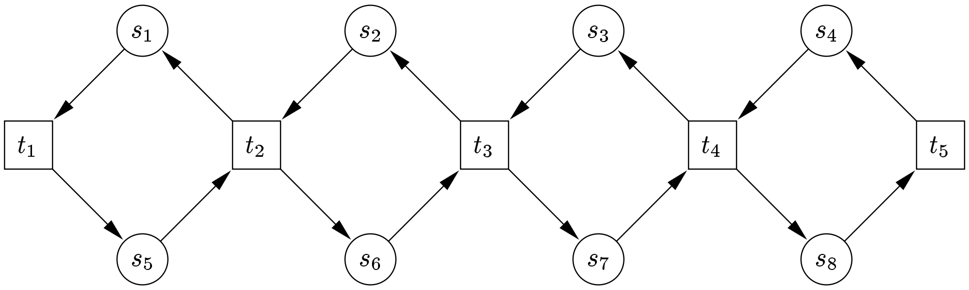


Abbildung 4: Ein Petrinetz

Aufgabe 3

Ist im S/T-Netz nach Abb. 4 ausgehend von der Anfangsmarkierung $(1,1,1,1,0,0,0,0)$ die Markierung $(0,0,0,0,1,1,1,1)$ erreichbar? Wenn ja, wie?

Lösung

Die Endmarkierung ist mit der Schaltfolge $t_1 t_2 t_3 t_4 t_1 t_2 t_3 t_1 t_2 t_1$ erreichbar. Dies kann man durch Erzeugung des Erreichbarkeitsgraphen festellen. Dabei ist die Beobachtung hilfreich, daß die oberen und die unteren Stellen jeweils komplementär sind. Dies garantiert eine maximale Anzahl von Marken und daher auch einen endlichen Erreichbarkeitsgraph.

Ohne die Gewißheit, daß der Erreichbarkeitsgraph endlich ist, müßte ein anderes Verfahren zur Erreichbarkeitsanalyse hinzugezogen werden.

Aufgabe 4

Im folgenden geben wir einige Bedingungen an, die für ein S/T-Netz $N = (S, T, F, K, W, m_0)$ erfüllt sein könnten. Geben Sie je ein Netz an, für das die Bedingung gilt, und eins, für das sie nicht gilt. Wenn es kein entsprechendes Netz gibt, beweisen Sie die Unmöglichkeit.

a) $F = (S \times T) \cup (T \times S)$.

b) $F = (S \cup T) \times (T \cup S)$.

c) $S \cup T$ ist unendlich.

d) $F \cap F^{-1} = \emptyset$.

e) $\exists t \in T : m_0 \xrightarrow{t}$.

f) $\forall t \in T : m_0 \xrightarrow{t}$.

g) $\exists w \in T^* : m_0 \xrightarrow{w} m_0$.

h) $\exists w \in T^+ : m_0 \xrightarrow{w} m_0$.

i) $\forall w \in T^* : m_0 \xrightarrow{w} m_0$.

j) $\forall w \in T^* : m_0 \xrightarrow{w} \Rightarrow \exists v \in T^* : m_0 \xrightarrow{wv} m_0$.

k) Erfinden Sie drei interessante Eigenschaften und untersuchen Sie sie auf die gleiche Weise.

Lösung

In Abb. 5 sind einige Netze aufgeführt, die in den Lösungen der Unteraufgaben als Beispiele dienen.

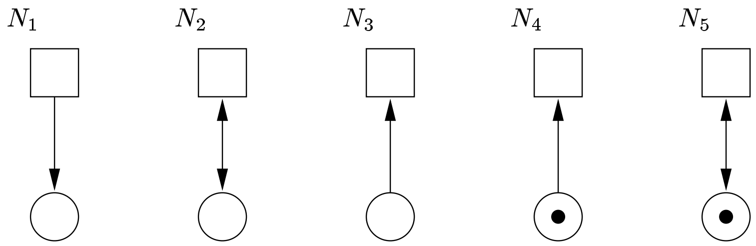


Abbildung 5: Einige Netze

a) Beispiel: N_2. Gegenbeispiel: N_1.

b) Diese Bedingung ist nie erfüllt. T ist nach Definition eines S/T-Netzes nicht leer, also gibt es ein $t \in T$ und $(t,t) \in (S \cup T) \times (T \cup S)$. Da aber $(t,t) \notin F$ wegen $F \subseteq (S \times T) \cup (T \times S)$, gilt $F \neq (S \cup T) \times (T \cup S)$.

c) Diese Bedingung ist nie erfüllt. T ist nach Definition endlich, ebenso S. Daher ist auch $S \cup T$ endlich.

d) Beispiel: N_1. Gegenbeispiel: N_2.

e) Beispiel: N_4. Gegenbeispiel: N_3.

f) Beispiel: N_4. Gegenbeispiel: N_3.

g) Diese Bedingung ist immer erfüllt, da stets $\lambda \in T^*$ und $m_0 \xrightarrow{\lambda} m_0$.

h) Beispiel: N_5. Gegenbeispiel: N_2.

i) Beispiel: N_5. Gegenbeispiel: N_1.

j) Beispiel: N_5. Gegenbeispiel: N_1.

k) Keine Musterlösung. Think!

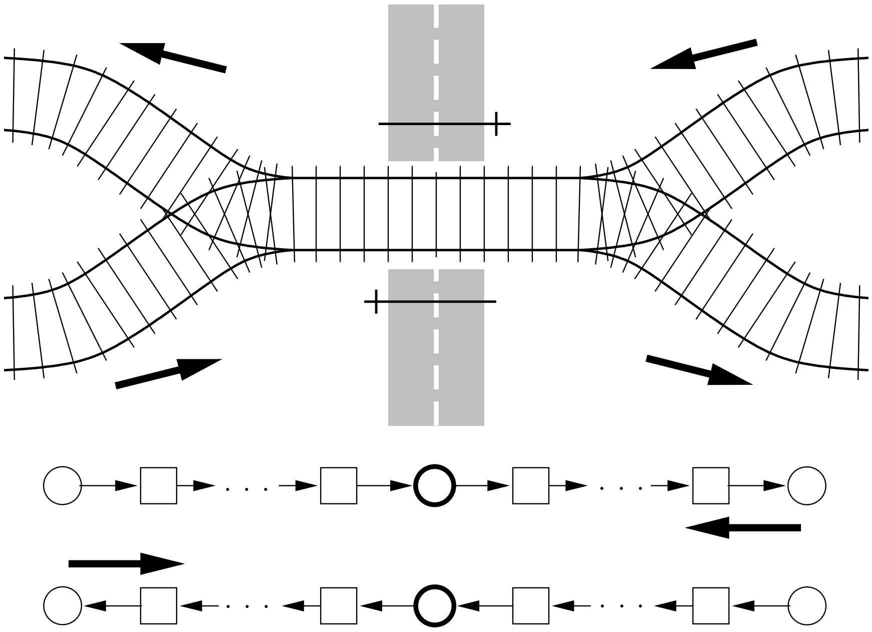
Aufgabe 5

Zwei Eisenbahnstrecken werden an einem beschrankten Bahnübergang auf einem einzigen Gleis geführt (Dies ist die Standardsituation bei der AKN.)

Modellieren Sie die notwendige Synchronisation durch ein S/T-Netz, indem Sie von folgendem Netz für die beiden Gleise ausgehen. Dabei bedeutet für jedes Gleis die fett dargestellte Stelle den „kritischen Abschnitt“ (die Kreuzung). Die Marken bedeuten Züge. (Autos auf der Straße werden nicht dargestellt.) Die Transitionen stellen wichtige Aktionen des Zuges dar, wie z.B. das Vorbeifahren an einem Signal oder das Auslösen eines Sensors.

a) Lösen Sie das Problem zunächst für je einen Zug auf jeder Strecke ohne Straße und Schranken (je eine Marke wie angegeben). Signale (nicht aktivierte Transitionen) stoppen ggf. die Züge.

b) Betrachten Sie auch die Schranken und deren korrektes Öffnen und Schließen. Benutzen Sie dazu das obige Teilnetz „Motorsteuerung“ und steuern es durch Sensoren an den Gleisen.



c) Nehmen Sie zusätzlich an, daß maximal $n > 0$ (z.B. $n = 5$) Züge hintereinander auf jeder Strecke fahren können. Es sei erlaubt, daß sich mehrere Züge in einer Richtung in dem gleichen Streckenabschnitt (auf der gleichen Stelle) befinden. Dies soll so interpretiert werden, daß sich die Züge in sicherem Abstand in verschiedenen Teilen des Streckenabschnitts bewegen. Synchronisieren Sie wieder den wechselseitigen Ausschluß auf der Kreuzung und das richtige Öffnen und Schließen der Schranken. (Beispielsweise: Nicht öffnen, wenn Zug in gleicher Richtung folgt.)

Lösung

Die Lösungsnetze finden sich auf den folgenden Seiten.

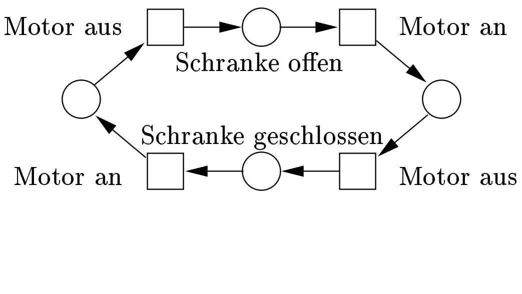
Aufgabe 6

Zeichnen Sie einen Prozeß des Netzes aus Abb. 6, bei dem jede Transition vorkommt. Erklären Sie am Beispiel der Transition b, wie sich hier die axiomatische Prozeßdefinition widerspiegelt.

Lösung

In Abb. 7 ist ein möglicher Prozeß des Netzes dargestellt, so wie er nach der operationalen Prozeßdefinition erstehen könnte.

Bei der Überprüfung der axiomatischen Definition stellt sich heraus, daß die Menge der Stellen



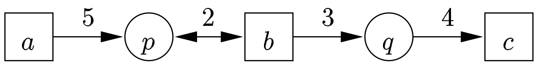


Abbildung 6: Ein Netz

ohne Vorbedingung kein Schnitt ist. Dies ist jedoch grundsätzlich bei Transitionen ohne Vorbedingungen unvermeidlich.

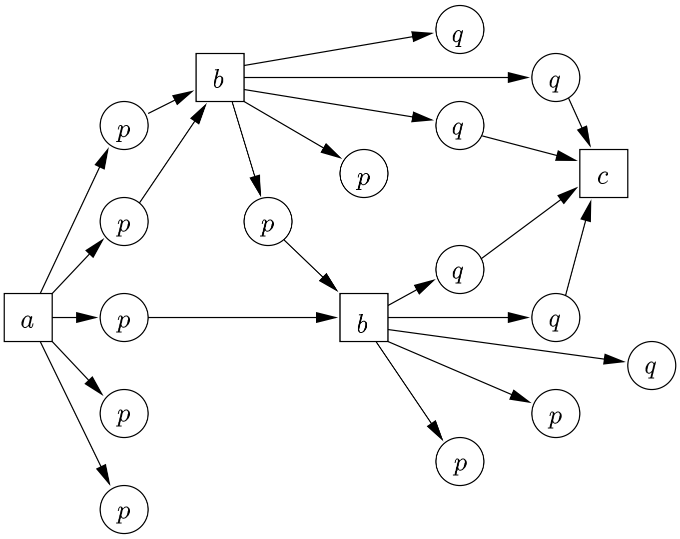


Abbildung 7: Ein Prozeß

Aufgabe 7

Das Netz aus Abbildung 8 beschreibt das Leser/Schreiber-Problem. (Vgl. auch *Jessen/Valk: Rechensysteme*, S. 42 ff.)

a) Verändern Sie das Netz so, daß Schreiber Vorrang vor Lesern haben, d.h. daß ein Schreiber, der sich zum Schreiben angemeldet hat, Vorrang vor bereits angemeldeten Lesern hat, jedoch keine lesenden Leser verdrängt.

b) Wodurch unterscheiden sich Stellen-Transitions-Netze von Bedingungs-Ereignis-Netzen?

c) Bilden Sie einen asynchronen Prozeß zu dem ursprünglichen Netz, bei dem mindestens folgende Situationen auftreten:

   1. Es wird zweimal geschrieben. (s ist markiert.)
   2. Es finden vier Lesezugriffe zusammenhängend statt, d.h. ohne daß ein Schreibauftrag zwischendurch schreibt.

d) Bilden Sie einen entsprechenden asynchronen Prozeß für das unter a) entwickelte Netz.

Ergänzung: Folgende Begriffe sollten für die Übung nochmals angesehen werden: Markierung, aktiviert, schalten, serieller Prozeß, Erreichbarkeitsmenge, Fallgraph, Konfliktstelle, Kontakt, komplementäre Stelle, asynchroner Prozeß.

Lösung

1. Siehe Abb. 9.

2. Ein B/E-Netz (S, T, F, m_0) ist ein spezielles S/T-Netz (S, T, F, K, W, m_0) mit Kapazität 1 für alle Stellen $s \in S$ und Kantengewicht 1 für alle $(x, y) \in F$.

3. In Abb. 10 ist folgender Prozeß dargestellt: Zuerst wird einmal geschrieben, dann finden vier Lesezugriffe statt, und schließlich wird noch einmal geschrieben. Die Namen der Netzelemente sind nicht angeben sondern nur die Beschriftungen. Die Stellen oben rechts sind mit den Stellen unten links zu identifizieren.

4. Der Prozeß in Abb. 10 kann durch Hinzufügen von mit p beschrifteten Stellen und entsprechenden Abhängigkeiten so modifiziert werden, daß sich ein Prozeß den Netzes in Abb. 9 ergibt. Insbesondere is zu beachten, daß die drei nebenläufigen Vorkommen von b durch p sequentialisiert werden.

Aufgabe 8

Berechnen Sie eine optimale, konfliktfreie Partition des Bindungsgraphen aus Abbildung 11 mit folgenden Fällen für festgelegte Ecken:

a) $\{p_1, p_2\}$, b) $\{p_1, p_3\}$, c) $\{p_1, p_4\}$, d) $\{p_2, p_3\}$,
e) $\{p_2, p_4\}$, f) $\{p_3, p_4\}$, g) $\{p_1, p_2, p_3, p_4\}$.

Wenden Sie auf den Fall g) das Verfahren mit dem Algorithmus von Kruskal an und prüfen Sie, ob damit eine optimale Lösung gefunden wird.

Lösung

Die optimalen Partitionierungen sind in Abb. 12 dargestellt. Die 20er Kanten nennen wir im folgenden „innere Kanten", die restlichen „äußere Kanten".

- Beweisidee für Optimalität von (a): Die Kosten der angegebenen Partitionierung sind 31. Jede andere Partitionierung fällt in eine der folgenden Klassen: (1) Keine innere Kante wird durchtrennt. (2) Mindestens eine innere Kante wird durchtrennt. Im Fall (1) muß mindestens eine äußere Kante durchtrennt werden. Dies ist im günstigsten Fall nur die Kante $\{p_1, q_1\}$. Im Fall (2) muß mindestens eine weitere Kante durchtrennt werden. Dies würde auf jeden Fall zu höhreren Kosten als 31 führen.

- Die Aufgaben (b)-(f) sind analog zu (1) zu behandeln.

