1.2 Ablaufpläne, Präzisierung des Begriffs "Implementation"

Definition 1.6

Sei P ein Prozeßsystem. Unter einem *Ablaufplan* wird ein Prädikat S auf der Menge der Abläufe von P verstanden.
Die Menge der unter einem Ablaufplan S tatsächlich möglichen Abläufe ergibt sich dann als die Menge aller aktiven Abläufe in P, die S erfüllen.

Definition 1.7

Unter einer Abarbeitung nach statischen Prioritäten wird der Ablaufplan

$$\mathrm{SPR}(\alpha) = \forall i(1 \le i \le \mathrm{Länge}(\alpha) \Rightarrow (\mathrm{bz}(\alpha_i) = \min\{\mathrm{bz}(a) | a \in \mathrm{lf}(_{i-1}\alpha)\}))$$

verstanden.

Satz 1.3

Sei P ein Prozeßsystem mit den Prozessen $P_1, P_2, ..., P_n$.
Dann existiert zu jeder Folge $n_1, n_2, ..., n_k$ von Elementen der Menge $\{1, 2, ..., n\}$ höchstens ein aktiver Ablauf mit $\mathrm{bz}(\alpha_i) = n_i$ für $1 \le i \le k$.

Definition 1.8

Ein Ablauf heißt semiaktiv, wenn $\alpha_i \in \mathrm{bef}_P(_{i-1}\alpha)$ ist für alle i mit $1 \le i \le \mathrm{Länge}(\alpha)$.

Definition 1.9

Die Menge $bl_P(\alpha)$ sei für beliebige Abläufe α und Prozeßsysteme P induktiv definiert durch

1. $bl_P(\lambda) = \emptyset$,
2. $bl_P(\alpha \bullet a) = bl_P(\alpha) \cup \{a\}$, falls $a \notin lf_P(\alpha)$,
3. $bl_P(\alpha \bullet a) = bl_P(\alpha) - \{a\}$, falls $a \in lf_P(\alpha)$.

Ein Ablaufplan heißt R-Ablaufplan, wenn er die Eigenschaft

$$\forall \alpha, i(\text{semiaktiv}(\alpha) \wedge 1 \leq i \leq \text{Länge}(\alpha) \wedge bl_P({}_{i-1}\alpha) \cap lf_P({}_{i-1}\alpha) \neq \emptyset \Rightarrow \alpha_i \in bl_P({}_{i-1}\alpha) \cap lf_P({}_{i-1}\alpha))$$

besitzt.

Definition 1.10

P und $\bar{P}$ seien Prozeßsysteme.

1. $\bar{P}$ ist bezüglich r eine *Implementation* von P, wenn r eine Funktion ist, die Abläufe von $\bar{P}$ so in die Menge der Abläufe von P abbildet, daß für alle (auch nicht aktiven) Abläufe $\bar{\alpha} \bullet \bar{\beta}$ von $\bar{P}$ die Beziehung $r(\bar{\alpha} \bullet \bar{\beta}) = r(\bar{\alpha}) \bullet r(\bar{\beta})$ erfüllt ist. r wird in der üblichen Weise auf die Potenzmenge von $\bar{A}^*$ ausgedehnt.
2. Sei $\bar{P}$ eine Implementation von P bezüglich r. Eine Aktion $\bar{a}$ aus $\bar{A}$ heißt *beobachtbar*, wenn $r(\bar{a}) \neq \lambda$ ist, sonst heißt sie *nicht-beobachtbar*. Aktionen $\bar{a}$ mit $r(\bar{a}) = \lambda$ werden auch als *Buchhaltungsaktionen* bezeichnet.
3. Sei $\bar{P}$ bezüglich r eine Implementation von P. Weiter sei S ein Ablaufplan für P und $\bar{S}$ ein Ablaufplan für $\bar{P}$. Dann heißt $(\bar{P}, \bar{S})$ eine *sichere* Implementation von (P, S) bezüglich r, wenn für alle Abläufe $\bar{\alpha}$ von $\bar{P}$ aus $\bar{S}(\bar{\alpha})$ die Gültigkeit von $S(r(\bar{\alpha}))$ folgt.

4. Eine Implementation $(\bar{P}, \bar{S})$ von (P, S) bezüglich r heißt *vollständig*, wenn

   $$r(\{\bar{\alpha} \mid \bar{\alpha} \in \bar{A}^* \wedge \bar{S}(\bar{\alpha})\}) \supseteq \{\beta \mid \beta \in A^* \wedge S(\beta)\}$$ erfüllt ist.

