1. Theorie paralleler Betriebssystem-Prozesse

Literaturhinweis: ZIMA, H.: Betriebssysteme. Parallele Prozesse. Bibl. Institut, Zürich 1986.

1.1. Prozeßsysteme und ihre Beschreibung

1.1.1. Prozesse und Betriebsmittel

- ***Prozeßbegriff:*** identifizierbare Folge von Aktionen eines Prozessors aufgrund einer strukturierten Menge von Anweisungen, die zu einer Folge von Zustandsänderungen im Rechensystem und in seiner Umgebung führen (nach WERNER).

- ***Parallele Prozesse:*** Prozesse, die sich gleichzeitig zwischen Start und Ende befinden.

  Asynchronität $\rightarrow$ Wechselwirkungen

- ***Koordination (Synchronisation i.w.S.):*** zielgerichtete Steuerung der logischen und zeitlichen Abläufe von parallelen Prozessen entsprechend der angestrebten Wechselwirkung.

- ***Betriebsmittel (BM):*** Komponenten eines Rechensystems, die von den einzelnen Prozessen während ihres Ablaufs benötigt werden.

  Klassifikation:
  globale BM; mehrere Prozesse können gleichzeitig zugreifen;
  lokale BM; nur durch jeweils einen Prozeß verwendbar.

  B: Menge der globalen Betriebsmittel, n: Anzahl.

- ***Wert (Zustand) der BM:*** $w_b \in W_b$, $b \in B$ für globale BM, analog für lokale Betriebsmittel.

- ***Zustand des Rechensystems:*** $z = (w_1,...,w_n, \ldots)$.

1.1.2. Verarbeitungsschritte

- ***Verarbeitungsschritt*** *s* **:** besteht aus

  Eingabeaktion e_σ von einem Eingabebereich $B^e_\sigma \subseteq B$

  Transformation f_σ

  Ausgabeaktion a_σ nach einem Ausgabebereich $B^a_\sigma \subseteq B$.

  Σ: Menge aller Verarbeitungsschritte eines Prozeßsystems,

  m: Anzahl.

- ***Uninterpretiertes Prozeßsystem:*** Vernachlässigung der Transformation f_σ, Reduzierung eines Verarbeitungsschrittes auf die Form

  $\sigma = e_\sigma a_\sigma$

- ***Inzidenzmatrizen:***

  Eingabematrix $\mathbf{B}^e = (m^e_{\sigma,b})_{\sigma\in\Sigma,\ b\in B}$

  $m^e_{\sigma,b}$: Anzahl der dem Schritt σ zugeordneten BM $b \in B^e_\sigma$

  Ausgabematrix $\mathbf{B}^a = (m^a_{\sigma,b})_{\sigma\in\Sigma,\ b\in B}$

  $m^a_{\sigma,b}$: Anzahl der dem Schritt σ zugeordneten BM $b \in B^a_\sigma$

  Bsp. 1.

$$\mathbf{B}^e = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{B}^a = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

1.1.3. Datenabhängigkeit

- Es sei für $\sigma,\sigma' \in \Sigma$:

  $B_{\sigma,\sigma'} := (B^a{}_\sigma \cap B^a{}_{\sigma'}) \cup (B^e{}_\sigma \cap B^a{}_{\sigma'}) \cup (B^a{}_\sigma \cap B^e{}_{\sigma'})$.

- $\sigma,\sigma' \in \Sigma$ heißen *datenabhängig*, wenn $B_{\sigma,\sigma'} \neq \emptyset$.

  Datenabhängigkeitsrelation:

  $G := \{(\sigma,\sigma') \mid B_{\sigma,\sigma'} \neq \emptyset \ \wedge \ \sigma \neq \sigma'\} \subseteq \Sigma \times \Sigma$.

  Bsp. 2. G = { }

  G ist per def. irreflexiv und symmetrisch.

  Darstellung von G als *Datenabhängigkeitsmatrix* **G.**

  Bsp. **G** = $\begin{pmatrix} & \\ & \\ & \end{pmatrix}$

- Darstellung als *Datenabhängigkeitsgraph*:

  Knoten: □ $\sigma \in \Sigma$, ○ $b \in B$

  Kanten: $\rightarrow$ (b, σ) bei $b \in B^e{}_\sigma$, (σ, b) bei $b \in B^a{}_\sigma$.

  Bsp. 4.