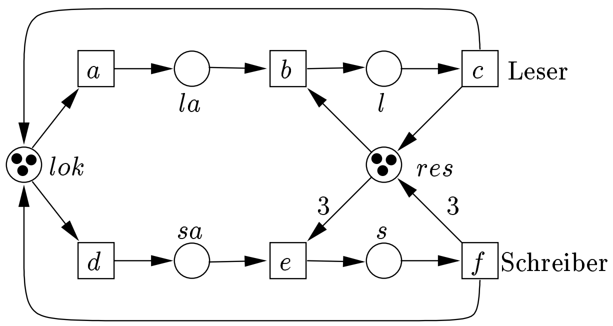


Abbildung 8: Netz zum Leser/Schreiber-Problem

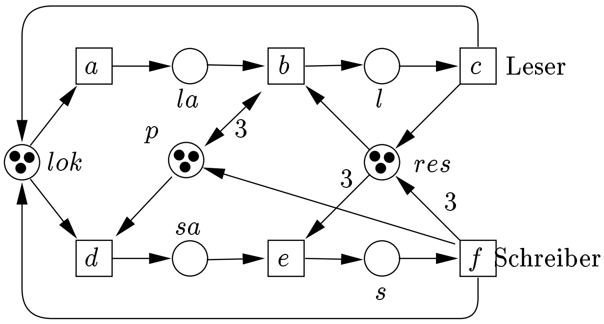


Abbildung 9: Priorisiertes Leser/Schreiber Problem

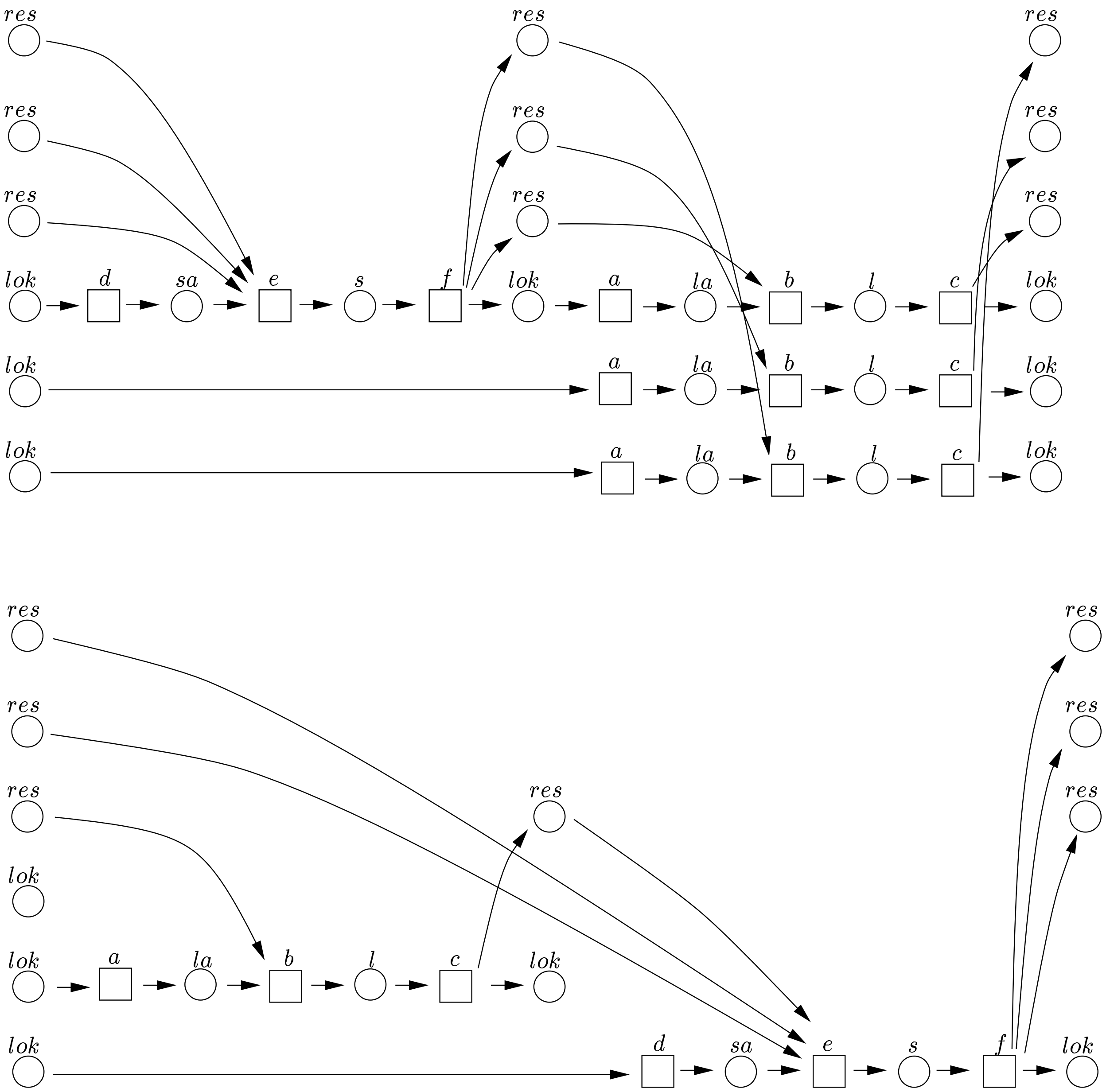


Abbildung 10: Ein Prozeß des ursprünglichen Netzes

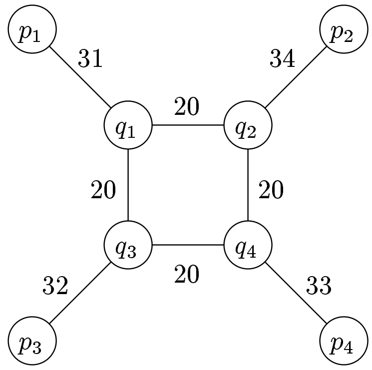


Abbildung 11: Bindungsgraph

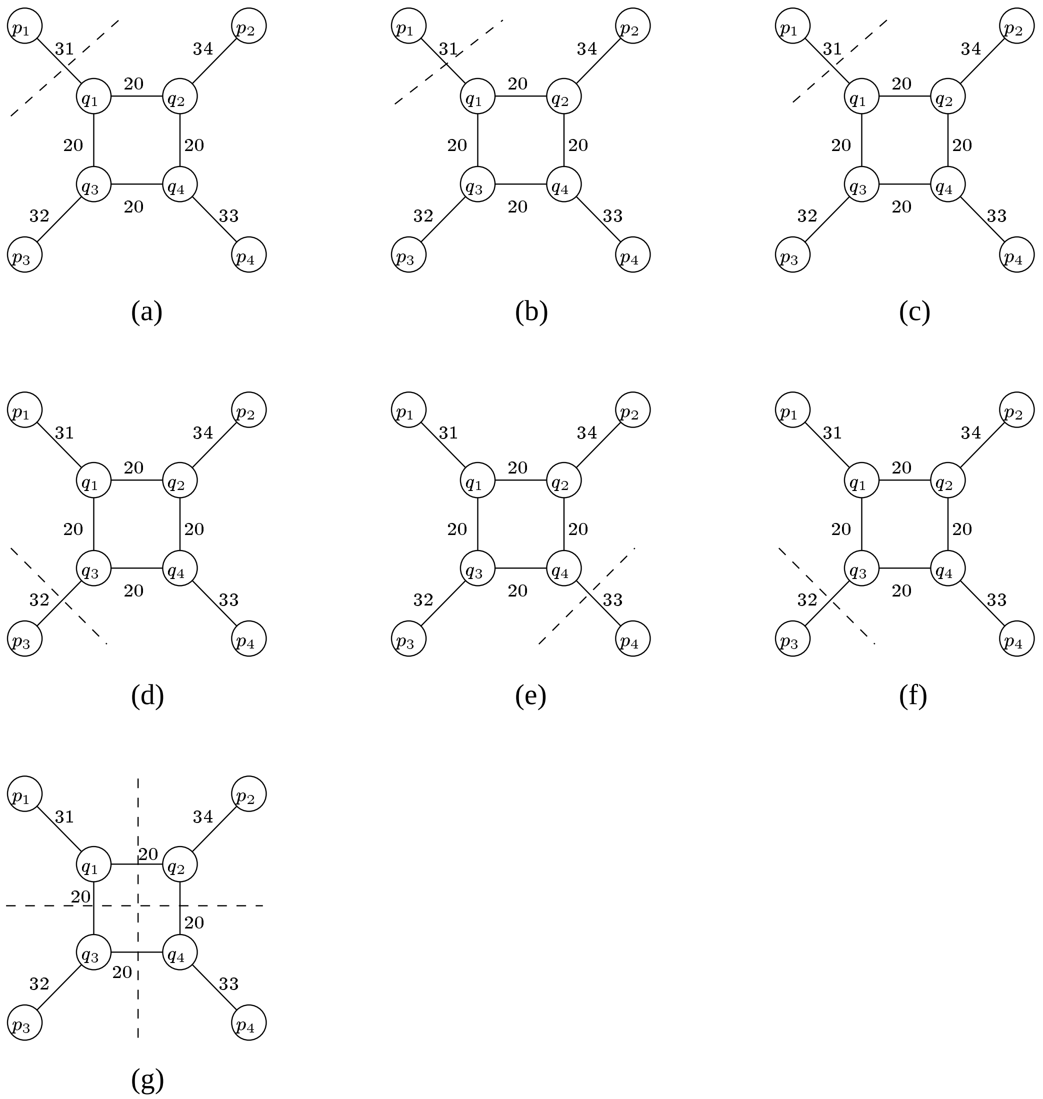


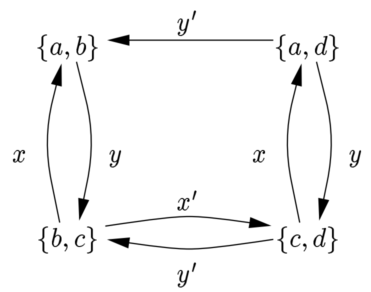
Abbildung 12: Optimale Partitionierungen

- Beweisidee für die Optimalität von (g): Die Kosten der angegebenen Partitionierung sind 80. Jede andere Partitionierung trennt genau $i \in \{0, \ldots, 3\}$ innere Kanten durch. Dies sind 4 Fälle. In den Fallen $i = 0$ und $i = 1$ müssen 3 äußere Kanten durchgetrennt werden. Im Fall $i = 2$ müssen 2 äußere Kanten getrennt werden. Im Fall $i = 3$ ist 1 äußere Kante durchzutrennen. In allen Fällen ergeben sich höhere Kosten als 80.

- Wird der Kruskal-Algorithmus auf (g) angewandt so ergibt sich tatsächlich die optimale Partitionierung: Man fügt einen Knoten x in den Graphen ein, der mit großem Gewicht (z.B. 1000) mit den vier Knoten p_1, p_2, p_3 und p_4 verbunden wird. Dann berechnet man einen aufspannenden Baum mit maximalen Kosten. Der Algorithmus konstruiert den aufspannenden Baum, indem zuerst die 1000er Kanten hinzugefügt werden und dann die Kanten $\{p_2, q_2\}$, $\{p_4, q_4\}$, $\{p_3, q_3\}$ und $\{p_1, q_1\}$ in dieser Reihenfolge.

Aufgabe 9

Gegeben sei untenstehender Übergangsgraph. Es ist nicht klar, ob es ein B/E-Netz gibt, das gerade diesen Übergangsgraphen als Fallgraphen besitzt. Untersuche diese Frage und konstruiere dieses B/E-Netz, sofern dies möglich ist, oder beweise seine Nichtexistenz.



Lösung

Aus dem B/E-Netz in Abb. 13 läßt sich – beispielsweise mit dem Anfangsfall $\{a, b\}$ – leicht das in der Aufgabe geforderte Fall/Transitionssystem konstruieren. Die reinen Vor- und Nachbedingungen ergeben sich direkt aus dem Fall/Transitionssystem, wir müssen jedoch zusätzlich x' mit der Nebenbedingung c versehen, damit x' nicht im Fall $\{a, b\}$ schalten kann.

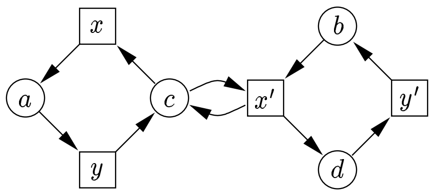


Abbildung 13: Das zum Fallgraphen passende Netz

Aufgabe 10

Sei N ein beliebiges S/T-Netz N mit Transitions-Anschrift (Def. siehe S. 72). Wir haben bereits die Schaltfolgen $F(N)$, die terminalen Schaltfolgen $F(N, M_E)$, die Typfolgen $L(N)$ und die terminalen Typfolgen $L(N, M_E)$ kennengelernt (Seite 40 bzw. 72). Damit sind jedem S/T-Netz N evtl.

abhängig von einer Menge von Endmarkierungen M_E formale Sprachen $F(N)$, $F(N, M_E)$, $L(N)$ und $L(N, M_E)$ zugeordnet.

a) Warum ist jede Schaltfolgensprache $F(N)$ eine Typfolgensprache $L(N')$?

b) Zeigen Sie, daß jede reguläre Sprache gleich einer terminalen Typfolgensprache $L(N, M_E)$ ist.

c) Gibt es eine endliche reguläre Sprache, die keine Schaltfolgensprache $F(N)$ ist ?

d) Warum ist nicht jede Sprache $F(N)$, $F(N, M_E)$, $L(M)$ bzw. $L(M, M_E)$ regulär ?

e) Zeigen Sie, daß $F(N)$, $F(N, M_E)$, $L(N)$ und $L(N, M_E)$ immer rekursiv aufzählbar sind. Warum ist es leicht zu sehen, daß $F(N)$ und $F(N, M_E)$ sogar rekursiv (d.h. entscheidbar) sind ?

f) Warum ist die Familie aller (terminalen) Schaltfolgensprachen von S/T-Netzen abgeschlossen unter Durchschnitt ? Gilt entsprechendes auch für die (terminalen) Typfolgensprachen ?

g) Geben Sie ein Beispiel für eine nicht-kontextfreie terminale Typfolgensprache $L(N, M_E)$. Gibt es vielleicht auch eine nicht-kontextfreie terminale Schaltfolgensprache $F(N, M_E)$?

Lösung

a) Wähle das Netz von N' wie N und die Identität als Transitions-Anschrift.

b) Die reguläre Sprache wird von einem endlichen Automaten A akzeptiert. Konstruiere das dem endlichen Automat entsprechende Netz N, das für jeden Zustand von A genau eine Stelle besitzt und für jeden Zustandsübergang von A eine entsprechende Transition enthält. Dieses Netz ist transitions-unverzweigt. Als Anfangsmarkierung wählt man den Startzustand von A, als Menge der Endmarkierungen M_E die Menge aller $\{q\}$ für Endzustände q von A.

c) Ein Beispiel für eine endliche reguläre Sprache, die keine Schaltfolgensprache ist, ist beispielsweise die leere Sprache, denn jede Schaltfolgensprache enthält nach Definition mindestens das leere Wort. Um Spitzfindigkeiten zu vermeiden, können wir aber auch jede nicht-präfix-abgeschlossene, endliche, reguläre Sprache wählen.

d) Man wähle ein Netz mit einer Stelle s sowie zwei Transitionen t_1 und t_2 und $F = \{(t_1, s), (s, t_2)\}$. Anfangsmarkierung sei $\emptyset$, Endmarkierung sei ebenfalls nur $\emptyset$. Die Sprachen $F(N)$, $F(N, M_E)$, (und damit auf $L(M)$ und $L(M, M_E)$ bei Wahl der Identität als Transitions-Anschrift) sind offensichtlich nicht regulär.

e) Das Schalten eines Netzes kann durch eine TM simuliert werden: Die Transitionen werden in der Reihenfolge in der sie in dem Eingabewort auftauchen geschaltet. Die TM bricht ab und akzeptiert nicht, wenn das Schalten nach der Schaltregel nicht möglich ist. Im Fall der Sprache $F(N)$ akzeptiert die TM nach Verarbeitung das Eingabewort. Im Fall der Sprache $F(N, M_E)$ akzeptiert die TM falls die aktuelle Markierung in der Menge der M_E enthalten ist. Damit ist gezeigt, daß $F(N)$ und $F(N, M_E)$ rekursiv sind. Da $F(N)$ bzw. $F(N, M_E)$ damit auch rekursiv aufzählbar sind, können von einer anderen TM auch die entsprechenden Typfolgen (also $L(N)$ bzw. $L(N, M_E)$ nach Übersetzung mittels der Transitions-Anschrift aufgezählt werden.

f) Gegeben seien S/T-Netze N und N' mit Transitions-Anschrift und Endmarkierungen M_E bzw. M'_E. Der Schnitt $F(N) \cap F(N')$ wird als Schaltfolgensprache durch das folgende Netz N'' generiert: Die Menge der Stellen sei $S \uplus S'$. Die Menge der Transitionen sei $T \times T'$. Die F Relation wird von N und N' ererbt, d.h. $sF''(t, t')$ gilt genau dann wenn sFt oder $sF't'$ gilt, und entsprechend für $(t, t')F''s$. Das Netz N'' erhalten wir nun indem wir die Transitionen

(t,t') löschen für die t und t' verschieden sind. Schliesslich benennen wir die verbleibenden Tranistionen (t,t) zu t um. Es ist offensichtlich, daß jede Schaltfolge des neuen Netzes auch Schaltfolge in N und N' sein muß und umgekehrt. Um den Schnitt $F(N,M_E) \cap F(N',M'_E)$ zu erhalten, müssen wir noch eine geeignete Endmarkierung angeben: Diese enthalten genau die Markierungen M, für die $M \cap S$ und $M \cap S'$ Endmarkierungen aus M_E bzw. M'_E sind. Um den Schnitt $L(N) \cap L(N')$ bzw. $L(N,M_E) \cap L(N',M'_E)$ zu generieren, löschen wir im obigen Netz mit Stellen $S \uplus S'$ und Transitionen $T \times T'$ nur diejenigen Transitionen (t,t') für die t und t' verschiedene Anschriften haben. Als Transitionsbeschriftung für (t,t') wählen wir dann die Anschrift von t (die gleich der von t' ist).

g) Eine nicht-kontextfreie terminale Typfolgensprache ist beispielsweise $\{a^n c^n e^n : n \in I\!N\}$. Eine nicht-kontextfreie terminale Schaltfolgensprache ist $\{a^n bc^n de^n f\}$. Für beide Fälle können wir das gleiche Netz verwenden, wobei im ersten Fall die Transitionen b, d und f durch leere Anschrift ausgeblendet werden: Wir haben Transitionen t_a, t_b, t_c, t_d, t_e, t_f und Stellen s_{ac}, s_{ce}, s_a, s_c, s_e, s_f wobei s_a, s_c und s_e Nebenbedingungen zu t_a, t_c bzw. t_e sind, t_b von s_a nach s_c und t_d von s_c nach s_e geht und s_{ac} und s_{ce} von t_a nach t_c bzw. von t_c nach t_e gehen. Schließlich haben wir noch eine Verbindung von s_e zu t_f. Anfangsmarkierung ist $\{s_a\}$. Endmarkierung ist die leere Markierung.