5. Wenn $(\bar{P}, \bar{S})$ eine (sichere und/oder vollständige) Implementation von (P, S) bezüglich r ist, so heißt (P, S) eine *Abstraktion* von $(\bar{P}, \bar{S})$ bezüglich r.

6. Eine Implementation $\bar{P}$ von P bezüglich r heißt *vollständig*, wenn $(\bar{P}, \mathrm{aktiv}_{\bar{P}}(\bar{\alpha}))$ eine vollständige Implementation von $(P, \mathrm{aktiv}_P(\alpha))$ bezüglich r ist.

7. Eine Implementation $\bar{P}$ von P bezüglich r heißt *sicher*, wenn $(\bar{P}, \mathrm{aktiv}_{\bar{P}}(\bar{\alpha}))$ bezüglich r eine sichere Implementation von $(P, \mathrm{aktiv}_P(\alpha))$ ist.

Definition 1.11

Eine Implementation $\bar{P}$ von P bezüglich r wird *verklemmungsfrei* genannt, wenn $(\bar{P}, \mathrm{lf}(\bar{\alpha}) = \emptyset)$ bezüglich r eine sichere Implementation von $(P, \mathrm{lf}(\alpha) = \emptyset)$ ist.

Beispiel: Die fünf Philosophen

$a_{i,1}$: denken	$\leftarrow$	$\bar{a}_{i,1}$: denken
$a_{i,2}$: P(g_{i-1}, g_i)	$\leftarrow$	$\bar{a}_{i,2}$: P(g_{i-1})
		$\bar{a}_{i,3}$: P(g_i)
$a_{i,3}$: speisen	$\leftarrow$	$\bar{a}_{i,4}$: speisen
$a_{i,4}$: V(g_{i-1}, g_i)	$\leftarrow$	$\bar{a}_{i,5}$: V(g_{i-1})
		$\bar{a}_{i,6}$: V(g_i)
$a_{i,5}$: goto 1	$\leftarrow$	$\bar{a}_{i,7}$: goto 1
	r	

vollständig?

$a_{i,1}$ ersetzen durch $\bar{a}_{i,1}$
$a_{i,2}$ ersetzen durch $\bar{a}_{i,2}\ \bar{a}_{i,3}$
$a_{i,3}$ ersetzen durch $\bar{a}_{i,4}$
$a_{i,4}$ ersetzen durch $\bar{a}_{i,5}\ \bar{a}_{i,6}$
$a_{i,5}$ ersetzen durch $\bar{a}_{i,7}$

sicher?

aktiv:

$\bar{a}_{0,1}$	$\bar{a}_{0,2}$	$\bar{a}_{0,3}$	$\bar{a}_{0,4}$	$\bar{a}_{0,5}$	$\bar{a}_{4,1}$	$\bar{a}_{4,2}$	$\bar{a}_{4,3}$	$\bar{a}_{4,4}$	$\bar{a}_{4,6}$
$\downarrow$		$\downarrow$	$\downarrow$		$\downarrow$		$\downarrow$	$\downarrow$	
$a_{0,1}$		$a_{0,2}$	$a_{0,3}$		$a_{4,1}$		$a_{4,2}$	$a_{4,3}$	

unzulässig

Definition 1.12

Das Prozeßsystem $\bar{P}$ sei bezüglich r eine Implementation von P.

1. Die Implementation heißt *verklemmungskonsistent*, wenn sie die Eigenschaft

$$\forall \bar{\alpha}(\bar{\alpha} \in \bar{A}^* \wedge \text{ aktiv}(\bar{\alpha}) \Rightarrow (\forall \bar{\beta}(\bar{\beta} \in \bar{A}^* \wedge \text{aktiv}(\bar{\alpha} \bullet \bar{\beta}) \Rightarrow r(\bar{\beta})=\lambda) \Rightarrow (\text{lf}(r(\bar{\alpha})) = \emptyset)))$$

$$\wedge \exists c\Big(c \in \mathbb{N} \wedge \Big(\forall \bar{\alpha}\Big(\Big(\bar{\alpha} \in \bar{A}^* \wedge \text{aktiv}(\bar{\alpha}) \wedge \text{lf}(r(\bar{\alpha})) = \emptyset\Big)$$

$$\Rightarrow \forall \bar{\beta}(\bar{\beta} \in \bar{A}^* \wedge \text{aktiv}(\bar{\alpha} \bullet \bar{\beta}) \Rightarrow r(\bar{\beta})=\lambda \wedge \text{Länge}(\bar{\beta}) < c)\Big)\Big)\Big)$$

   besitzt.