1.1.4. Präzedenz und Adjazenz

- ***Präzedenzrelation:***

  $R \subseteq \Sigma \times \Sigma$ irreflexiv, asymmetrisch; transitiv.

  Schreibweise bei $(\sigma,\sigma') \in R$: $\sigma < \sigma'$.

  Interpretation bei $(\sigma,\sigma') \in R$: σ muß vor Beginn von σ' beendet sein.

  Sprechweise bei $(\sigma,\sigma') \notin R \wedge (\sigma',\sigma) \notin R$:
  σ,σ' sind parallel ausführbar.

  Darstellung als *Präzedenzmatrix* **R**.

  Bsp. 5.

$$\mathbf{R} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- ***Adjazenzrelation:***

  $A_R = \{(\sigma,\sigma') \in R \mid \sigma$ ist unmittelbarer Vorgänger von $\sigma'\}$.

  Schreibweise bei $(\sigma,\sigma') \in A_R$: $\sigma \lhd \sigma'$.

  Bsp. 6.

  $\mathbf{A} = \begin{pmatrix} & \\ & \\ & \end{pmatrix}$

  Ist R eine Halbordnungsrelation, so gilt: $A_R = R \setminus R^2$.

  Bsp. 7.

  Darstellung als (Präzedenz-)Graph:

1.1.5. Formale Darstellung eines Prozeßsystems

- Ein ***Prozeßsystem über einer Menge B*** von globalen Betriebsmitteln ist ein Quadrupel $\Pi = (\Sigma, R, Z, z_0)$ mit
  - Σ: Menge der Verarbeitungsschritte; zu $\sigma \in \Sigma$ gehört
    Eingabebereich $B^e_\sigma \subseteq B$, Ausgabebereich $B^a_\sigma \subseteq B$,
    Einbageaktion e_σ, Ausgabeaktion a_σ;
  - R: Präzedenzrelation (strikte Halbordnungsrelation);
    Reihenfolge, in der Verarbeitungsschritte auszuführen sind;
  - Z: Menge der möglichen Zustände;
    $Z = \bigtimes_{b \in B} W_b$, W_b Wertemenge von $b \in B$;
  - z_0: Anfangszustand, $z_0 \in Z$.

- Implizite Annahmen: Prozeßsystem ist asynchron und uninterpretiert.

- Weiter sei

  T: Menge aller Ein- und Ausgabeaktionen von Σ.

  R wird auf T fortgesetzt durch die zusätzliche Festlegung

  $(e_\sigma, a_\sigma) \in R \quad \forall \sigma \in \Sigma$.

1.2. Determinierte Prozeßsysteme

1.2.1. Aktions-, Zustands- und Wertefolgen

Gegeben sei ein Prozeßsystem $\Pi = (\Sigma, R, Z, z_0)$ über B;
weiter sei $|\Sigma| = m$, $|B| = n$, $Z = \bigtimes_{b \in B} W_b$, T Menge aller Aktionen.

- ***Zulässige Aktionsfolge*** α***:*** jede Folge der Länge 2m von Elementen aus T (Permutationen ohne Wiederholung), die R genügt.

A: Menge der zulässigen Aktionsfolgen.

- ***Teilaktionsfolge*** α' ***einer zulässigen Aktionsfolge*** α***:*** jeder zusammenhängende Abschnitt der Länge l ($0 \leq l \leq 2m$) einer zulässigen Aktionsfolge.

- ***Von*** $\alpha \in A$ ***erzeugte Zustandsfolge*** $z(\alpha)$***:*** Folge der zugehörigen Zustände des Prozeßsystems.

- ***Von*** $\alpha \in A$ ***erzeugte Wertefolge*** $w_b(\alpha)$ ***des Betriebsmittels*** $b \in B$***:***
  Sei $b \in B$, $\alpha = t_1 ... t_{2m} \in A$, $\zeta = \zeta(\alpha) = z_1 ... z_{2m}$; dann werden aus ζ alle Zustände t_j ($j \in \{1,...,2m\}$) gestrichen, für die gilt:

  $$\exists\, \sigma \in \Sigma: (t_j = e_\sigma) \vee (t_j = a_\sigma \wedge b \notin B^a_\sigma).$$

  Die von α erzeugte Wertefolge $w_b(\alpha)$ des Betriebsmittels b ist dann die Folge der b-ten Projektionen der verbleibenden Zustandsfolge.

  Analog für Teilaktionsfolgen.