Aufgabe 11

In dieser Aufgabe soll das Leser/Schreiber-Problem mit Semaphoren gelöst werden. Verwenden Sie dabei die um die Funktionen `P()` und `V()` erweiterte Sprache PROG (s. S. 143/144).

a) Lösen Sie das Problem für $n > 0$ Lese- und $m > 0$ Schreib-Aufträge mit der Spezifikation (vergl. S. 188):

- Wenn ein Schreiber schreibt, dann darf kein Leser lesen.
- Es darf höchstens ein Schreiber schreiben.
- Wenn ein Leser liest, dann darf kein Schreiber schreiben.

b) Lösen Sie die Aufgabe so, daß keine maximale Anzahl von Aufträgen vorgegeben werden muß, indem beim Entwurf der Protokolle für den i-ten Leser und den j-ten Schreiber n bzw. m nicht benutzt werden.

c) Schreiber sollen priorisierten Zugriff haben.

Lösung

a) Leser-/Schreiber-Problem mit m Leseaufträgen.

Global:

```
var L := m : semaphor;
var X := 1 : semaphor;
```

Leser:

```
P(L);
read;
V(L);
```

Schreiber:

```
P(X);
do m times P(L);
write;
do m times V(L);
V(X);
```

b) Leser-/Schreiber-Problem mit unbegrenzter Anzahl von Leseaufträgen.

Global:

```
var anz := 0 : integer;
var S := 1 : semaphor;
var X := 1 : semaphor;
```

Leser:

```
P(X);
if anz=0 → P(S) □ else → skip fi;
anz:=anz+1;
V(X);
read;
P(X);
anz:=anz-1;
if anz=0 → V(S) □ else → skip fi;
V(X);
```

Schreiber:

```
P(S);
write;
V(S);
```

Die Kernidee für dieses Programm ist es, immer den ersten Leser das Schreibsemaphor sperren zu lassen und es den letzten Leser wieder freigeben zu lassen. Mit einem zusätzlichen Semaphor `X` erreichen wir den wechselseitigen Ausschluß der Leser während der Veränderungen an der Variable `anz`, damit diese immer korrekt die Anzahl der lesenden Prozesse angibt.

c) Leser-/Schreiber-Problem mit unbegrenzter Anzahl von Leseaufträgen und Priorisierung der Schreiber. Wir beginnen mit dem Programm aus der letzten Unteraufgabe, fügen jedoch ein weiteres Semaphor ein, mit dem der erste Schreiber alle weiteren Leser bei der Anmeldung warten läßt.

Global:

```
var W := 1 : semaphor;
...
```

Leser:

```
P(W);
V(W);
...
```

Schreiber:

```
P(W);
...
V(W);
```

Aufgabe 12

Betrachten Sie ein Auftragssystem A aus den – nicht untereinander präzedenten – Aufträgen a_1, a_2, a_3.

a) Konstruieren Sie zu der folgenden Interpretation ein vollständiges schematisches Auftragssystem (Anfangsauftrag a_0, Endauftrag a_4):

$$\begin{aligned} a_1 &: \quad x := y + 1 \\ a_2 &: \quad y := z + 1 \\ a_3 &: \quad z := x + 1 \end{aligned}$$

b) Betrachten Sie die zugehörigen Ausführungsfolgen:

$$\begin{aligned} w_1 &= l_0 s_0[x,y,z] l_1[y] l_3[x] s_1[x] l_2[z] s_2[y] s_3[z] l_4[x,y,z] s_4 \\ w_2 &= l_0 s_0[x,y,z] l_1[y] l_3[x] s_3[z] l_2[z] s_1[x] s_2[y] l_4[x,y,z] s_4 \end{aligned}$$

Entscheiden Sie nach dem in der Vorlesung angegebenen Verfahren, ob die Folgen serialisierbar sind und geben Sie ggf. alle Serialisierungen an.

c) Diskutieren Sie die Ergebnisse von Punkt b) anhand der Interpretation von Punkt a).

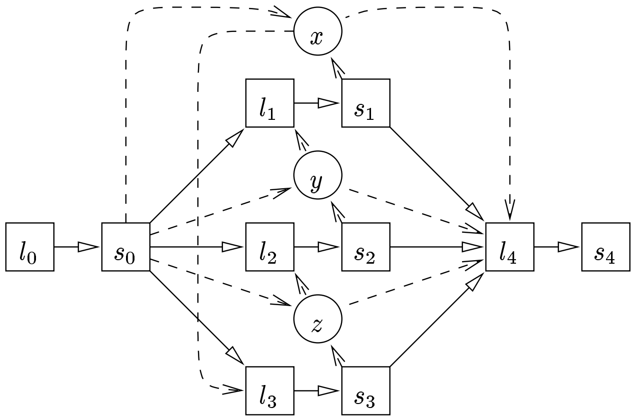


Abbildung 14: Schematisches Auftragssystem

Lösung

a) In Abb. 14 ist das schematische Auftragssystem graphisch dargestellt. Es wurden Anfangs- und Endeaufträge hinzugefügt. Die gelesenen Variablen ergaben sich direkt aus der Aufgabenstellung.

b) Die liest-von-Relation für w_1 ergibt sich aus:

$$w_1 = l_0 s_0[x,y,z] l_1[y] l_3[x] s_1[x] l_2[z] s_2[y] s_3[z] l_4[x,y,z] s_4$$

Also z.B. l_3 liest x von s_0. Die Wertübertragungsrelation ist $\text{WÜ}(w_1) = \{(a_0,a_1),(a_0,a_2),(a_0,a_3),(a_1,a_4),(a_2,a_4),(a_3,a_4)\}$. Damit sind alle Aufträge relevant. Zufällig entspricht die Wertübertragungsrelation der Präzedenzrelation. Bei der Konstruktion des Serialisierungsgraphen ist daher $\lessdot' \cup \lessdot_1 = \text{WÜ}$.

Jetzt muß noch je ein Element aus jeder der folgenden Zeilen zur Relation $\lessdot_2$ hinzugenommen werden.

- (a_1,a_0) oder (a_3,a_1), denn a_1 schreibt x und a_3 liest x von a_0.
- (a_2,a_0) oder (a_1,a_2), denn a_2 schreibt y und a_1 liest y von a_0.
- (a_3,a_0) oder (a_2,a_3), denn a_3 schreibt z und a_2 liest z von a_0.
- (a_0,a_1) oder (a_4,a_0), denn a_0 schreibt x und a_4 liest x von a_1.
- (a_0,a_2) oder (a_4,a_0), denn a_0 schreibt y und a_4 liest y von a_2.
- (a_0,a_3) oder (a_4,a_0), denn a_0 schreibt z und a_4 liest z von a_3.

Wir suchen nur solche Serialisierungsgraphen, die zyklenfrei sind. Daher sind z.B. (a_1,a_0), (a_2,a_0) und (a_3,a_0) prinzipiell unmöglich. Also müssen die Alternativen (a_3,a_1), (a_1,a_2) und (a_2,a_3) hinzugenommen werden. Wieder ergibt sich ein Zyklus. Also ist w_1 nicht serialisierbar.

Die liest-von-Relation für w_2 ergibt sich aus:

$$w_1 = l_0 s_0[x,y,z] l_1[y] l_3[x] s_3[z] l_2[z] s_1[x] s_2[y] l_4[x,y,z] s_4$$

Die Wertübertragungsrelation ist nun $\text{WÜ}(w_2) = \{(a_0,a_1),(a_0,a_3),(a_3,a_2),(a_1,a_4),(a_2,a_4),(a_3,a_4)\}$. Damit sind alle Aufträge relevant. Bei der Konstruktion des Serialisierungsgraphen ist $\lessdot' \cup \lessdot_1 = \text{WÜ} \cup \{(a_0,a_2)\}$, siehe dazu Abb. 15.

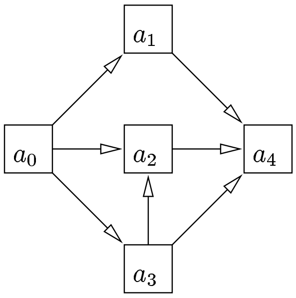


Abbildung 15: $\lessdot' \cup \lessdot_1$ für w_2

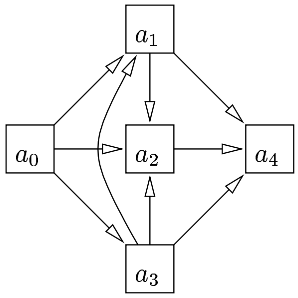


Abbildung 16: Serialisierungsgraph für w_2

Jetzt muß noch je ein Element aus jeder der folgenden Zeilen zur Relation $\lessdot_2$ hinzugenommen werden.

- (a_1, a_0) oder (a_3, a_1), denn a_1 schreibt x und a_3 liest x von a_0.
- (a_2, a_0) oder (a_1, a_2), denn a_2 schreibt y und a_1 liest y von a_0.
- (a_0, a_3) oder (a_2, a_0), denn a_0 schreibt z und a_2 liest z von a_3.
- (a_0, a_1) oder (a_4, a_0), denn a_0 schreibt x und a_4 liest x von a_1.
- (a_0, a_2) oder (a_4, a_0), denn a_0 schreibt y und a_4 liest y von a_2.
- (a_0, a_3) oder (a_4, a_0), denn a_0 schreibt z und a_4 liest z von a_3.

(a_1, a_0), (a_2, a_0) und (a_4, a_0) scheiden von vornherein aus, da sich Zyklen ergeben. Also ergibt sich der Serialisierungsgraph aus Abb. 16. Dies ergibt $w_2' = l_0 s_0 l_3 s_3 l_1 s_1 l_2 s_2 l_4 s_4$ als einzige Serialisierung von w_2.

c) Wir setzen als Anfangszustand $(x, y, z) = (1, 1, 1)$. Wir untersuchen alle seriellen Ausführungsfolgen. (Aus Gründen der Übersichtlichkeit geben wir die Ausführungsfolgen des vergröberten Auftragssystems an.)

- $res(a_0 a_1 a_2 a_3 a_4, I) = (2, 2, 3)$.
- $res(a_0 a_1 a_3 a_2 a_4, I) = (2, 4, 3)$.
- $res(a_0 a_2 a_1 a_3 a_4, I) = (3, 2, 4)$.
- $res(a_0 a_2 a_3 a_1 a_4, I) = (3, 2, 2)$.
- $res(a_0 a_3 a_1 a_2 a_4, I) = (2, 3, 2)$.
- $res(a_0 a_3 a_2 a_1 a_4, I) = (4, 3, 2)$.

Nun gilt aber $res(w_1, I) = (2, 2, 2)$, was durch keine serielle Ausführungsfolge erreichbar ist. Damit ist bereits durch diese Interpretation bewiesen, daß w_1 nicht serialisierbar ist.

Aufgabe 13

Gegeben ist ein vollständiges schematisches Auftragssystem A mit Variablen x, y, z und Aufträgen

$$l_0[],\ s_0[x, y, z],\ l_1[y],\ s_1[x],\ l_2[x, z],\ s_2[y],\ l_3[],\ s_3[z],\ l_4[x, y, z],\ s_4[]$$

und mit genau den Präzedenzen

$$a_0 \lessdot a_1, a_2, a_3, a_4 \text{ und } a_0, a_1, a_2, a_3 \lessdot a_4.$$

Offenbar sind

$$w_1 = l_0[]\ s_0[x,y,z]\ l_1[y]\ l_2[x,z]\ l_3[]\ s_3[z]\ s_2[y]\ s_1[x]\ l_4[x,y,z]\ s_4[],$$
$$w_2 = l_0[]\ s_0[x,y,z]\ l_1[y]\ s_1[x]\ l_2[x,z]\ s_2[y]\ l_3[]\ s_3[z]\ l_4[x,y,z]\ s_4[],$$
$$w_3 = l_0[]\ s_0[x,y,z]\ l_3[]\ l_2[x,z]\ s_3[z]\ l_1[y]\ s_1[x]\ s_2[y]\ l_4[x,y,z]\ s_4[]$$

verschiedene Ausführungsfolgen von A.

a) Gibt es Interpretationen für die die Ausführungsfolgen w_1 und w_2 bzw. w_1 und w_3 gleiche bzw. verschiedene Resultate liefern?

b) Man bestimme die Wertübertragungsrelation für w_1, w_2 und w_3. Welche Aufträge sind jeweils relevant?

c) Legen Sie Funktions- und Konstantensymbole für die Herbrand-Interpretation des Auftragssystems A fest und definieren Sie die Menge der Terme.

d) Geben Sie die Herbrand-Interpretation des Auftragssystems A an und berechnen Sie für w_1, w_2 und w_3 die Resultate unter dieser Interpretation. In welchem Sinne ist die Herbrand-Interpretation allgemeiner als diejenigen in Teilaufgabe (a)?

e) Man verwende die Herbrand-Interpretationen um die Äquivalenz bzw. Nichtäquivalenz von w_1, w_2 und w_3 nachzuweisen (vgl. Satz 2.3.4.12).

Lösung

a) Als Wertemenge für die folgenden Interpretationen wählen wir $\mathbb{N}$. Den Zustand während der Ausführung beschreiben wir entsprechend den Werten für x, y und z durch ein Tripel.

Zur Vorbereitung definieren wir Funktionen $f : \mathbb{N} \to \mathbb{N}$, $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ und Konstanten $a, b, c, d \in \mathbb{N}$ durch $f(i) = i + 2$, $g(i, j) = i + j$ und $a = 1$, $b = 2$, $c = 3$, $d = 17$.

Die Interpretation I definieren wir wie folgt:
Auftrag a_0 schreibt a, b, c in x, y, z.
Auftrag a_1 liest y und schreibt $f(y)$ in x.
Auftrag a_2 liest x, z und schreibt $g(x, z)$ in y.
Auftrag a_3 schreibt d in z.
Auftrag a_4 liest x, y, z.

Als Resultate erhalten wir
$res(w_1, I) = (f(b), g(a, c), d) = (4, 4, 17)$,
$res(w_2, I) = (f(b), g(f(b), c), d) = (4, 7, 17)$ und
$res(w_3, I) = (f(b), g(a, c), d) = (4, 4, 17)$.

Daß sich für w_1 und w_3 das gleiche Resultat ergibt ist unvermeidlich wie in der letzten Teilaufgabe gezeigt wird.

Eine andere Interpretation I' läßt sich so definieren, daß alle Funktionen und Konstanten der Interpretation gleich 0 sind. In diesem Fall sind natürlich die Resultate von w_1, w_2 und w_3 gleich $(0, 0, 0)$. Unter dieser Interpretation sind also w_1, w_2 und w_3 ununterscheidbar.

b) Werteübertragungsrelation für w_1:
a_1 liest y von a_0,
a_2 liest x und z von a_0,
a_4 liest x von a_1,
a_4 liest y von a_2,
a_4 liest z von a_3.

