Lösungsvorschlag Aufgabe 1: Begriffe

a) Spezifikation und Implementierung

Spezifikation ist eine abstrakte Verhaltensbeschreibung eines Prozesses, die zur Gewinnung des dokumentierten Programms einer weiteren Detaillierung bedarf.

Eine Implementierung ist eine Konkretisierung einer Spezifikation, wobei diese durch Details angereichert wird.

b) Bedeutung der Funktion *rep*

Die Funktion *rep* stellt eine Beziehung zwischen einer Implementierung und der zugehörigen Spezifikation her, wobei die Zustandsinformation der Implementierung auf einen Zustand der Spezifikation abgebildet wird.

$N_h('y)$ bzw. $N_n('x)$ ist das Nichblockierungsprädikat der Spezifikation bzw. der Implementierung für den Zustand *'y* bzw. *'x*.
$E_h('y,y)$ bzw. $E_n('x,x)$ ist das Effektprädikat der Spezifikation bzw. der Implementierung.

Eine Verfeinerung wird als Implementierung akzeptiert, wenn folgende Kriterien zutreffen:

1)

$$\mathrm{INIT}_n\,(x) \xrightarrow{rep} \mathrm{INIT}_h\,(y)$$

2)

$$\begin{tikzcd}
\text{Spezifikation} & 'y \arrow[r, "N_h('y) \wedge E_h('y,\, y)"] & y \\
\text{Implementierung} & 'x \arrow[u, "rep"] \arrow[r, "N_n('x) \wedge E_n('x,\, x)"] & x \arrow[u, "rep"']
\end{tikzcd}$$

c) Was ist eine Aktion ?

F.Hofmann, Betriebssysteme: Grundkonzepte und Modellvorstellungen, 1991: S. 46

Eine Aktion ist die Durchführung einer Aktivität durch einen Aktivitätsträger. Für Überlegungen im Zusammenhang mit Betriebssystemen reicht es aus, Aktionen als zeitlich nicht weiter zerlegbar anzusehen (atomar !).

Eine Aktion hat die Struktur:

$a_{i,j}$: WHEN $'L_i = j$ AND $B_{i,j}(\,'G\,)$
DO $L_i = N_{i,j}(\,'G\,)$
AND FORALL k(1<=k<=n AND i != k IMPL $L_k = {}'L_k$)
AND $G = F_{i,j}(\,'G\,)$

oder etwas informeller und um so mühsamer zu schreiben (achten Sie besonders auf die Anführungsstriche) :

aktion$_{\text{prozeßnummer, adreßnummer}}$:
WHEN 'Befehlszähler$_{\text{prozeßnummer}}$ = adreßnummer
AND Ausführbarkeitsbedingung$_{\text{prozeßnummer, adreßnummer}}$('Variablen)
DO Befehlszähler$_{\text{prozeßnummer}}$ =
Nachfolgeaktion$_{\text{prozeßnummer, adreßnummer}}$ ('Variablen)
AND **FORALL** k(1<=k<=n **AND** k != prozeßnummer
IMPL Befehlszähler$_k$ = 'Befehlszähler$_k$)
AND Variablen =
Variablenänderung$_{\text{prozeßnummer,adreßnummer}}$('Variablen)

Was ist die Zustandsmenge?

$D = (D_1 \times D_2 \times \ldots \times D_n \times H)$ ist die Zustandsmenge eines Prozeßsystems. Dabei sind D_1bis D_n die Befehlszählermengen der einzelnen Prozesse, H ist die Menge der möglichen Zustandsinformationen der Modulinstanzen.

Was ist die Befehlszählerfunktion *bz(a)* ?

Sie ermittelt den Prozeß, zu dem die betreffende Aktion gehört.

Was ist die Adreßfunktion *adr(a)* ?

Sie ermittelt die Adresse der betreffenden Aktion innerhalb des Prozesses.

Der Anfangszustand *w* ist ein ausgezeichnetes Element der Menge D, das die Anfangsbefehlszähler der Prozesse und den Startzustand der Variablen enthält.