1.2.2. Determiniertheit und Störungsfreiheit

- Ein *Prozeßsystem* heißt *determiniert*, wenn die Wertefolgen $w_b(\alpha)$ aller globalen Betriebsmittel $b \in B$ nur vom Anfangszustand, nicht aber von den zulässigen Aktionsfolgen abhängen.

- Ein *Prozeßsystem* heißt *schwach determiniert*, wenn der Endzustand z_{2m} nur vom Anfangszustand, nicht aber von den zulässigen Aktionsfolgen abhängt:

  $$z_{2m}(\alpha) = z_{2m}(\alpha') = f(z_0) \quad \forall \alpha, \alpha' \in A.$$

- Zwei verschiedene *Verarbeitungsschritte* $\sigma, \sigma' \in \Sigma$ heißen *störungsfrei*, wenn gilt:

  $$(\sigma, \sigma') \in R \vee (\sigma', \sigma) \in R \vee B_{\sigma,\sigma'} = \emptyset \qquad (\sigma \neq \sigma').$$

- $\sigma \in \Sigma$ heißt *verlustfrei* bei $B^a_\sigma \neq \emptyset$.

- Ein *Prozeßsystem* Π heißt *störungsfrei*, wenn es nur einen Verarbeitungsschritt enthält oder alle Schritte paarweise störungsfrei sind. Es heißt *verlustfrei*, wenn alle Verarbeitungsschritte verlustfrei sind.

1.2.3. Charakterisierung determinierter Prozeßsysteme

Hilfssatz. Sei $\Pi = (\Sigma, R, Z, z_0)$ ein störungsfreies Prozeßsystem über B, und es sei $\sigma = ea$ ein maximales Element bzgl. R. Wenn es eine Aktionsfolge $\alpha = \alpha_1 e \alpha_2 a \alpha_3 \in A$ gibt, so ist auch $\alpha' = \alpha_1 \alpha_2 \alpha_3 ea \in A$, und es gilt:

$w_b(\alpha) = w_b(\alpha') \quad \forall b \in B.$

Zum Beweis.

a) α zulässig $\rightarrow \neg(\alpha_3 < a)$

σ maximal $\rightarrow \neg(a < \alpha_3)$

Also sind a, α_3 parallel ausführbar.

Analog für e und $\alpha_2\alpha_3$.

b) Fallunterscheidungen bzgl. $b \notin B^a{}_\sigma$ bzw. $b \in B^a{}_\sigma$;
wegen $a \parallel \alpha_3$ bzw. $e \parallel \alpha_2\alpha_3$ gilt:
Störungsfreiheit $\rightarrow$ Datenunabhängigkeit.
Damit: $w_b(\alpha) = w_b(\alpha')$.

Satz 1. Jedes störungsfreie Prozeßsystem ist determiniert.

Beweis. vollständige Induktion über $|\Sigma|$.

a) $|\Sigma| = 1$: Π störungsfrei per def. und determiniert (trivial).

b) $\forall\ \tilde{\Pi} = (\tilde{\Sigma}, \tilde{R}, \tilde{Z}, \tilde{z}_0)$ mit $|\tilde{\Sigma}| = m$ und $\tilde{\Pi}$ störungsfrei gelte:
$\tilde{\Pi}$ ist determiniert.

c) Sei $\Pi = (\Sigma, R, Z, z_0)$ mit $|\Sigma| = m+1$ ein störungsfreies PS.

Dann gibt es ein $\sigma = ea \in \Sigma$, das maximal bzgl. R ist.

Seien $\alpha = \alpha_1 e \alpha_2 a \alpha_3 \in A$, $\beta = \beta_1 e \beta_2 a \beta_3 \in A$, $b \in B$ bel.

Dann ist nach obigem Hilfssatz auch
$\alpha' = \alpha_1 \alpha_2 \alpha_3 ea \in A$, $\beta' = \beta_1 \beta_2 \beta_3 ea \in A$

mit
$w_b(\alpha) = w_b(\alpha'), \qquad w_b(\beta) = w_b(\beta') \qquad \forall\ b \in B. \qquad (*)$

Aus Π werde $\tilde{\Pi}$ gebildet mit

Nach b) ist $\tilde{\Pi}$ determiniert, also

$$w_b(\tilde{\alpha}) = w_b(\tilde{\beta}) \qquad \forall b \in B.$$

Dann gilt für α bzw. β:

– $b \notin B^a_\sigma$: $w_b(\alpha') = w_b(\tilde{\alpha})$, $w_b(\beta') = w_b(\tilde{\beta})$.