Werteübertragungsrelation für w_1:
a_1 liest y von a_0,
a_2 liest x von a_1,

a_2 liest z von a_0,
a_4 liest x von a_1,
a_4 liest y von a_2,
a_4 liest z von a_3.

Werteübertragungsrelation für w_3: wie w_1.

Formal enthalten die Werteübertragunsrelationen natürlich nur die Paare der Aufräge, z.B. (a_0, a_1), etc. für w_1, aber nicht die Variablen.

Alle Aufträge sind also relevant für w_1, w_2 und w_3.

c) Als Funktionssymbole wählen wir f (einstellig), g (zweistellig) und a, b, c, d (nullstellig, d.h. Konstantensymbole). Daß wir die gleichen Namen wie oben verwendet haben ist reiner Zufall. Die Terme sind induktiv definiert: a, b, c, d sind Terme und $f(M)$, $g(M, N)$ sind Terme für Terme M, N. Beispiele für Terme sind $a, f(d), g(a, b), g(f(a), f(b))$.

d) Die Herbrand-Interpretation I_H definieren wir wie folgt:
Auftrag a_0 schreibt a, b, c in x, y, z.
Auftrag a_1 liest x und schreibt $f(x)$ in y.
Auftrag a_2 liest x, z und schreibt $g(x, z)$ in y.
Auftrag a_3 schreibt d in z.
Auftrag a_4 liest x, y, z.

Als Resultate erhalten wir:
$res(I_H, w_1) = (f(b), g(a, c), d)$,
$res(I_H, w_2) = (f(b), g(f(b), c), d)$ bzw.
$res(I_H, w_3) = (f(b), g(a, c), d)$.

Die Resultate für w_1 und w_3 sind gleich, was nicht verwunderlich ist, da w_1 und w_3 die gleiche liest-von-Relation haben.

Die Herbrand-Interpretation ist allgemeiner als die oben angegebenen Interpretationen I und I', da sich I und I' aus I_H durch Zuweisung spezieller Bedeutungen für die Symbole a, b, c, d, f, g ergeben.

e) Die Herbrand-Interpretation I_H ist eine spezielle Interpretation mit Termen als Wertebereich. Also zeigt die Verschiedenheit von $res(I_H, w_1)$ und $res(I_H, w_2)$ die Nichtäquivalenz von w_1 und w_2. Entsprechendes gilt für w_2 und w_3. Da die Herbrand-Interpretation die allgemeinste Interpretation ist, zeigt die Gleichheit von $res(I_H, w_1)$ und $res(I_H, w_3)$ die Äquivalenz von w_1 und w_3.

Aufgabe 14

Ein Computer erhält folgendes Stück eines Programmes:

```
x := a+b;
con z := x*a || r := p/z || u := r*b || v := b/x noc;
u := a*a;
s := z/r;
u := x/a;
```

a) Bezeichnen Sie die Zuweisungen in der angegebenen Reihenfolge der Aufschreibung als Aufträge a_1 bis a_8 und konstruieren Sie das schematische Auftragssystem AS entsprechend der Semantik von PROG.

b) Entscheiden Sie mit Satz 2.3.5.4, ob AS funktional bzw. spurfunktional ist. Falls nicht, solle eine minimale Anzahl von Präzedenzen eingefügt werden, um dies zu erreichen.

c) Konstruieren Sie daraus das für Funktionalität bzw. Spurfunktionalität maximal nebenläufige Auftragssystem.

Lösung

a) In Abb. 17 ist das resultierende schematische Auftragssystem dargestellt.

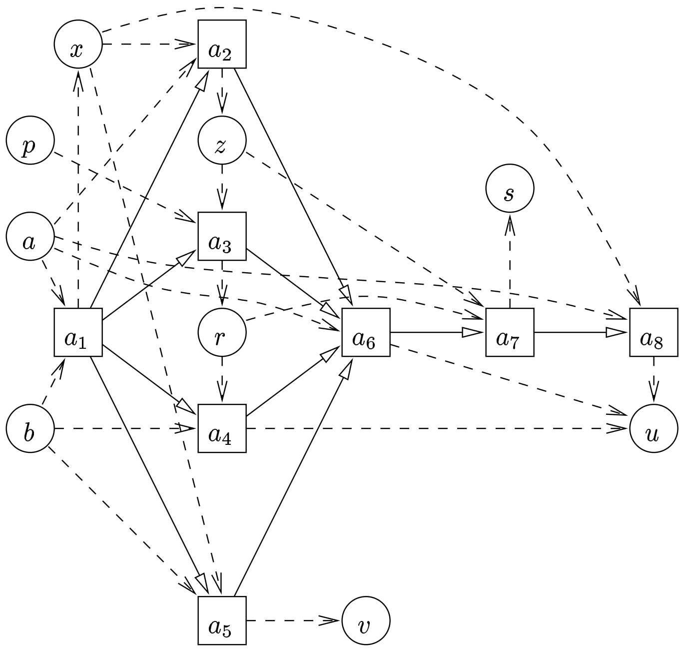


Abbildung 17: Schematisches Auftragssystem

b) Alle Aufträge sind verlustfrei. Die Aufträge a_1, a_2, a_3, a_5, a_7 und a_8 sind relevant. Die Aufträge a_4 und a_6 sin nicht relevant.

In der folgenden Tabelle werden alle Präzedenzen ($\lessdot$ bzw. $\gtrdot$) und alle nicht disjunkten Paare ($\neg dis(a_i, a_j)$, notiert als !) verzeichnet.

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
a_1		$!\lessdot$	$\lessdot$	$\lessdot$	$!\lessdot$	$\lessdot$	$\lessdot$	$!\lessdot$
a_2	$!\gtrdot$		!			$\lessdot$	$!\lessdot$	$\lessdot$
a_3	$\gtrdot$	!		!		$\lessdot$	$!\lessdot$	$\lessdot$
a_4	$\gtrdot$		!			$!\lessdot$	$\lessdot$	$!\lessdot$
a_5	$!\gtrdot$					$\lessdot$	$\lessdot$	$\lessdot$
a_6	$\gtrdot$	$\gtrdot$	$\gtrdot$	$!\gtrdot$	$\gtrdot$		$\lessdot$	$!\lessdot$
a_7	$\gtrdot$	$!\gtrdot$	$!\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$		$\lessdot$
a_8	$!\gtrdot$	$\gtrdot$	$\gtrdot$	$!\gtrdot$	$\gtrdot$	$!\gtrdot$	$\gtrdot$	

Um Funktionalität zu erreichen, müßte $a_2 \lessdot a_3$ oder $a_3 \lessdot a_2$ hinzugefügt werden. Wir wählen $a_2 \lessdot a_3$. Um Spurfunktionalität zu erreichen, müßte $a_3 \lessdot a_4$ oder $a_4 \lessdot a_3$ hinzugefügt werden. Wir wählen $a_3 \lessdot a_4$.

c) In Abb. 18 und 19 sind die Auftragssysteme angegeben, die maximal nebenläufig für Funktionalität bzw. Spurfunktionalität sind.

Aufgabe 15 Invarianten in S/T-Netzen

Gegeben seien die S/T-Netze N und N', wobei N das Gesamtnetz aus der Abb. 20 ist und N' das Teilnetz ist, dessen Elemente mit durchgezogenen Linien gezeichnet sind.

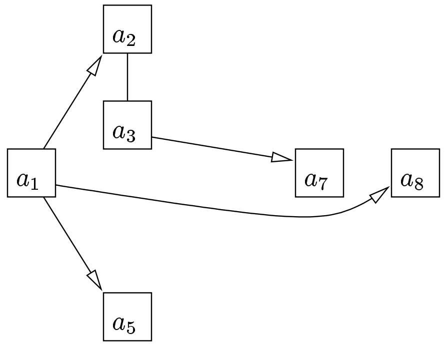


Abbildung 18: Maximal für Funktionalität

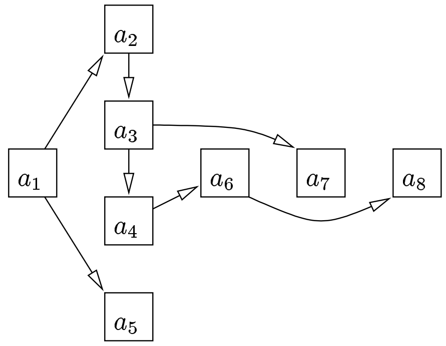


Abbildung 19: Maximal für Spurfunktionalität

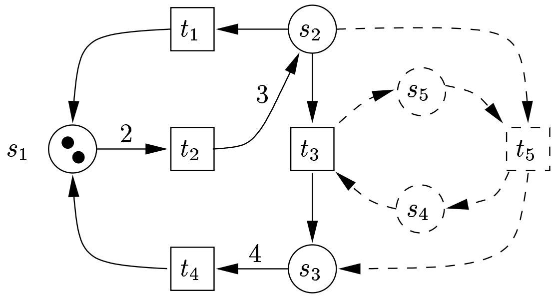


Abbildung 20: Netz und Teilnetz

Berechnen Sie mit Hilfe der Sätze 2.4.2.3 (Satz von Lautenbach) und 2.4.2.6 S-Invarianten und T-Invarianten für das Gesamtnetz N und das Teilnetz N' (falls diese existieren). Untersuchen Sie, ob es in den Netzen Schaltfolgen gibt, die die Anfangsmarkierung wieder herstellen.

Lösung

Gesamtnetz

Die Inzidenzmatrix für das Gesamtnetz N ist

$$\Delta_N = \begin{pmatrix} 1 & -2 & 0 & 1 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & 0 & 1 & -4 & 1 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 \end{pmatrix}$$

Damit ergeben sich die T-Invariantenvektoren $(5x, 3x, 2x, x, 2x)^T$ für beliebiges $x > 0$. S-Invariantenvektoren sind $(0, 0, 0, x, x)^T$ für beliebiges $x \neq 0$.

Teilnetz

Die Inzidenzmatrix für das Teilnetz N' ist

$$\Delta_N = \begin{pmatrix} 1 & -2 & 0 & 1 \\ -1 & 3 & -1 & 0 \\ 0 & 0 & 1 & -4 \end{pmatrix}$$

Damit haben wir $(5x, 3x, 4x, x, 2x)^T$ als T-Invariantenvektoren für beliebiges $x > 0$. Es gibt keine S-Invariantenvektoren.

Aufgabe 16

Es sollen Invarianten in Petrinetzen gefunden und Systemeigenschaften bewiesen werden – soweit möglich mit Hilfe der gefundenen Invarianten. Sei N das Petrinetz in der Abbildung 2.4.4.5[1], wobei die gestrichelten, eingefaßten Gebiete eine Transition sind, sei N' das verfeinerte Netz (ohne Zusammenfassen der eingefaßten Transitionen). Finden Sie für beide Netze die Invarianten und geben Sie ihnen inhaltliche Bedeutungen (z.B. „Die Zahl der Philosophen ist konstant“). Beweise für beide Netze:

a) Wenn ein Philosoph i ißt, sind die Stellen g_i und g_{i+1} unmarkiert.

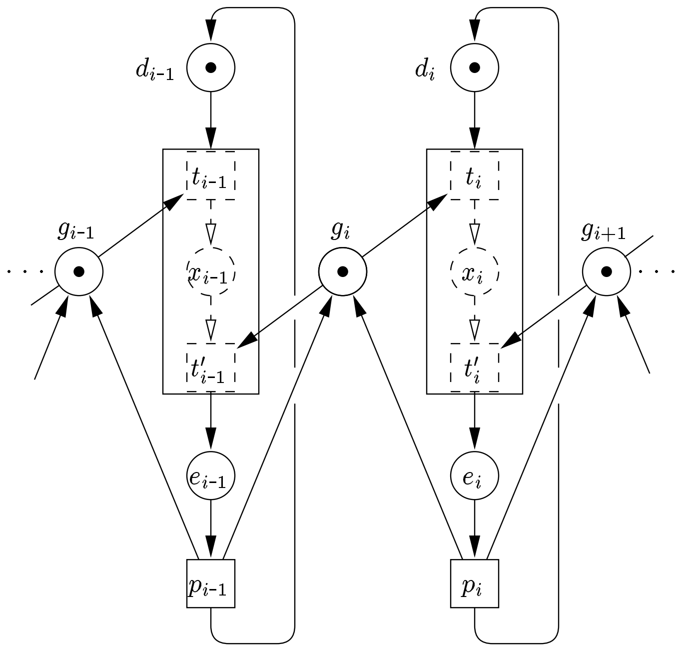
b) Nie essen zwei Nachbarn gleichzeitig.

c) Auf jeder Stelle liegt jeweils höchstens eine Marke.

d) Die Netze sind verklemmungsfrei – oder geben Sie eine Verklemmung an.

Zusatzaufgabe (setzt Abschnitt 2.4.4 voraus):

e) Das Netz N erlaubt verschleppungsfreie (faire) Schaltfolgen, in denen nicht alle Philosophen essen. Geben Sie eine Folge nach der verschleppungsfreien und eine nach der fairen Schaltregel hierfür an.



Lösung

Es lassen sich folgende Invarianten finden: (e_i ist eine Abkürzung für $m(e_i)$)

	Netz N	Netz N'	
1)	$d_i + e_i = 1$	$d_i + x_i + e_i = 1$	für alle $1 \leq i \leq 5$
2)	$e_{i-1} + e_i + g_i = 1$	$e_{i-1} + e_i + x_i + g_i = 1$	für alle $1 \leq i \leq 5$

[1]Die Nummern der Abbildungen, Sätze und Abschnitte beziehen sich auf *Jessen/Valk: Rechensysteme.*

Die Korrektheit der Gleichungen unter 1) und 2) läßt sich leicht überprüfen, indem man zeigt, daß das Schalten keiner (relevanten) Transition die Gleichungen beeinflußt. Für N muß man die Transitionen t_{i-1}, p_{i-1}, t_i und p_i betrachten, für N' die Transitionen t'_{i-1}, p_{i-1}, t_i, t'_i und p_i. Natürlich sind auch alle Linearkombinationen dieser 10 Invarianten wieder Invarianten.

Inhaltliche Bedeutung der Invarianten unter 1) und 2)

1) Philosoph i befindet sich genau in einem der Zustände „Denken“, „Essen“ oder (nur für N') „x_i“ („hat die rechte Gabel genommen“).

2) Gabel i kann auf einer der Stellen g_i, e_i, e_{i-1} oder x_i lokalisiert werden.

Die Beweise für beide Netze

a) Wenn ein Philosoph i ißt sind die Stellen g_i und g_{i+1} unmarkiert. $e_{i-1}+2\cdot e_i+e_{i+1}+g_i+g_{i+1} = 2$ für N (bzw. $e_{i-1}+2\cdot e_i+e_{i+1}+x_i+x_{i+1}+g_i+g_{i+1} = 2$ für N') ist eine Invariante. Falls also $e_i = 1$, d.h. markiert, folgt $g_i = 0$ und $g_{i+1} = 0$.

b) Nie essen zwei Nachbarn gleichzeitig. Dies ist direkt aus den Invarianten $e_{i-1}+e_i+g_i = 1$ (bzw. $e_{i-1}+e_i+x_i+g_i = 1$) zu sehen.

c) Auf jeder Stelle liegt jeweils höchstens eine Marke. Die zehn Invarianten überdecken alle Stellen des Netzes und jede stellt sicher, daß nicht mehr als eine Marke auf den Stellen liegt, die in ihr vorkommen.

d) N' besitzt eine Verklemmung: $t_1t_2t_3t_4t_5$ führt das Netz in eine Verklemmung – keine Transition kann mehr schalten.

   N ist verklemmungsfrei, sogar lebendig: Da das Netz für alle i symmetrisch ist, muß nur gezeigt werden, daß t_i und p_i nicht verklemmt werden können. Mit der Invariante 1) gilt: Entweder auf e_i oder auf d_i liegt eine Marke. Wir zeigen: Wenn e_i markiert ist, kann p_i schalten und d_i wird markiert, und wenn d_i markiert ist, gibt es eine Folge, so daß t_i schalten kann und e_i markiert wird.
   Fallunterscheidung:

   1. $e_i = 1$: p_i kann schalten und markiert g_i, g_{i+1} und d_i.
   2. $d_i = 1$: Es muß eine weitere Fallunterscheidung gemacht werden:

      2.1 g_i und g_{i+1} sind markiert: t_i kann schalten und markiert e_i.