Lösungsvorschlag Aufgabe 2: Prozeßsysteme

a) P = (A, D, bz, adr, w) mit

D = ({1,2,3,4,5,6,7} x {1,2,3,4,5,6,7} x IN_0^3)

mit einem typischen Element (L_1, L_2, n_frei, n_belegt, mutex)

w = (1, 1, 5, 0, 1)

A = {

/* **Erzeuger :** */

$a_{1,1}$: WHEN 'L_1=1 DO L_1=2
AND *erzeugen*;

$a_{1,2}$: WHEN 'L_1=2 AND 'n_frei > 0 DO L_1=3
AND n_frei = 'n_frei - 1;

$a_{1,3}$: WHEN 'L_1=3 AND 'mutex > 0 DO L_1=4
AND mutex = 'mutex - 1;

$a_{1,4}$: WHEN 'L_1=4 DO L_1=5
AND *eintragen*;

$a_{1,5}$: WHEN 'L_1=5 DO L_1=6
AND mutex = 'mutex + 1;

$a_{1,6}$: WHEN 'L_1=6 DO L_1=7
AND n_belegt = 'n_belegt + 1;

$a_{1,7}$: WHEN 'L_1=7 DO L_1=1;

/* **Verbraucher :** */

$a_{2,1}$: WHEN 'L_2=1 AND 'mutex > 0 DO L_2=2
AND mutex = 'mutex - 1;

$a_{2,2}$: WHEN 'L_2=2 AND 'n_belegt > 0 DO L_2=3
AND n_belegt = 'n_belegt - 1;

$a_{2,3}$: WHEN 'L_2=3 DO L_2=4
AND *entnehmen*;

$a_{2,4}$: WHEN 'L_2=4 DO L_2=5
AND mutex = 'mutex + 1;

$a_{2,5}$: WHEN 'L_2=5 DO L_2=6
AND n_frei = 'n_frei + 1;

$a_{2,6}$: WHEN 'L_2=6 DO L_2=7
AND *verbrauchen*;

$a_{2,7}$: WHEN 'L_2=7 DO L_2=1;

}

b)

	$val_P(.)$								
ausgeführte Aktion	L_1	L_2	n_frei	n_belegt	mutex	bef_P		lf_P	
last(α)	4	1	4	0	0	$a_{1,4}$	$a_{2,1}$	$a_{1,4}$	
$a_{1,4}$	5	1	4	0	0	$a_{1,5}$	$a_{2,1}$	$a_{1,5}$	
$a_{1,5}$	6	1	4	0	1	$a_{1,6}$	$a_{2,1}$	$a_{1,6}$	$a_{2,1}$
$a_{2,1}$	6	2	4	0	0	$a_{1,6}$	$a_{2,2}$	$a_{1,6}$	
$a_{1,6}$	7	2	4	1	0	$a_{1,7}$	$a_{2,2}$	$a_{1,7}$	$a_{2,2}$
$a_{1,7}$	1	2	4	1	0	$a_{1,1}$	$a_{2,2}$	$a_{1,1}$	$a_{2,2}$
$a_{2,2}$	1	3	4	0	0	$a_{1,1}$	$a_{2,3}$	$a_{1,1}$	$a_{2,3}$
$a_{2,3}$	1	4	4	0	0	$a_{1,1}$	$a_{2,4}$	$a_{1,1}$	$a_{2,4}$
$a_{2,4}$	1	5	4	0	1	$a_{1,1}$	$a_{2,5}$	$a_{1,1}$	$a_{2,5}$
$a_{1,1}$	2	5	4	0	1	$a_{1,2}$	$a_{2,5}$	$a_{1,2}$	$a_{2,5}$
$a_{1,2}$	3	5	3	0	1	$a_{1,3}$	$a_{2,5}$	$a_{1,3}$	$a_{2,5}$
last(α·β)	3	5	3	0	1				

c) β ist eine aktive Aktionenfolge, da jede Aktion in β ausführbar ist.

d) i) $\gamma = a_{1,1}$ weil $a_{1,1} \notin bef(\alpha\beta)$

$\alpha\cdot\beta\cdot\gamma$ ist nicht semiaktiv

$\gamma = a_{2,5}\, a_{2,6}\, a_{2,7}\, a_{2,1}\, a_{1,3}$ weil $a_{1,3} \notin lf(\alpha\beta_4\gamma)$

$\alpha\cdot\beta\cdot\gamma$ ist semiaktiv

ii) $\delta = a_{2,5}\, a_{2,6}\, a_{2,7}\, a_{2,1}$

$\alpha\cdot\beta\cdot\delta$ ist verklemmt, weil der Verbraucher (im kritischen Abschnitt!) auf das Füllen des Puffers wartet, während der Erzeuger den kritischen Abschnitt nicht betreten kann.

e) $a_{2,1}$ und $a_{2,2}$ müssen vertauscht werden. Ein kritischer Abschnitt darf nur dann betreten werden, wenn sichergestellt ist, daß der Prozeß sich in diesem Abschnitt nicht blockiert.