– $b \in B^a_\sigma$: Sowohl nach $\tilde{\alpha}$ als auch nach $\tilde{\beta}$ liegt gleicher Zustand vor $\rightarrow$ gleicher Ausgabewert x,
mithin
$w_b(\alpha') = w_b(\tilde{\alpha})x$, $w_b(\beta') = w_b(\tilde{\beta})x$.

Also gilt in beiden Fällen:

$$w_b(\alpha') = w_b(\beta')$$

und wegen (*)

$$w_b(\alpha) = w_b(\beta). \qquad \blacksquare$$

Satz 2. Jedes determinierte und verlustfreie Prozeßsystem ist störungsfrei.

Beweis. Indirekt.

Ang., es gebe determiniertes, verlustfreies Prozeßsystem Π über B, das nicht störungsfrei ist.

Dann gibt es $\sigma, \sigma' \in \Sigma$: parallel ausführbar und datenabhängig.

Damit:

$$\alpha = \alpha_1\sigma\sigma'\alpha_2 \in A, \qquad \beta = \beta_1\sigma'\sigma\beta_2 \in A.$$

a) $\mathbf{B^a}_s \cap \mathbf{B^a}_{s'} \neq \emptyset$:

$\exists\, b \in B^a_\sigma \cap B^a_{\sigma'}$;

Konstruktion zweier Interpretationen:

f_σ schreibt x auf b, $f_{\sigma'}$ schreibt y auf b
mit $x \neq y$.

Dann ist

$w_b(\alpha) = w_b(\alpha_1)\, xy\, w_b(\alpha_2)$

$w_b(\beta) = w_b(\beta_1)\, yx\, w_b(\beta_2)$.

b) $\mathbf{B^a}_s \cap \mathbf{B^a}_{s'} = \emptyset \vee \mathbf{B^a}_s \cap \mathbf{B^e}_{s'} \neq \emptyset$:

$\exists\, b \in B^a_\sigma \cap B^e_{\sigma'}$;

Π verlustfrei $\rightarrow \exists\, b' \in B^a_{\sigma'} \setminus B^a_\sigma$

Konstruktion zweier Interpretationen:

f_σ ändert Wert x von b in x',

f_σ ist 1-1-Abbildung von b nach b': $\varphi: W_b \rightarrow W_{b'}$.

Dann erzeugen α und β unterschiedliche Wertefolgen auf b':

α: $\sigma\sigma'$ schreibt auf b' $\varphi(x')$

β: $\sigma'\sigma$ schreibt auf b' $\varphi(x)$

c) $\mathbf{B^a}_s \cap \mathbf{B^a}_{s'} = \emptyset \vee \mathbf{B^e}_s \cap \mathbf{B^a}_{s'} \neq \emptyset$:
analog. ■

Weiter gilt: Ein schwach determiniertes verlustfreies Prozeßsystem ist nicht notwendig störungsfrei.

1.2.4. Maximal parallele Prozeßsysteme

- Ein störungsfreies Prozeßsystem $\Pi = (\Sigma, R^p, Z, z_0)$ über B heißt *maximal parallel*, wenn es nach Streichen eines Elements aus der zu R^p gehörigen Adjazenzmatrix A_{R^p} nicht mehr störungsfrei ist.

Satz 3. Sei $\Pi = (\Sigma, R, Z, z_0)$ ein störungsfreies Prozeßsystem über B und G seine Datenabhängigkeitsmatrix. Dann ist das Prozeßsystem $\Pi^p = (\Sigma, R^p, Z, z_0)$, wobei R^p die transitive Hülle von $U = R \cap G$ ist, maximal parallel.

Beweis. Sei A_U die Adjazenzrelation von U (und damit von R_p). Offenbar ist $A_U \subseteq U$. Nun werde für ein beliebiges $(\sigma,\sigma') \in A_U$ die Relation

$S = A_U \setminus \{(\sigma,\sigma')\}$

betrachtet. Dann gilt weiterhin $(\sigma,\sigma') \in G$, d.h. $B_{\sigma,\sigma'} \neq \emptyset$. Ferner ist wegen $(\sigma,\sigma') \in R$ und der Asymmetrie von R

$(\sigma',\sigma) \notin R$, damit $(\sigma',\sigma) \notin S$.