      2.2 g_i oder g_{i+1} oder beide sind nicht markiert. Falls g_i nicht markiert ist, dann ist e_{i-1} markiert (nach Invariante 2)) und damit kann p_{i-1} schalten (siehe oben) und g_i markieren. Falls g_{i+1} unmarkiert ist, gilt das gleiche für e_{i+1} und p_{i+1}. Also gibt es immer eine Folge von 1 oder 2 Transitionen, nach deren Schalten g_i und g_{i+1} markiert sind. Weiter unter Fall 2.1. Beide Transitionen t_i und p_i können also aus jeder Situation heraus wieder schalten – es müssen höchstens eine, zwei oder drei der Transitionen t_i, p_i, p_{i-1} und p_{i+1} zuvor schalten.

e) Das Netz N erlaubt verschleppungsfreie (faire) Schaltfolgen, in denen nicht alle Philosophen essen – gib eine Folge nach der verschleppungsfreien und eine nach der fairen Schaltregel hierfür an:

   - Verschleppungsfrei: $(t_1p_1t_3p_3)^\omega$,
     denn t_2 und t_4 sind immer wieder *nicht* aktiviert (nach t_1 und nach t_3), müssen daher auch nicht vorkommen.
   - Fair: $t_1(t_4p_4t_3p_1t_5p_5t_1p_3)^\omega$,
     denn t_2 kommt nicht vor, ist aber auch nur einmal am Anfang aktiviert.

Aufgabe 17

Für das Netz N_1 aus Abbildung 21 sollen einige Eigenschaften entschieden werden, die durch die untenstehende Definition gegeben sind.

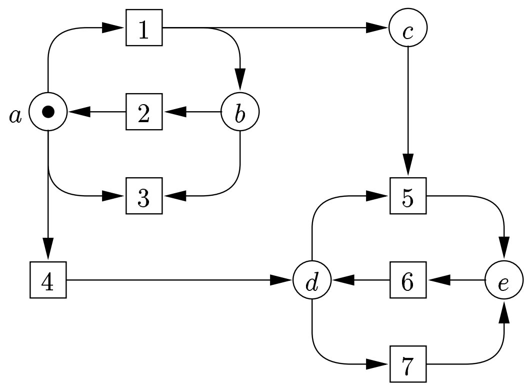


Abbildung 21: Netz N_1

a) Zeichnen Sie zunächst den Markierungsgraphen. Falls der Markierungsgraph unendlich ist, versuchen Sie ihn zu parametrisieren.

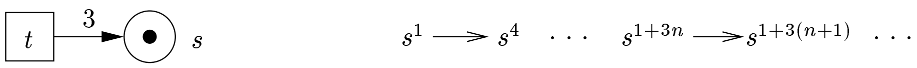


Abbildung 22: Beispielnetz mit parametrisiertem Markierungsgraph

b) Welche Transitionen sind tot?

c) Welche Transitionen sind aktivierbar?

d) Welche Transitionen sind unbeschränkt aktivierbar?

e) Welche Transitionen sind ω-aktivierbar?

f) Gibt es eine Schaltfolge, in der alle ω-aktivierbaren Transitionen unendlich oft vorkommen? Wenn ja, nennen Sie ein Beispiel.

g) Welche Transitionen sind lebendig?

h) Ist das Netz N_1 lebendig?

i) Ist das Netz N_1 verklemmungsfrei?

Definition: Sei $N = (S, T; F, W, m_0)$ ein S/T-Netz ohne Kapazitäten und t eine Transition. t heißt (jeweils in der Anfangsmarkierung m_0):

- *tot*, falls t in keiner Markierung $m \in R(N)$ aktiviert ist.
- *aktivierbar*, falls t nicht tot ist, d.h. $\exists m \in R(N) : m \xrightarrow{t}$.
- *unbeschränkt aktivierbar* (oder *warm*), falls $\forall n \in I\!N : \exists w \in T^* : m_0 \xrightarrow{w} \wedge |w|_t \geq n$.
  (Dabei gibt $|w|_t$ an, wie oft t in w vorkommt, d.h. $|w|_t = |\{i \in \{1, ..., |w|\} | w(i) = t\}|$.)

- *ω-aktivierbar* (oder *heiß*), falls es eine unendliche Folge $w \in T^\omega$ über T gibt, mit $m_0 \xrightarrow{w} \wedge\ |w|_t = \omega$.

  (Dabei ist T^ω die Menge aller unendlichen Folgen über T. Für $w \in T^\omega$ gilt $|w| = \omega$ und $m \xrightarrow{w}$, falls $\forall i \in I\!N : m \xrightarrow{w(1)\ldots w(i)}$.)

- *lebendig*, falls t von jeder Markierung $m \in R(N)$ aus aktivierbar ist, d.h. $\forall m \in R(N) : \exists w \in T^* : m \xrightarrow{wt}$.

N heißt:

- *lebendig*, falls alle Transitionen lebendig sind.
- *verklemmungsfrei*, falls in jeder Markierung $m \in R(N)$ eine Transition aktiv ist, d.h. $\forall m \in R(N) : \exists t \in T : m \xrightarrow{t}$.

Lösung

a) Da der Markierungsgraph unendlich ist, können wir ihn nur durch Parametrisierung exakt angeben.

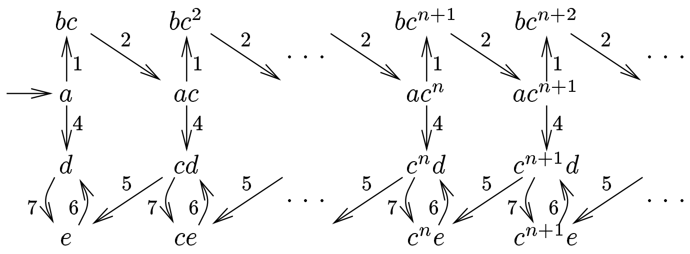


Abbildung 23: Der parametrisierte Markierungsgraph

b) Transition 3 ist tot.

c) Transitionen 1, 2, 4, 5, 6, 7 sind aktivierbar.

d) Transitionen 1, 2, 5, 6, 7 sind unbeschränkt aktivierbar.

e) Transitionen 1, 2, 6, 7 sind jeweils ω-aktivierbar

f) Es gibt keine unendliche Folge $w \in T^\omega$ mit m_0 aktiviert w und

$$|w|_1 = |w|_2 = |w|_6 = |w|_7 = \omega.$$

g) Transitionen 6, 7 sind lebendig.

h) N_1 ist nicht lebendig.

i) N_1 ist verklemmungsfrei.

Aufgabe 18

In Abbildung 24 ist qualitativ der Lebenszyklus eines Regenwurms dargestellt. Der Zyklus beginnt mit der Geburt eines neuen Regenwurms, die natürlich nur mit zwei erwachsenen Regenwürmern als Nebenbedingung stattfinden kann. (Ähnliche Situationen können überall dort auftreten, wo eine minimale Anzahl an Resourcen vorhanden sein muß, um weitere Resourcen zu erzeugen.)

Die Stellen des Netzes seien mit J und A bezeichnet, für die Transitionen wählen wir die Namen G, E und S.

a) Bestimmen Sie das Residuum $res(K)$ der T-fortsetzbaren Markierungen K. Benutzen Sie dazu eine beliebige Methode, die Ihnen geeignet erscheint.

b) Generieren Sie mit Hilfe von Definition 2.4.3.4 aus *Jessen/Valk: Rechensysteme* das K-gesteuerte Netz, um den Regenwürmern eine dauerhafte Erhaltung ihrer Art zu sichern.

Lösung

Wir konstruieren zunächst einen Teil des Markierungsgraphen, das Ergebnis ist in Abbildung 25 zu sehen.

Wir erkennen unmittelbar, daß die Markierung $(0, 2)$ T-fortsetzbar ist, da in ihr die unendliche Folge $GESGESGES\ldots$ schalten kann. Gleiches gilt für die Markierungen $(1, 1)$ und $(2, 0)$, da von ihnen aus die Markierung $(0, 2)$ erreichbar ist. Da die Eigenschaft der T-Fortsetzbarkeit bei Hinzufügen von Marken erhalten bleibt, sind auch alle Markierungen in $K = \{(0, 2), (1, 1), (2, 0)\} + \mathbb{N}^2 = \{(a, b) | a, b \in \mathbb{N} \wedge a + b \geq 2\}$ T-fortsetzbar.

Die einzige Markierungen, die nicht in dieser Menge enthalten sind, sind $(1, 0)$, $(0, 1)$ und $(0, 0)$. Sie sind jedoch auch nicht T-fortsetzbar, da nur die Schaltfolgen ES, S bzw. die leere Schaltfolge möglich sind und wir danach eine tote Markierung erreichen. Also haben wir mit K genau die Menge der T-fortsetzbaren Markierungen bestimmt.

Da $\{(0, 2), (1, 1), (2, 0)\}$ auch eine Menge unvergleichbarer Markierungen ist, ist $res(K) = \{(0, 2), (1, 1), (2, 0)\}$ das Residuum der T-fortsetzbaren Markierungen.

Wir können nun das K-gesteuerte Netz konstruieren. Dazu bestimmen wir zunächst die Mengen T_1 und T_2, diese ergeben sich nach der Definition im Buch *„Rechensysteme“* zu $T_1 = \{G, E\}$ und $T_2 = \{S_{(0,2)}, S_{(1,1)}, S_{(2,0)}\}$. Das resultierende Netz ist in Abbildung 26 dargestellt.

Aufgabe 19

a) Kann man generell für ein Wahrscheinlichkeitsmaß P beweisen: Ein Ereignis E mit $P[E] = 0$ kann bei Durchführung eines Zufallsexperimentes nicht auftreten?

b) Man definiere das Wahrscheinlichkeitsmaß für das Experiment „Einmal würfeln“.

Lösung

a) Ein Ereignis E mir $P[E] = 0$ kann durchaus auftreten. Wir beweisen dies mit einem Beispiel.

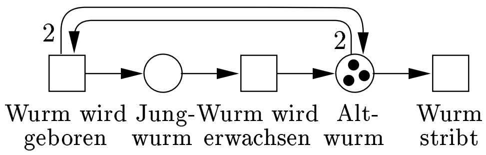


Abbildung 24: Leben und Sterben der Regenwürmer

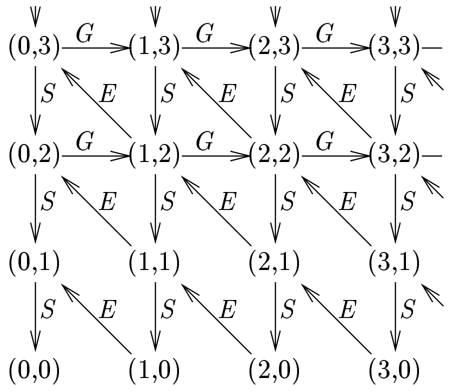


Abbildung 25: Der Markierungsgraph des Regenwurmnetzes, $(m(J), m(A))$

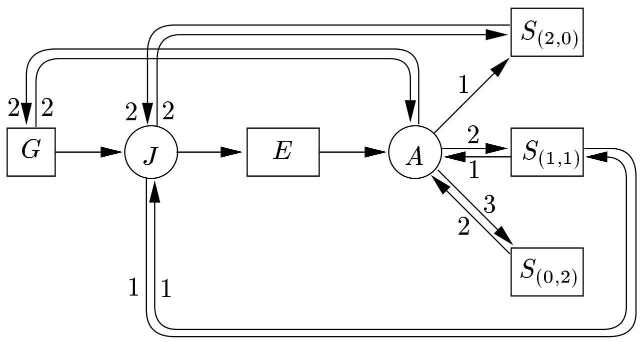


Abbildung 26: K-gesteuertes Regenwurmnetz

Sei R die Menge der reellen Zahlen, x eine kontinuierlich verteilte ZVA mit WSK-Verteilung F und Dichtefunktion f.

Sei nun $a \in R$ mit $f(a) > 0$, dann kann die ZVA den Wert $x = a$ annehmen.

Behauptung: $P[x = a] = 0$

Beweis: $\forall n : P[x = a] \leq P[a - \frac{1}{n} \leq x \leq a + \frac{1}{n}]$, da für alle n gilt: $a \in [a - \frac{1}{n}, a + \frac{1}{n}]$.
Somit gilt: $\forall n : P[x = a] \leq F(a + \frac{1}{n}) - F(a - \frac{1}{n})$ und daher auch

$$0 \leq P[x = a] \leq \lim_{n\to\infty} (F(a + \frac{1}{n}) - F(a - \frac{1}{n})) = 0$$

Also:

$$P[x = a] = 0.$$

Ein Beispiel zur Veranschaulichung ist die Messung der Körpergröße x. Dann ist $P[x = 180\text{cm}] = 0$ im *kontinuierlichen Modell.* Aber: Wegen der Grenzen der Meßgenauigkeit wäre ein *diskretes Modell* duchaus sinnvoll (z.B. Messungen nur auf cm genau). Dann gilt: $P[x = 180\text{cm}] > 0$.

b) Wir modellieren den Ereignisraum für einmaliges Würfeln mit

$$\Omega = \{1, 2, 3, 4, 5, 6\}.$$

Für ein beliebiges Ereignis $A \subseteq \Omega$ ist (bei einem gerechten Würfel)

$$P[A] = \frac{|A|}{6},$$

für einen fehlerhaften Würfel müßten wir die Wahrscheinlichkeiten anpassen.

Aufgabe 20

Ein Würfelspiel hat die folgende Regel: Es darf eine beliebige Anzahl mal mit einem Würfel gewürfelt werden. Falls eine *Eins* bei einem der Würfe vorkommt, kann nichts angeschrieben werden, sonst die Gesamtzahl der Augen. Gewonnen hat, wer am Schluß (nach vielen Runden) die meisten Augen (größte Augensumme) angeschrieben hat. Man gebe für zwei Spielalternativen mit Hilfe der Wahrscheinlichkeitsrechnung Strategien an.

a) Es muß jeweils am Beginn einer Runde die Anzahl der Würfe festgelegt werden, die man machen will.

b) Es darf nach jedem Wurf entschieden werden, ob man aufhören oder noch einen Wurf riskieren möchte.

Lösung

a) Wir müssen die Anzahl der Würfe vorher festlegen.

Wir berechnen den erwarteten Gewinn $G(n)$ bei n Versuchen:

- Die WSK, daß keine 1 in n Versuchen: $\left(\frac{5}{6}\right)^n$, wenn Unabhängigkeit vorausgesetzt werden kann. (Im allg. hinreichend realistische Voraussetzung.)
- Die erwartete Augenzahl pro erfolgreichem Wurf:

$$\frac{1}{5} \cdot \sum_{i=2}^{6} i = \frac{20}{5} = 4$$

(nicht 3.5, da hier 1 kein erfolgreicher Versuch).
- Die erwartete Augenzahl bei n Würfen ist also $n \cdot 4$.

Also ist

$$G(n) = \left(\frac{5}{6}\right)^n \cdot n \cdot 4$$

Für welches x wird $G(x)$ maximal?

Mit der Produktregel und $(a^x)' = ln(a) \cdot a^x$ erhält man

$$\frac{dG(x)}{dx} = \ln\left(\frac{5}{6}\right)\left(\frac{5}{6}\right)^x \cdot x \cdot 4 + 4 \cdot \left(\frac{5}{6}\right)^x$$

Also:

$$\frac{dG(x)}{dx} = 0 \iff \ln\left(\frac{5}{6}\right)\left(\frac{5}{6}\right)^x \cdot x \cdot 4 = -4 \cdot \left(\frac{5}{6}\right)^x$$

$$x = -\frac{1}{\ln(\frac{5}{6})} = -\frac{1}{\ln(0.8\overline{3})} \approx 5.4848\ldots$$

Dieses Maximum in $I\!R$ ist jedoch nicht zulässig, deshalb muß das Maximum in der Menge $\{5, 6\}$ gesucht werden. Ausrechnen zeigt, daß $G(5) = G(6)$, so daß die beste Strategie für den ersten Teil der Aufgabe ist, 5 oder 6 Versuche auszuführen (gleich gute Entscheidungen).