2. Die Implementation heißt *prompt*, wenn sie die Eigenschaft

$$\exists c(c \in \mathbb{N} \wedge \forall \bar{\alpha}, \bar{\beta}(\bar{\alpha}, \bar{\beta} \in \bar{A}^* \wedge \text{aktiv}(\bar{\alpha} \bullet \bar{\beta}) \wedge r(\bar{\beta})=\lambda \Rightarrow \text{Länge}(\bar{\beta}) < c))$$

   besitzt.

3. Die Implementation heißt *prozeßtreu*, wenn sie die Eigenschaft

$$\forall \bar{a}, \bar{b}(\bar{a}, \bar{b} \in \bar{A} \wedge r(\bar{a}) \neq \lambda \wedge r(\bar{b}) \neq \lambda \Rightarrow (\text{proc}_{\bar{P}}(\bar{a}, \bar{b}) \Leftrightarrow \text{proc}_P(r(\bar{a}), r(\bar{b}))))$$

   besitzt.

4. Die Implementation heißt *schwach prozeßtreu*, wenn sie die Eigenschaft

$$\forall \bar{a}, \bar{b}((\bar{a}, \bar{b} \in \bar{A} \Rightarrow ((r(\bar{a}) = r(\bar{b}) \neq \lambda) \Rightarrow \text{proc}_{\bar{P}}(\bar{a}, \bar{b}))$$

$$\wedge (\text{proc}_{\bar{P}}(\bar{a}, \bar{b}) \wedge r(\bar{a}) \neq \lambda \wedge r(\bar{b}) \neq \lambda \Rightarrow \text{proc}_P(r(\bar{a}), r(\bar{b})))))$$

   besitzt.

5. Die Implementation heißt *behinderungsfrei*, wenn sie die Eigenschaft

$$\forall \bar{\alpha}, \bar{P}'(\bar{\alpha} \in \bar{A}^* \wedge \bar{P}' \text{ Teilprozeß von } \bar{P} \wedge (\text{lf}_{\bar{P}}(\bar{\alpha}) \cap \bar{A}' = \emptyset) \Rightarrow (\text{lf}_P(r(\bar{\alpha})) \cap r(\bar{A}') = \emptyset))$$

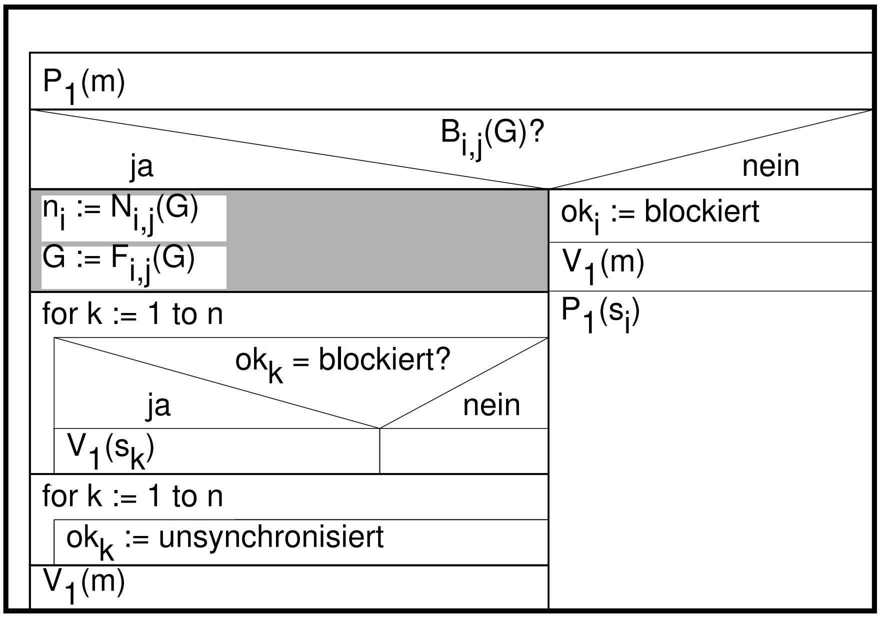
   besitzt.

Prompte und verklemmungskonsistente Implementation mit binären Semaphoren

- Aktionen der Form

$$a_{i,j}\text{: WHEN } L_i = j \wedge B_{i,j}(G) \text{ DO } L_i = N_{i,j}('G) \wedge G = F_{i,j}('G)$$

werden ersetzt durch



- Aktionen der Form

$$a_{i,j}\text{: WHEN } L_i = j \text{ DO } L_i = N_{i,j}('G) \wedge G = F_{i,j}('G)$$

werden ersetzt durch