Mit

$\tilde{S}$: transitive Hülle von S

folgt

$\tilde{S} \subseteq R$ und damit $(\sigma',\sigma) \notin \tilde{S}$

und weiter

$(\sigma,\sigma') \notin \tilde{S}$,

also ist Π nach Streichen von (σ,σ') nicht störungsfrei. ■

- ***Bemerkung.*** Das so konstruierte maximale Prozeßsystem ist bis auf Äquvalenz eindeutig bestimmt.

Bsp. 8.

$$\mathbf{B}^e = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad \mathbf{B}^a = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Datenabhängigkeitsgraph:

$$\mathbf{G} = \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix}$$

1.3. Überlagerung von Prozeßsystemen

- (Serielle) Verkettung von Prozeßsystemen
- Parallelität von Prozeßsystemen
- Verwaltete Betriebsmittel

1.4. Koordinierung paralleler Prozesse

1.4.1. Begriffe

- ***Konkurrenz:*** mehrere Prozesse bewerben sich unabhängig voneinander um die zeitweilig exklusive Nutzung von verwalteten Betriebsmitteln.

- ***Kritischer Abschnitt:*** Gegeben sei ein Prozeßsystem Π sowie eine Menge K von Teilaktionsfolgen. Dann heißt $\kappa \in K$ *kritischer Abschnitt zur Klasse K*, wenn κ nicht gleichzeitig mit einer anderen Teilaktionsfolge aus K aktiv sein darf.

  (*aktiv*: mind. 1 Verarbeitungsschritt ist noch nicht beendet)

- ***Wechselseitiger Ausschluß:*** Koordinierung der Abläufe von Prozeßsystemen Π_1, Π_2,... so, daß kritische Abschnitte derselben Klasse nur von 1 Prozeßsystem betreten werden können.

- ***Protokoll:*** Steueralgorithmus, der der gegenseitigen Verständigung von parallelen, asynchronen Prozessen zwecks Koordinierung ihrer Abläufe dient.

  Vorprotokoll: Herstellen einer zweckgebundenen logischen Verbindung zu einem oder mehreren Prozessen;

  Nachprotokoll: Lösen der logischen Verbindung.

- ***Anforderungen*** *an Protokolle*, unabhängig von Geschwindigkeit der Prozesse:

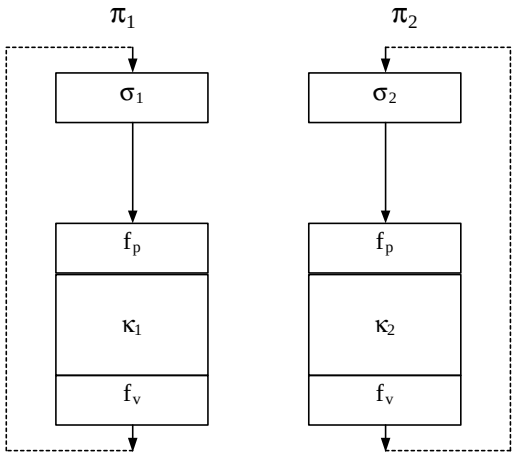
– Sicherheit: wechselseitiger Ausschluß muß gewährleistet sein;

– Lebendigkeit: Jeder Prozeß muß nach endlicher Zeit k.A. betreten können.

  Nichtvorhandensein von Fernwirkung – Ausgrenzung – Verklemmung.

1.4.2. Dezentrale Lösung für 2 Prozesse nach DEKKER/DIJKSTRA

- ***Grobstruktur***



- ***Algorithmus DD***

- ***Temporale Logik:***

KRÖGER, F.: Temporal Logic of Programs. Springer 1987.