Alternative zur formalisierten Maximumsuche: Berechnung der Werte $G(0), \ldots, G(7)$, insbesondere $G(0) = 0$, $G(1) = 3\frac{1}{3}$, $G(2) = 5\frac{5}{9}, G(3) = 6.9\overline{4}$, $G(4) \approx 7.716\ldots, G(5) \approx 8.03755\ldots$, $G(6) = G(5)$, $G(7) \approx 7.814\ldots$. Da G für $n \geq 7$ streng monoton fallend ist, liegen die Maxima bei 5 und 6.

b) Wir müssen uns entscheiden, ob wir weiter würfeln.

Die aktuelle Augensumme zum Zeitpunkt der Entscheidung sei n_0. Was passiert bei erneutem Würfeln?

- erwarteter Gewinn (Zuwachs): $\left(\frac{5}{6}\right) \cdot 4 = \frac{20}{6}$, unabhängig vom aktuellen Stand.
- erwarteter Verlust: $n_0 \cdot \frac{1}{6}$

Also bleibt die Teilfrage zu lösen, bei welchen n_s der zu erwartende Verlust den zu erwartenden Gewinn übersteigt, d.h.

$$n_s = \min_{n \in I\!N}\{n \cdot \frac{1}{6} > \frac{5}{6} \cdot 4\} = \min_{n \in I\!N}\{n > 20\} = 21$$

Daraus folgt für unsere Strategie

- bei $n_0 < 20$ weiter würfeln,
- bei $n_0 = 20$ weiter würfeln *oder* Stop, denn der erwartete Gewinn ist identisch mit dem erwarteten Verlust,
- bei $n_0 > 20$ aufhören.

Zusatzfragen

- Weshalb erhalten wir in Teil 1. und 2. unterschiedliche Resultate? Wir haben bei 2. zusätzliches Wissen zum Zeitpunkt der Entscheidung.
- Interessiert bei 2., in wievielen Versuchen n_0 Punkte erzielt werden? Nein, der Würfel ist gedächtnislos.

Aufgabe 21

In einem der klassischen Versuche von Rutherford und Geiger über die Emission von α-Teilchen wurde die Anzahl der in $n = 2608$ Zeitabschnitten (von 7,5 Sekunden) emittierten Teilchen registriert. Es wurde notiert, in wieviel (n_ν) Zeitabschnitten ν Teilchen ($\nu = 0, 1, 2, \ldots, 10$) anfielen:

ν	0	1	2	3	4	5	6	7	8	9	10
n_ν	57	203	383	525	532	408	273	139	45	27	16

Man versuche, dieses Ergebnis als eine Poisson-Verteilung zu deuten, der Erwartungswert λ ergibt sich aus $\lambda = \sum_{\nu=0}^{10} \nu \cdot n_\nu \cdot \frac{1}{n}$. Dazu vergleiche man die empirisch bestimmten Zahlen n_ν mit den zu berechnenden $n \cdot p_\nu = n \cdot \frac{\lambda^\nu}{\nu!} \cdot e^{-\lambda}$.

Lösung

Wir stellen die bekannten Daten zusammen und berechnen die gewünschten zusätzlichen Werte in Tab. 1.

Die Zahlen in der Tabelle wurden nach 4 Stellen abgeschnitten, jedoch bei der Berechnung mit höherer Präzision behandelt. Schon die hier dargestellte Genauigkeit liegt deutlich über den zu erwartenden Fehlermargen. Die letzte Spalte zeigt die von einer Poissionverteilung zu erwartenden Häufigkeiten bei $\lambda = 3.867$. Die Übereinstimmung mit den gemessenen Daten in der zweiten Spalte ist annehmbar.

Die Summe von 2602.2 in der letzten Spalte ist darauf zurückzuführen, daß die Verteilung bei $\nu = 10$ abgeschnitten wurde.

Es wäre möglich, nun eine Fehlerrechnung zu betreiben oder einen stochastischen Test (z.B. den χ^2-Test) durchzuführen. Dies war jedoch nicht Teil der Aufgabe.

ν	n_ν	$\nu \cdot n_\nu \cdot \frac{1}{n}$	$n \cdot \frac{\lambda^\nu}{\nu!} \cdot e^{-\lambda}$
0	57	0.000	54.5
1	203	0.077	210.9
2	383	0.293	407.9
3	525	0.603	525.8
4	532	0.815	508.3
5	408	0.782	393.1
6	273	0.628	253.3
7	139	0.373	139.9
8	45	0.138	67.6
9	27	0.093	29.0
10	16	0.061	11.2
$\sum$	2608	3.867	2602.2

Tabelle 1: Berechnung zum Versuch von Rutherford

Aufgabe 22

a) Ein Programmsystem enthält einen Fehler. Während einer abschliessenden Testphase von 100 Stunden könnte dieser Fehler behoben werden, wenn er denn entdeckt würde. Der Fehler tritt mit einer Rate von λ auf. Wie wahrscheinlich ist es, daß der Fehler nicht während des Tests, sondern während der Garantiezeit von 1000 Betriebsstunden auftritt?

b) Welches sind in diesem Sinne die gefährlichsten Fehler?

c) Nehmen sie an, während des Tests wäre die Fehlerrate zehnfach überhöht, z.B. weil Grenzfälle getestet werden. Wie ändert sich die Situation? Bei 100-fach überhöhter Rate?

d) Ein sehr schwieriges Problem wurde von 3 Programmierern bearbeitet. Ihre Programme haben eine Fehlerwahrscheinlichkeit von 10^{-1}, 10^{-2} und 10^{-3}, weil sie nur die Vorlesung, Vorlesung und Übung, bzw. auch das Tutorium zur Rechensysteme-Vorlesung besuchten. Ist es sinnvoll, einen 2-aus-3-Voter einzusetzen, um die Wahrscheinlichkeit eines korrekten Ergebnisses zu erhöhen?

Lösung

a) Die Wahrscheinlichkeit, den Fehler während des Tests in einem Zeitraum x zu übersehen, ist $e^{-\lambda x}$. Die Wahrscheinlichkeit, den Fehler während des Betrieb über den Zeitraum y zu übersehen, ist $e^{-\lambda y}$. Die gesuchte Wahrscheinlichkeit ist damit $p = e^{-\lambda x}(1 - e^{-\lambda y})$.

   Für $x = 100h$ und $y = 1000h$ ergibt sich $p_1 = e^{-\lambda 100h}(1 - e^{-\lambda 1000h})$.

b) Wir suchen das Maximum in Abhängigkeit von λ. Wir bilden die Ableitung unter Verwendung von Produktregel und Kettenregel.

$$\begin{aligned}
p' &= (e^{-\lambda x}(1 - e^{-\lambda y}))' \\
&= (e^{-\lambda x})(1 - e^{-\lambda y})' + (e^{-\lambda x})'(1 - e^{-\lambda y}) \\
&= (e^{-\lambda x})(-y)(-e^{-\lambda y}) + (-x)(e^{-\lambda x})(1 - e^{-\lambda y}) \\
&= (x + y)e^{-\lambda x}e^{-\lambda y} - xe^{-\lambda x}
\end{aligned}$$

Wir setzen gleich 0, um das Maximum zu bestimmen.

$$\begin{aligned} p' &= 0 \\ (x+y)e^{-\lambda x}e^{-\lambda y} &= xe^{-\lambda x} \\ (x+y)e^{-\lambda y} &= x \\ e^{-\lambda y} &= \tfrac{x}{x+y} \\ -\lambda y &= \ln \tfrac{x}{x+y} \\ \lambda &= \tfrac{1}{y} \ln \tfrac{x+y}{x} \end{aligned}$$

Für $x = 100h$ und $y = 1000h$ ergibt sich das Fehlermaximum bei $\lambda = 2{,}39 \cdot 10^{-3}h^{-1}$. Die Wahrscheinlichkeit, daß der Fehler im Betrieb auftritt, ist $p = 0.715$.

c) Eine 10-fach überhöhte Rate wirkt genau wie eine 10-fach verlängerte Testzeit. Für $x = 1000h$ und $y = 1000h$ ergibt sich nun das Fehlermaximum bei $\lambda = 6{,}93 \cdot 10^{-4}h^{-1}$. Es ergibt sich $p = 0.25$ als Wahrscheinlichkeit.

   Für $x = 10000h$ und $y = 1000h$ liegt es bei $\lambda = 9{,}53 \cdot 10^{-5}h^{-1}$. Es ergibt sich $p = 0.035$ als Wahrscheinlichkeit.

d) Die Wahrscheinlichkeit für ein falsches Ergebnis mit 2-aus-3-Voter ergibt sich als

$$\begin{aligned} p = {} & P[\textit{alle falsch}] + P[\textit{nur Programm 1 richtig}] + \\ & P[\textit{nur Programm 2 richtig}] + P[\textit{nur Programm 3 richtig}] \\ = {} & 0.1 \cdot 0.01 \cdot 0.001 + 0.9 \cdot 0.01 \cdot 0.001 + 0.1 \cdot 0.99 \cdot 0.001 + 0.1 \cdot 0.01 \cdot 0.999 \\ = {} & 0.001108 \end{aligned}$$

   Das ist schlechter, als wenn man lediglich das beste der drei Programme verwendet hätte. Der Einsatz des Voters lohnt sich nicht.

Aufgabe 23 Leistungsanalyse elementarer Wartesysteme

Wir wollen annehmen, daß sich die Essensausgabe an einem bestimmten Schalter in der Mensa (z.B. bei Essen 4) wie ein $M/M/1$-Wartesystem verhält, bei dem sich die potentiellen Kunden von der Länge der Warteschlange abschrecken lassen und sich dann dort seltener anstellen, wenn die Warteschlange größer ist. Die Ankunftsraten λ_k sind also nun nicht mehr gleich, sondern bestimmen sich etwa durch

$$\lambda_k := \frac{\alpha}{k+1}.$$

Die Ankunftsrate ist also kleiner, wenn die Warteschlange länger ist (größeres k). Die Größe von α soll später eingesetzt werden. Die Bedienraten μ_k sollen unabhängig von der Länge der Warteschlange alle konstant gleich μ sein.

a) Berechnen Sie die stationären Grenzwahrscheinlichkeiten unter Beachtung von

$$\sum_{i=0}^{\infty} \frac{x^i}{i!} = e^x$$

   und zeigen Sie so, daß die Anzahl der Personen an der Essenausgabe 4 eine Poisson-Verteilung besitzt.

b) Wie groß ist der Erwartungswert für die Anzahl der Kunden an der Essenausgabe im Gleichgewichtszustand?

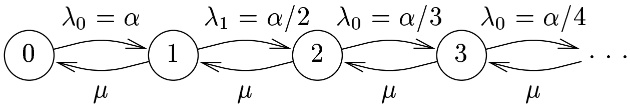


Abbildung 27: Markovkette für die Mensa

Lösung

a) Gemäß der Markovkette in Abb. 27 berechnen wir die stationären Grenzwahrscheinlichkeiten nach dem Ansatz „Abfluß = Zufluß“:

$$\begin{array}{ccccc} p_0 \cdot \frac{\alpha}{1} & = & p_1 \cdot \mu & \text{für Zustand } 0 \\ p_1 \cdot (\mu + \frac{\alpha}{2}) & = & p_2 \cdot \mu + p_0 \cdot \frac{\alpha}{1} & \text{für Zustand } 1 \\ p_2 \cdot (\mu + \frac{\alpha}{3}) & = & p_3 \cdot \mu + p_1 \cdot \frac{\alpha}{2} & \text{für Zustand } 2 \\ \vdots & & \vdots & \vdots \\ p_i \cdot (\mu + \frac{\alpha}{i+1}) & = & p_{i+1} \cdot \mu + p_{i-1} \cdot \frac{\alpha}{i} & \text{für Zustand } i \end{array}$$

Durch einfaches Umstellen und Einsetzen erhalten wir:

$$\begin{array}{ll} p_1 = & p_0 \cdot \frac{\alpha}{\mu} \\ p_2 = & \frac{1}{\mu} \cdot [p_1 \cdot (\mu + \frac{\alpha}{2}) - p_1 \cdot \mu] \quad = \quad \frac{1}{\mu} \cdot p_1 \cdot \frac{\alpha}{2} \quad = \quad \frac{p_1 \cdot \mu}{\mu^2} \cdot \frac{\alpha}{2} \quad = p_0 \cdot \frac{\alpha^2}{\mu^2} \cdot \frac{1}{2} \\ p_3 = & \frac{1}{\mu} \cdot [p_0 \cdot \frac{\alpha^2}{\mu^2} \cdot \frac{1}{2}(\mu + \frac{\alpha}{3}) - p_0 \cdot \frac{\alpha}{\mu} \cdot \frac{\alpha}{2}] = \frac{1}{\mu} \cdot p_0 \cdot \frac{\alpha^2}{\mu^2} \cdot \frac{\alpha}{2 \cdot 3} = p_0 \cdot \frac{\alpha^3}{\mu^3} \cdot \frac{1}{3!} \\ \vdots & \vdots \\ p_i = & p_0 \cdot \left(\frac{\alpha}{\mu}\right)^i \cdot \frac{1}{i!} \end{array}$$

Da zusätzlich $\sum\limits_{i=0}^{\infty} p_i = 1$ gelten muß, erhalten wir:

$$\sum_{i=0}^{\infty} p_o \cdot \left(\frac{\alpha}{\mu}\right)^i \cdot \frac{1}{i!} = 1$$

$$\Rightarrow p_0 \cdot e^{\frac{\alpha}{\mu}} = 1$$

$$\Rightarrow p_0 = e^{-\frac{\alpha}{\mu}}$$

Also für $x = \frac{\alpha}{\mu}$:

$$p_i = e^{-\frac{\alpha}{\mu}} \cdot \left(\frac{\alpha}{\mu}\right)^i \cdot \frac{1}{i!} = e^{-x} \cdot x^i \cdot \frac{1}{i!} = e^{-x} \cdot \frac{x^i}{i!}$$

Somit ist gezeigt, daß die Anzahl der Personen an der Essenausgabe 4 einer Poisson-Verteilung folgt.

b) Der Erwartungswert $E[N]$ für die Anzahl der Kunden im Gleichgewichtszustand berechnet sich wie folgt:

$$\begin{aligned} E[N] &= \sum_{i=0}^{\infty} i \cdot p_i = \sum_{i=0}^{\infty} i \cdot e^{-x} \cdot \frac{x^i}{i!} = e^{-x} \cdot \sum_{i=1}^{\infty} \frac{x^i}{(i-1)!} = e^{-x} \cdot x \cdot \sum_{i=1}^{\infty} \frac{x^{i-1}}{(i-1)!} \\ &= x \cdot e^{-x} \cdot \sum_{j=0}^{\infty} \frac{x^j}{j!} = x \cdot e^{-x} \cdot e^x = x = \frac{\alpha}{\mu} \end{aligned}$$

Aufgabe 24

Gegeben seien ein Auftragssystem AS_1 vom Typ A (Ankunftszeit $t = 0\,\mathrm{s}$) und zwei Auftragssysteme AS_2 und AS_3 vom Typ B (Ankunftszeiten $t = 0\,\mathrm{s}$ und $t = 3\,\mathrm{s}$).

a) Berechnen Sie die Auslastung jeder der vier Funktionseinheiten F_1 bis F_4 und damit den Verkehrsengpaß. Berechnen Sie die Grenzdurchsätze, das betrachtete Zeitintervall sei $[t_1, t_2] = [0\,\mathrm{s}, 30\,\mathrm{s}]$.

b) Geben Sie einen möglichen Ablauf in Balkendarstellung an.

c) Wie groß ist der Grenzdurchsatz des Systems?

d) Berechnen Sie die Mittelwerte der Füllung, der Verweilzeit und den Durchsatz in Teilaufträgen für die vier Funktionseinheiten.

e) Zeigen Sie für alle vier Funktionseinheiten das allgemeine Durchsatzgesetz.

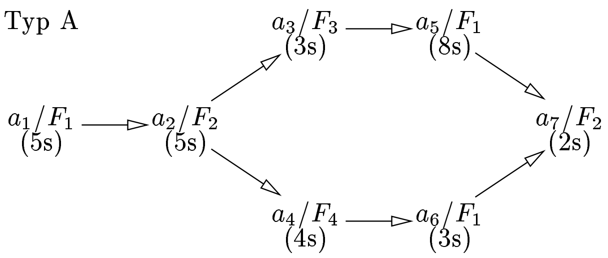
Lösung

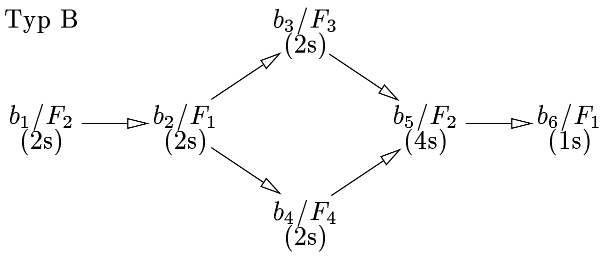
Diese Aufgabe ist dem Beispiel 5.3.3.2 aus *Rechensysteme* sehr ähnlich. Die Funktionseinheiten seien alle einfach.

a) Wir bestimmen zunächst Belegtzeit und Auslastung unter Berücksichtigung der Tatsache, daß zwei Aufträge vom Typ B zu erledigen sind.

Bedien-einheit	Belegtzeit Typ A (in s)	Typ B (in s)	Belegtzeit gesamt (in s)	Auslastung ρ_i
F_1	$5+3+8=16$	$2+1=3$	$16+2\times 3=22$	$22/30$
F_2	$5+2=7$	$2+4=6$	$7+2\times 6=19$	$19/30$
F_3	3	2	$3+2\times 2=7$	$7/30$
F_4	4	2	$4+2\times 2=8$	$8/30$

F_1 ist der Verkehrsengpaß. Besuchszahlen und Durchsatz sind offensichtlich, der Grenzdurchsatz ergibt sich z. B. aus Belegtzeit und Anzahl der zu bearbeitenden Teilaufträge.





Bedien-einheit	Besuchszahl Typ A	Typ B	mittlere Besuchszahl $\bar{v}_i$	Durchsatz d_i (in Teilauftr./s)	Grenzdurchsatz c_i (in Teilauftr./s)
F_1	3	2	7/3	7/30	7/22
F_2	2	2	6/3	6/30	6/19
F_3	1	1	3/3	3/30	3/7
F_4	1	1	3/3	3/30	3/8

b) Wird hier nicht ausgeführt.

c) Der Grenzdurchsatz des Systems ergibt sich aus den Kenngrößen des Verkehrsengpasses zu

$$c_{FES} = \frac{c_{VE}}{\bar{v}_{VE}} = \frac{c_1}{\bar{v}_1} = \frac{7/22}{7/3} = \frac{3}{22} \text{ Auftr./s.}$$

d) Für einfache Funktionseinheiten gilt: $\bar{f}_i = \rho_i$ und nach dem Gesetz von Little $\bar{f}_i = d_i \bar{y}_i$, wobei $\bar{y}_i$ die mittlere Verweilzeit in der Funktionseinheit i ist.

Bedien-einheit	mittlere Verweilzeit $\bar{y}_i$ (in s)
F_1	22/7
F_2	19/6
F_3	7/3
F_4	8/3

e) Wir zeigen das allgemeine Durchsatzgesetz gemäß der Formel 5.3.3.4 aus *Rechensysteme*

$$d_i = d_{FES} \times \bar{v}_i,$$

wobei in unserem Fall offensichtlich $d_{FES} = 3/30$.

Bedieneinheit	Probe für das allg. Durchsatzgesetz
F_1	$7/30 = 3/30 \times 7/3$
F_2	$6/30 = 3/30 \times 6/3$
F_3	$3/30 = 3/30 \times 3/3$
F_4	$3/30 = 3/30 \times 3/3$

Aufgabe 25

Das Wartenetz („Central-Server-Modell“) aus Abb. 28 beinhaltet vier Knoten K_1, K_2, K_3, K_4 und $k = 3$ Aufträge. Die Bedienzeit eines Auftrages im Knoten K_i, $i \in \{1, 2, 3, 4\}$, sei negativ-exponentiell verteilt mit den folgenden Erwartungswerten:

$$\frac{1}{\mu_1} = 0.02\,\text{s}, \qquad \frac{1}{\mu_2} = 0.2\,\text{s}, \qquad \frac{1}{\mu_3} = 0.4\,\text{s}, \qquad \frac{1}{\mu_4} = 0.6\,\text{s}.$$

Die Besuchshäufigkeiten seien wie folgt:

$$E[V_1] = 1, \qquad E[V_2] = 0.4, \qquad E[V_3] = 0.2, \qquad E[V_4] = 0.1.$$

a) Berechnen Sie sämtliche Übergangswahrscheinlichkeiten für das gegebene Wartenetz.

b) Berechnen Sie mit Hilfe der Mittelwertanalyse für sämtliche elementaren Wartesysteme des Netzes die Leistungskenngrößen mittlere Verweilzeit, Durchsatz, mittlere Anzahl von Aufträgen und Auslastung. Ein Umlauf eines Auftrags gilt als beendet, wenn er in K_1 eintritt.

(Hilfestellung liefern die Beispiele in den Abschnitten 5.4.3 und 5.4.4 aus *Jessen/Valk: Rechensysteme*).

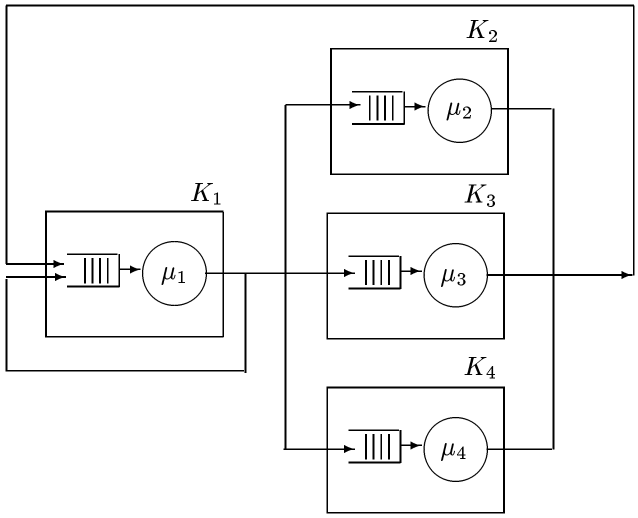


Abbildung 28: Central-Server-Modell

Lösung

a) Die Übergangswahrscheinlichkeiten ergeben sich direkt aus den Besuchshäufigkeiten. Dazu berücksichtigt man, daß für geschlossene Wartenetze gilt:

$$E[V_i] = \sum_{j=1}^{N} E[V_j] \cdot p_{ji} \text{ für } i = 1, 2, \ldots, N$$

Somit:

$$\begin{aligned} E[V_1] &= E[V_1] \cdot p_{11} + E[V_2] \cdot p_{21} + E[V_3] \cdot p_{31} + E[V_4] \cdot p_{41} \\ E[V_2] &= E[V_1] \cdot p_{12} + E[V_2] \cdot p_{22} + E[V_3] \cdot p_{32} + E[V_4] \cdot p_{42} \\ E[V_3] &= E[V_1] \cdot p_{13} + E[V_2] \cdot p_{23} + E[V_3] \cdot p_{33} + E[V_4] \cdot p_{43} \\ E[V_4] &= E[V_1] \cdot p_{14} + E[V_2] \cdot p_{24} + E[V_3] \cdot p_{34} + E[V_4] \cdot p_{44} \end{aligned}$$

Für das gegebene Netz ist klar, daß $p_{21} = p_{31} = p_{41} = 1$ gilt. Die übrigen vier Werte lassen sich leicht durch ein lineares Gleichungssystem mit vier Unbekannten und vier Gleichungen berechnen, zu:

$$p_{11} = 0.3, \quad p_{12} = 0.4, \quad p_{13} = 0.2, \quad p_{14} = 0.1$$

b) Der Ansatz der Mittelwertanalyse (vgl. „Rechensysteme“, S. 406 ff.) ergibt

$$E[Y_i(k)] = E[B_i](1 + E[F_i(k-1)]),$$

mit $Y_i(k)$ als Verweilzeit im Knoten i bei k Aufträgen im Netz und $F_i(k-1)$ als Füllung des Knotens i bei $k-1$ Aufträgen im Netz.

1. Netzfüllung $k = 1$:

$$\begin{aligned} E[Y_1(1)] &= 0.02s \cdot 1 &= 0.02s \\ E[Y_2(1)] &= 0.2s \cdot 1 &= 0.2s \\ E[Y_3(1)] &= 0.4s \cdot 1 &= 0.4s \\ E[Y_4(1)] &= 0.6s \cdot 1 &= 0.6s \end{aligned}$$

Die mittlere Umlaufzeit ist:

$$E[Y(1)] = \sum_{i=1}^{4} E[V_i] \cdot E[Y_i(1)] = 1 \cdot 0.02s + 0.4 \cdot 0.2s + 0.2 \cdot 0.4s + 0.1 \cdot 0.6s = 0.24s$$

Der Durchsatz ist nach Little ($E[F] = E[D] \cdot E[Y]$):

$$E[D(1)] = \frac{1}{E[Y(1)]} = 4.2/s$$

Knotenfüllungen:

$$\begin{aligned}
E[F_i(1)] &= E[D_i(1)] \cdot E[Y_i(1)] &&= E[V_i] \cdot E[Y_i(1)] \cdot E[D(1)] \\
E[F_1(1)] &= 1 \cdot 0.02s \cdot 4.2/s &&= 0.084 \\
E[F_2(1)] &= 0.4 \cdot 0.2s \cdot 4.2/s &&= 0.336 \\
E[F_3(1)] &= 0.2 \cdot 0.4s \cdot 4.2/s &&= 0.336 \\
E[F_4(1)] &= 0.1 \cdot 0.6s \cdot 4.2/s &&= 0.252
\end{aligned}$$

2. Netzfüllung $k = 2$:

$$\begin{aligned}
E[Y_1(2)] &= 0.02s \cdot (1 + 0.084) &&= 0.022s \\
E[Y_2(2)] &= 0.2s \cdot (1 + 0.336) &&= 0.267s \\
E[Y_3(2)] &= 0.4s \cdot (1 + 0.336) &&= 0.534s \\
E[Y_4(2)] &= 0.6s \cdot (1 + 0.252) &&= 0.751s
\end{aligned}$$

Die mittlere Umlaufzeit ist:

$$E[Y(2)] = \sum_{i=1}^{4} E[V_i] \cdot E[Y_i(2)] = 0.3107s$$

Der Durchsatz ist nach Little:

$$E[D(2)] = \frac{2}{E[Y(2)]} = 6.4/s$$

Knotenfüllungen:

$$\begin{aligned}
E[F_1(2)] &= 1 \cdot 0.022s \cdot 6.4/s &&= 0.141 \\
E[F_2(2)] &= 0.4 \cdot 0.267s \cdot 6.4/s &&= 0.684 \\
E[F_3(2)] &= 0.2 \cdot 0.534s \cdot 6.4/s &&= 0.684 \\
E[F_4(2)] &= 0.1 \cdot 0.751s \cdot 6.4/s &&= 0.481
\end{aligned}$$

3. Netzfüllung $k = 3$:

$$\begin{aligned}
E[Y_1(3)] &= 0.02s \cdot (1 + 0.141) &&= 0.023s \\
E[Y_2(3)] &= 0.2s \cdot (1 + 0.684) &&= 0.337s \\
E[Y_3(3)] &= 0.4s \cdot (1 + 0.684) &&= 0.674s \\
E[Y_4(3)] &= 0.6s \cdot (1 + 0.481) &&= 0.887s
\end{aligned}$$

Die mittlere Umlaufzeit ist:

$$E[Y(3)] = \sum_{i=1}^{4} E[V_i] \cdot E[Y_i(3)] = 0.3813s$$

Der Durchsatz ist nach Little:

$$E[D(3)] = \frac{3}{E[Y(3)]} = 7.9/s$$

Knotenfüllungen:

$$\begin{aligned}
E[F_1(3)] &= 1 \cdot 0.023s \cdot 7.9/s &&= 0.181 \\
E[F_2(3)] &= 0.4 \cdot 0.337s \cdot 7.9/s &&= 1.065 \\
E[F_3(3)] &= 0.2 \cdot 0.674s \cdot 7.9/s &&= 1.065 \\
E[F_4(3)] &= 0.1 \cdot 0.887s \cdot 7.9/s &&= 0.7
\end{aligned}$$

Somit ist die mittlere Verweilzeit $E[Y(3)] = 0.3813s$ und der mittlere Durchsatz ergibt sich zu $E[D(3)] = 7.9/s$. Die mittlere Anzahl von Aufträgen in den einzelnen Funktionseinheiten ist $E[F_1(3)] = 0.181$, $E[F_2(3)] = 1.065$, $E[F_3(3)] = 1.065$, $E[F_4(3)] = 0.700$.

Wir können jetzt eine Verprobung nach Little durchführen. Demnach wäre die erwartete Füllung

$$E[F(3)] = E[D(3)] \cdot E[Y(3)] = 3.01 \approx 3.$$

Die Abweichung ist durch Rundungsfehler entstanden.

Die Auslastungen ergeben sich aus

$$\rho_i = \frac{E[D_i(3)]}{\mu_i} = \frac{E[D(3)] \cdot E[V_i]}{\mu_i} = E[D(3)] \cdot E[V_i] \cdot E[B_i]$$

zu

$$\rho_1 = 0.158, \rho_2 = 0.632, \rho_3 = 0.632, \rho_4 = 0.474,$$

womit die Wartesysteme K_2 und K_3 den Verkehrsengpaß darstellen.

Berechnung des Grenzdurchsatzes c_s:

$$c_s = \frac{c_{VE}}{E[V_{VE}]} = \frac{1}{\mu_{VE} \cdot E[V_{VE}} = 12.5 \text{ Auftr./s}$$

Als Bedienzeit gilt die Verweilzeit bei Füllung 1:

$$E[B] = E[Y(1)] = 0.24s$$

Nun ergibt sich die Sättigungsfüllung f_s^* zu:

$$f_s^* = c_s \cdot E[B] = 3.0$$

Zusatzfrage: Ist eine Füllung des Wartenetzes mit $f > f_s^*$

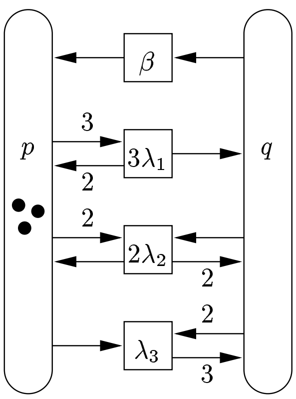
(1) möglich und

(2) sinnvoll?

Antwort: (1) $f > f_s^*$ ist problemlos möglich und (2) evtl. auch sinnvoll, sofern die dadurch steigenden Verweilzeiten noch akzeptabel sind. Die mittlere Warteschlangenlänge für K_2 und K_3 wird dabei mit zunehmender Füllung immer stärker anwachsen. (Zeigt sich bereits für Füllungen $k = 1, 2, 3, \ldots, 6$!)

Aufgabe 26

Wir betrachten folgende Modellierung eines Machine-Repair-Konzepts mit einem SPN: Ein System bestehe aus 3 (hoffentlich) arbeitenden Komponenten. Die Rate λ_1 für das Ausfallen einer der drei Komponenten des voll funktionsfähigen Systems ist kleiner als die Rate λ_2 für das Ausfallen einer Komponente, wenn nur noch zwei der Komponenten funktionsfähig sind. Enstprechend ist wiederum $\lambda_2 < \lambda_3$. (Nach Ausfall einer Komponente werden die anderen Komponenten stärker beansprucht.)
Die Zahl der Marken in der Stelle p gibt jeweils an, wieviele Komponenten des Systems noch arbeiten. Ein Reparaturtrupp repariert eine defekte Komponente mit der Rate β.



a) Zeichnen Sie den Erreichbarkeitsgraphen für das SPN und stellen Sie den infinitesimalen Generator Q auf. (Wir gehen von negativ-exponentiellen Schaltzeitintervallen aus.)

b) Bestimmen Sie die Wahrscheinlichkeitsverteilung im stationären Grenzprozess in Abhängigkeit von den Raten λ_i und β.

c) Vergleichen Sie ihr Ergebnis mit dem des Machine-Repair-Modells aus *Jessen/Valk: Rechensysteme*, S. 337–341. Setzen Sie dazu $\lambda_1 = \lambda_2 = \lambda_3$.