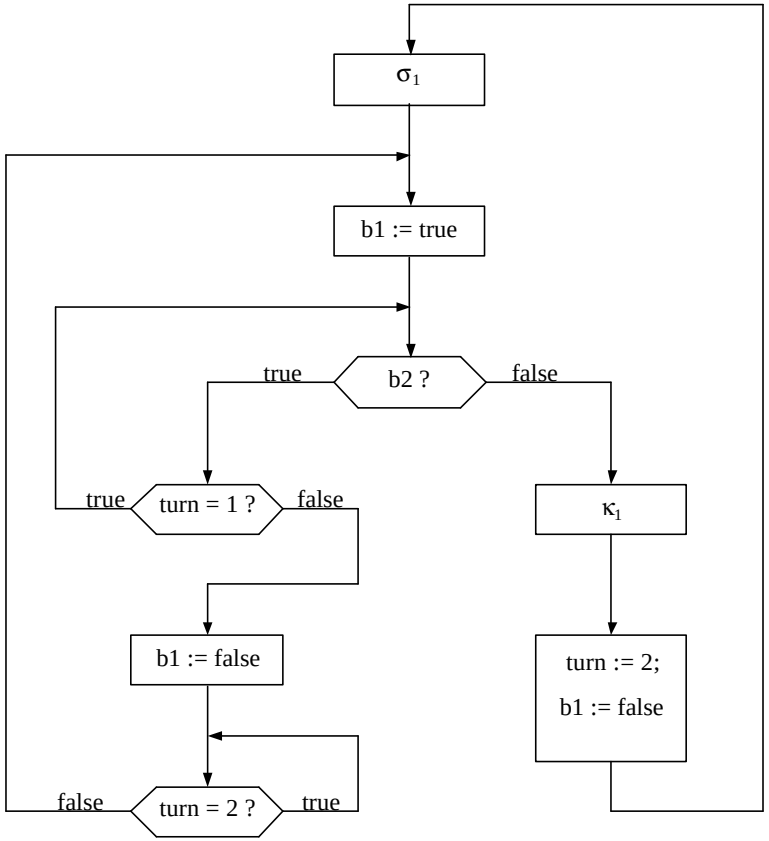
Zeit: unendliche diskrete linear geordnete Menge mit kleinstem Element. Operatoren u. a.:

ALWAYS A	□A	A gilt von nun an zu jedem Zeitpunkt
SOMETIMES A	◊A	gilt zu einem zukünftigen Zeitpunkt
A WHILE B		A gilt, solange B gilt.

Satz 4. Das Protokoll des Algorithmus DD ist sicher.

Satz 5. Das Protokoll des Algorithmus DD ist lebendig.


```
b1 := false;  b2 := false;  turn := 1;      bi: Anmeldung, turn: Vorrecht
{σ1}                                        anderer Verarbeitungsschritt
b1 := true;                                 Anmeldung
while b2 do                                 warten, bis p2 Anmeldung zurücknimmt
 if turn = 2 then begin b1 := false;        p2 hat Vorrang: Anmeldung zurückn.
                        while turn = 2 do;  warten, bis p2 Vorrang abgibt
                        b1 := true          Anmeldung wiederholen
                  end;
{κ1}                                        kritischer Abschnitt
turn := 2;  b1 := false;                    Vorrang abgeben, Anmeld. zurückn.
```



Beweis von Satz 4. DD ist sicher.

Offenbar gilt zu jedem Zeitpunkt:

π_1 tritt in κ_1 ein $\rightarrow$ $\neg b_2$

π_2 ist in κ_2 $\rightarrow$ b_2

d.h.

$\neg b_2 \rightarrow \neg(\pi_2 \text{ ist in } \kappa_2)$,

woraus unmittelbar folgt

π_1 tritt in κ_1 ein $\rightarrow$ $\neg(\pi_2 \text{ ist in } \kappa_2)$. (*)

Analog ergibt sich:

π_1 ist in κ_1 $\rightarrow$ b_1

$b_1 \rightarrow \neg(\pi_2 \text{ tritt in } \kappa_2 \text{ ein})$,

also zusammengefaßt

π_1 ist in κ_1 $\rightarrow$ $\neg(\pi_2 \text{ tritt in } \kappa_2 \text{ ein})$

und damit

$\neg(\pi_2 \text{ tritt in } \kappa_2 \text{ ein})$ WHILE $(\pi_1 \text{ ist in } \kappa_1)$.

Mit (*) folgt daher insgesamt:

$\neg(\pi_2 \text{ ist in } \kappa_2)$ WHILE $(\pi_1 \text{ ist in } \kappa_1)$.

Entsprechend für vertauschte Indizes. ■

Beweis von Satz 5. DD ist lebendig.

Voraussetzungen.

– $\alpha_1,...\alpha_8$ bzw. $\beta_1,...,\beta_8$ bezeichnen die logischen Variablen dafür, daß im PAP der Prozeß π_1 bzw. π_2 die entsprechende Stelle erreicht hat.