Lösung

$$\begin{tikzcd} m_1 = p^3 \arrow[r, "3\lambda_1"] & m_2 = p^2q \arrow[l, bend left, "\beta"] \arrow[r, "2\lambda_2"] & m_3 = pq^2 \arrow[l, bend left, "\beta"] \arrow[r, "\lambda_3"] & m_4 = q^3 \arrow[l, bend left, "\beta"] \end{tikzcd}$$

Abbildung 29: Erreichbarkeitsgraph des SPN

a) Der Erreichbarkeitsgraph ist in Abb. 29 dargestellt. Der infinitesimale Generator Q lautet:

$$Q = \begin{pmatrix} -3\lambda_1 & 3\lambda_1 & 0 & 0 \\ \beta & -(\beta + 2\lambda_2) & 2\lambda_2 & 0 \\ 0 & \beta & -(\beta + \lambda_3) & \lambda_3 \\ 0 & 0 & \beta & -\beta \end{pmatrix}$$

b) Zu Lösen ist die Gleichung $\pi Q = 0$. Durch Spaltenumformungen erhält man aus Q

$$Q' = \begin{pmatrix} -3\lambda_1 & 0 & 0 & 0 \\ \beta & -2\lambda_2 & 2\lambda_2 & 0 \\ 0 & \beta & -(\beta + \lambda_3) & \lambda_3 \\ 0 & 0 & \beta & -\beta \end{pmatrix}$$

und

$$Q'' = \begin{pmatrix} -3\lambda_1 & 0 & 0 & 0 \\ \beta & -2\lambda_2 & 0 & 0 \\ 0 & \beta & -\lambda_3 & \lambda_3 \\ 0 & 0 & \beta & -\beta \end{pmatrix}$$

Das äquivalente Gleichungssystem $\pi Q'' = 0$ lautet also

$$\begin{aligned} 3 \cdot \lambda_1 \pi_1 &= \beta \pi_2 \\ 2 \cdot \lambda_2 \pi_2 &= \beta \pi_3 \\ \lambda_3 \pi_3 &= \beta \pi_4 \end{aligned}$$

Auflösen nach π_i liefert

$$\begin{aligned}
\pi_2 &= 3\cdot\frac{\lambda_1}{\beta}\pi_1\\
\pi_3 &= 2\cdot\frac{\lambda_2}{\beta}\pi_2\\
&= 3\cdot 2\cdot\frac{\lambda_1\lambda_2}{\beta^2}\pi_1\\
\pi_4 &= \frac{\lambda_3}{\beta}\pi_3\\
&= 3\cdot 2\cdot\frac{\lambda_1\lambda_2\lambda_3}{\beta^3}\pi_1
\end{aligned}$$

Zusätzlich gilt $\sum \pi_i = 1$, also

$$\pi_1+\pi_2+\pi_3+\pi_4 = 1$$

$$\pi_1 + 3\frac{\lambda_1}{\beta}\pi_1 + 3\cdot 2\cdot\frac{\lambda_1\lambda_2}{\beta^2}\pi_1 + 3\cdot 2\cdot\frac{\lambda_1\lambda_2\lambda_3}{\beta^3}\pi_1 = 1$$

$$\pi_1 = (1+3\cdot\frac{\lambda_1}{\beta}+3\cdot 2\cdot\frac{\lambda_1\lambda_2}{\beta^2}+3\cdot 2\cdot\frac{\lambda_1\lambda_2\lambda_3}{\beta^3})^{-1}$$

$$\pi_1 = (\frac{\beta^3+3\cdot\lambda_1\beta^2+3\cdot 2\cdot\lambda_1\lambda_2\beta+3\cdot 2\cdot\lambda_1\lambda_2\lambda_3}{\beta^3})^{-1}$$

Mit $N := \beta^3+3\cdot\lambda_1\beta^2+3\cdot 2\cdot\lambda_1\lambda_2\beta+3\cdot 2\cdot\lambda_1\lambda_2\lambda_3$ folgt

$$\begin{aligned}
\pi_1 &= \frac{\beta^3}{N}\\
\pi_2 &= \frac{3\cdot\beta^2\lambda_1}{N}\\
\pi_3 &= \frac{3\cdot 2\cdot\beta\lambda_1\lambda_2}{N}\\
\pi_4 &= \frac{3\cdot 2\cdot\lambda_1\lambda_2\lambda_3}{N}
\end{aligned}$$

c) Für $\lambda_1=\lambda_2=\lambda_3=\lambda$ folgt

$$N = \beta^3+3\cdot\beta^2\lambda+3\cdot 2\cdot\beta\lambda^2+3\cdot 2\cdot\lambda^3$$

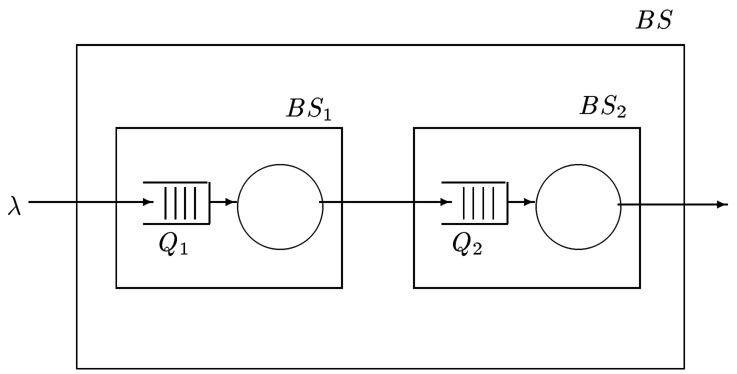
$$\pi_1 = (\frac{N}{\beta^3})^{-1} = (1+\frac{3!}{2!}\frac{\lambda}{\beta}+\frac{3!}{1!}\frac{\lambda^2}{\beta^2}+\frac{3!}{0!}\frac{\lambda^3}{\beta^3})^{-1} = (\sum_{i=0}^{3}\frac{3!}{(3-i)!}(\frac{\lambda}{\beta}))^{i-1}$$

$$\pi_i = \frac{3!}{(3-(i-1))!}(\frac{\lambda}{\beta})^{i-1}\pi_1$$

Die letzten beiden Gleichungen entsprechen (4.4.47) und (4.4.48) in *Rechensysteme*. Dabei ist Zahl der Token in q als die Anzahl der ausgefallenen Komponenten zu interpretieren, d.h. m_i entspricht dem Zustand $i-1$ des Machine-Repair-Modells.

Aufgabe 27 Leistungsanalyse eines Tandem-Wartenetzes

Bei der Modellierung eines Rechen- oder Kommunikationssystems ergebe sich das folgende Bediensystem BS („Tandem-Wartenetz“) bestehend aus zwei $M/M/1$-Bediensystemen BS_1 und BS_2 in Reihe. Die Ankunftsrate von Aufträgen sei $\lambda = 3$ [Aufträge/s].



Bekannt seien die mittleren Verweilzeiten ($\overline{Y}$) von Aufträgen/Kunden

- im gesamten Bediensystem $BS : \overline{Y}_{BS} = 2[s]$ sowie
- im Bediensystem $BS_1 : \overline{Y}_{BS_1} = 1[s]$.

a) Berechnen Sie die Bedienraten für BS_1 und BS_2.

b) Wie groß ist die mittlere Anzahl wartender Aufträge in den (unbegrenzten) Warteschlangen Q_1 und Q_2? Besteht eine Abhängigkeit dieser Ergebnisse von der Art des Ankunftstroms (insbesondere Verteilung der Zwischenankunftszeiten) und wenn ja, welche?

c) Wie ändert sich die mittlere Anzahl von Aufträgen in Q_2, wenn (bei ansonsten ungeänderten Randbedingungen) die maximale Warteschlangenlänge von Q_2 zunehmend stark begrenzt wird? Die maximale Länge von Q_1 sei weiterhin unbegrenzt.

Lösung

a) Für $M/M/1$ – Systeme gilt:

$$\overline{Y} = \frac{1}{\mu - \lambda},$$

also auch $\mu - \lambda = \frac{1}{\overline{Y}} \Rightarrow \mu = \frac{1}{\overline{Y}} + \lambda$. Da $\overline{Y}_{BS}$ und $\overline{Y}_{BS_1}$ bekannt sind, läßt sich $\overline{Y}_{BS_2}$ leicht bestimmen:

$$\overline{Y}_{BS_2} = \overline{Y}_{BS} - \overline{Y}_{BS_1} = 1[s]$$

Somit erhält man $\mu_1 = \mu_2 = 4/s$.

b) Bei gegebenem $\overline{Y}$ erlaubt das Littlesche Gesetz die Berechnung der mittleren Füllung $\overline{L}$:

$$\overline{L}_{BS_1} = \lambda \cdot \overline{Y}_{BS_1} = 3$$

$$\overline{L}_{BS_2} = \lambda \cdot \overline{Y}_{BS_2} = 3$$

Es besteht also keine Abhängigkeit von der Verteilung der Zwischenankunftszeiten, sondern nur von λ. λ und $\overline{Y}$ sind gegeben und λ wirkt auch auf BS_2.

Für unbegrenzte Warteschlangen ergibt sich folgendes Bild: Beide Bediensysteme/Wartesysteme können als $M/M/1$-Systeme angesehen werden (BS_1 mit λ, μ_1 und BS_2 mit λ, μ_2 als Parameter). Somit ist auch die Bedienrate berechenbar, wenn $\overline{Y}$ gegeben ist.

c) Für diesen Fall muß man sich zunächst überlegen, wie die begrenzte Warteschlange Q_2 wirken könnte. Dafür gibt es mehr oder weniger plausible Annahmen:

(i) Ein durch BS_1 bedienter Auftrag geht verloren, sofern Q_2 die maximale Länge erreicht hat.

(ii) Die Bedieneinheit in BS_1 blockiert, sofern sie den gerade bedienten Auftrag A_0 nicht an BS_2 weiterleiten kann (solange, bis wieder freier Platz in Q_2 zur Aufnahme von A_0 vorhanden ist).

(iii) Durch BS_1 vollständig bediente Auftrge warten noch in BS_1, solange BS_2 die maximale Füllung besitzt (ohne indes BS_1 zu blockieren, d.h. am Bedienen weiterer Auftrge zu hindern). Diese Annahme ist *nicht* mit dem Standardmodell vereinbar und würde einem zwischengeschalteten Bediensystem BS_3 entsprechen, das nur als Wartepool fungiert und dessen Bedieneinheit Aufträge – soweit möglich – unverzüglich und unverändert weiterleitet. BS_3 müßte dann eine unbegrenzte Warteschlange erlauben, um nicht nur eine Verlagerung des eigentlichen Problems darzustellen. Im Fall einer unbegrenzten Warteschlange jedoch verhält sich das System wie in Unteraufgabe b).

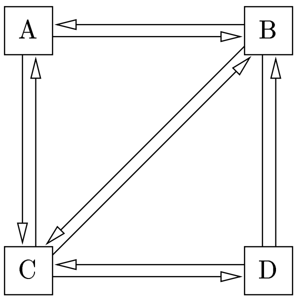
Die ursprünglichen Ankunftsraten $\lambda_1 = \lambda_2 = \lambda$ gelten auch in den Fällen (ii) und (iii) weiter, da hier keine Aufträge verloren gehen. Allerdings ist der Ankunftsstrom für BS_2 kein Poissonstrom mehr! Bei (i) gilt: $\lambda_2 < \lambda_1 = \lambda$

Bei allen Varianten reduziert sich $\overline{L}$ für Q_2 gegenüber $\overline{L}(Q_2)$ aus Teilaufgabe (b) und zwar mit zunehmender Begrenzung zunehmend stark.

Zusatzfrage: Unter welcher Voraussetzung können wir näherungsweise BS_2 als ein $M/M/1/k$-System betrachten (mit *guter* Näherung)?
Antwort: Da der Zugangsstrom für BS_2 nur durch die Blockadezustnde modifiziert wird, gilt: Wenn die Füllung von BS_2 den Maximalwert k (bzw. $k+1$) sehr selten erreicht (z.B. wenn $\mu_2 \ll \mu_1$ und k groß) können wir näherungsweise BS_2 als ein $M/M/1/k$-System betrachten. Wenn μ_2 sehr sehr groß gegenüber μ_1 ist, erfüllen auch noch relativ kleine k-Werte die Bedingung seltener Blockade-Zustnde.

Aufgabe 28 Leistungsanalyse eines Kommunikationsnetzes

Gegeben sei das folgende Kommunikationsnetz N, das die vier Knoten(-rechner) A, B, C und D miteinander verbindet.



		Ziel			
	$E[Z_{ij}]$	A	B	C	D
Quelle	A	0	8	11	7
	B	9	0	7	6
	C	3	3	0	10
	D	4	7	10	0

An den Zugängen zu den Knoten werden Nachrichten in das Netz eingespeist, die dann (evtl. über Zwischenstationen) zu den jeweiligen Zielknoten befördert werden sollen. Durch Schätzungen sei die Zugangsmatrix $(E[Z_{ij}])$, $i, j \in \{A, B, C, D\}$ bekannt, wobei $E[Z_{ij}]$ den Erwartungswert für die Anzahl der Pakete pro Sekunde angibt, die in den Knoten i eingespeist werden und an den Knoten j ausgeliefert werden sollen. Der (externe) Zugang zu dem Knoten i ist dann $E[Z_i] = \sum_j E[Z_{ij}]$.

Die Nachrichten werden vor dem Verschicken in Pakete fester Größe (1 kbit) zerlegt, benötigen dann aber noch einen Verpackungsoverhead von 20%, so daß sich eine Paketgröße von 1,2 kbit ergibt.

Der Grenzdurchsatz der Leitungen zwischen den Knoten betrage 30 kbit/s.

a) Berechnen Sie den Durchsatz des Netzes $E[D_N]$.

b) Folgende Leitungswahlregel für die Wegeermittlung verwendet: *„Nachrichten zwischen A und D werden über B ausgetauscht."*

- Berechnen Sie die Leitungsdurchsätze $E[D_{ij}]$.
- Welche Leitung ist der Verkehrsengpaß und wie groß ist die Auslastung dieser Leitung?
- Wie groß ist der Grenzdurchsatz des Netzes?

Lösung

a) Sei $\mathcal{L}$ die Menge der Leitungen. Dann gilt, sofern die Auslastung sämtlicher Leitungen kleiner als eins ist, d.h. falls $\forall j \in \mathcal{L} : \rho_j < 1$ ist:

$$E[D_N] = \sum_{i=1}^{m} E[Z_i] = \sum_{i=1}^{m} \sum_{j=1}^{m} E[Z_{ij}] = 26 + 22 + 16 + 21 = 85 \,[\text{Pakete/s}]$$

b) Berechnung von $E[Dij]$:

$$\begin{array}{rcrcr}
A \to B & : & 8\,[\text{Pakete/s}] + 7\,[\text{Pakete/s}] & = & 15\,[\text{Pakete/s}] \\
A \to C & : & & & 11\,[\text{Pakete/s}] \\
B \to A & : & 9\,[\text{Pakete/s}] + 4\,[\text{Pakete/s}] & = & 13\,[\text{Pakete/s}] \\
B \to C & : & & & 7\,[\text{Pakete/s}] \\
B \to D & : & 6\,[\text{Pakete/s}] + 7\,[\text{Pakete/s}] & = & 13\,[\text{Pakete/s}] \\
C \to A & : & & & 3\,[\text{Pakete/s}] \\
C \to B & : & & & 3\,[\text{Pakete/s}] \\
C \to D & : & & & 10\,[\text{Pakete/s}] \\
D \to B & : & 7\,[\text{Pakete/s}] + 4\,[\text{Pakete/s}] & = & 11\,[\text{Pakete/s}] \\
D \to C & : & & & 10\,[\text{Pakete/s}]
\end{array}$$

wobei der Verkehr zwischen A und D über zwei Leitungen läuft.

$$\sum E[D_{ij}] = 96 \,[\text{Pakete/s}]$$

Die Leitung $A \to B$ stellt den Verkehrsengpaß dar. Die Auslastung dieser Leitung beträgt

$$\rho_{VE} = \frac{15 \cdot 1200}{30000} = 0.6$$

Da sich der Faktor „kilo" bei der Berechnung herauskürzen läßt, ist es gleichgültig, ob 1 kbit mit 1024 bit oder 1000 bit angesetzt wird!

Der Grenzdurchsatz des Netzes c_N ist durch den Durchsatz und die Auslastung des Verkehrsengpassen gegeben zu

$$c_n = \frac{E[D_N]}{\rho_V E} = \frac{85\,[\text{Pakete/s}]}{0.6} = 141.6\,[\text{Pakete/s}]$$

obwohl die 10 Leitungen bis zu

$$10 \cdot \frac{30000}{1200} \,[\text{Pakete/s}] = 250\,[\text{Pakete/s}]$$

transportieren könnten.