– Da b_i nur von π_i (i = 1,2) geändert werden kann, gilt:

$$\left.\begin{array}{l} b_1 \Leftrightarrow \alpha_3 \vee \alpha_4 \vee \alpha_5 \vee \alpha_6 \vee \alpha_7 \\ b_2 \Leftrightarrow \beta_3 \vee \beta_4 \vee \beta_5 \vee \beta_6 \vee \beta_7 \end{array}\right\} (1)$$

– Zu beweisen: $\alpha_2 \rightarrow \Diamond\alpha_5$.

Beweis indirekt: Es gelte

$\alpha_2 \wedge \neg\Diamond\alpha_5$.

a) Angenommen, es gilt

$\Box(\text{turn} = 2)$. (2)

Dann hat π_1 folgenden Ablauf:

$\alpha_2 - \alpha_3 - \alpha_4 - \alpha_6 - \alpha_8 -$ Schleife.

Damit wäre von nun an immer $\neg b_1$ erfüllt. Daher könnte nun π_2 in κ_2 eintreten. Nach Verlassen von κ_2 folgt nach β_7

turn := 1

b) Also kann ausgegangen werden von

$\alpha_2 \wedge \neg\Diamond\alpha_5 \wedge \Diamond$ (turn = 1). (3)

Dann folgt auch $\neg\Diamond\alpha_7$; da aber turn := 2 nur nach α_7 ausgeführt wird, folgt aus (3) sogar

$\Diamond \Box$ (turn = 1). (4)

Mithin käme π_1 irgendwann nach α_4 in eine Schleife:

$\Diamond \Box$ (π_1 in Schleife $\alpha_3 - \alpha_4$). (5)

In dieser Schleife wäre offenbar b_1 erfüllt, mithin folgt

$\Diamond \Box b_1$. (6)

c) Betrachtung in dieser Situation von π_2:

$\beta_2 - \beta_3 - \beta_4 - \beta_6 - \beta_8 -$ Schleife,

und nach (1) gälte dann $\neg b_2$, also

$\Diamond \Box \neg b_2$. (7)

Dann folgt aus (5) und (7) für π_1:

$\Diamond (\alpha_3 \wedge \neg b_2)$

und somit

$\Diamond \alpha_5$

q.e.d.

Aufgaben I.

1. Ein System bestehe aus 2 Verarbeitungsschritten (parallel ausführbar) σ_1, σ_2 und besitze 2 globale Betriebsmittel b_1, b_2 (Werte: natürliche Zahlen). Die Summe der Werte der Betriebsmittel wird durch σ_1 auf b_1 geschrieben, durch σ_2 auf b_2; der Anfangszustand sei $(1\ 2)^T$.
   a) Geben Sie die formale Beschreibung als Prozeßsystem an, untersuchen Sie die Datenabhängigkeit!
   b) Ermitteln Sie alle zulässigen Aktionsfolgen und die zugehörigen Wertefolgen der beiden Betriebsmittel!

2. Gegeben seien

   $\Sigma = \{1,...,7\}$, $B = \{a,...,g\}$,

$$\mathbf{B}^e = \begin{pmatrix} 1&0&1&0&0&0&0\\ 0&1&0&0&0&0&0\\ 1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0\\ 0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0\\ 0&0&0&0&1&0&1 \end{pmatrix}, \quad \mathbf{B}^a = \begin{pmatrix} 0&1&0&0&0&0&0\\ 0&0&1&0&0&0&0\\ 0&0&0&1&0&0&0\\ 0&0&1&0&1&0&0\\ 0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1 \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} 0&1&1&1&1&1&1\\ 0&0&0&1&0&0&1\\ 0&0&0&0&1&1&1\\ 0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0 \end{pmatrix}.$$

   Bestimmen Sie
   a) den Datenabhängigkeitsgraphen
   b) die Datenabhängigkeitsmatrix
   c) die zu $\mathbf{R}$ gehörige Adjazenzrelation
   d) den Präzedenzgraphen!

3. Es sei

   $\Sigma = \{\sigma_1, \sigma_2, \sigma_3\}$, $R = \{(\sigma_1, \sigma_3), (\sigma_2, \sigma_3)\}$, $B = \{b_1, b_2\}$, $z = (x\ y)^T$ $(x,y \in \mathbb{N})$, $z_0 = (2\ 1)^T$,

   $f_1: x \mapsto 2x$, $f_2: y \mapsto 3y$, $f_3: x \mapsto 2x,\ y \mapsto 3y$.

   Untersuchen Sie die Datenabhängigkeit und die Determiniertheit!