1.2 Ablaufpläne, Präzisierung des Begrifss “Implementation"

D1.6 **Definition: Ablaufplan**

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.

D1.7 **Definition: Spezielle Ablaufpläne**

Unter einer Abarbeitung nach statischen Prioritäten wird der Ablaufplan

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

verstanden.

Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann
Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors unzulässig

S1.3 **Satz**

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 $bz(\alpha_i) = n_i$ für $1 \le i \le k$.

D1.8 **Definition**

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

Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann
Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors unzulässig

D1.9 Definition

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

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

besitzt.

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors unzulässig

D1.10 Definition: Implementation

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. Wheiter 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.

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors unzulässig

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

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

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 $(\bar{P}, \text{aktiv}_{\bar{P}}(\bar{\alpha}))$ eine vollständige Implementation von $(P, \text{aktiv}_P(\alpha))$ bezüglich r ist.

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

D1.11 Definition: Verklemmungsfrei

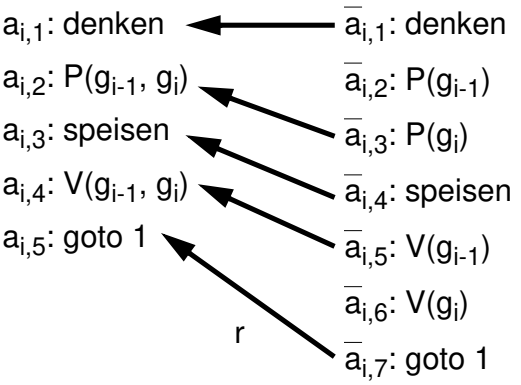
Eine Implementation $\bar{P}$ von P bezüglich r wird *verklemmungsfrei* genannt, wenn $(\bar{P}, \text{lf}(\bar{\alpha}) = \varnothing)$ bezüglich r eine sichere Implementation von $(P, \text{lf}(\alpha) = \varnothing)$ ist.

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors unzulässig

❑ Beispiel: Die fünf Philosophen

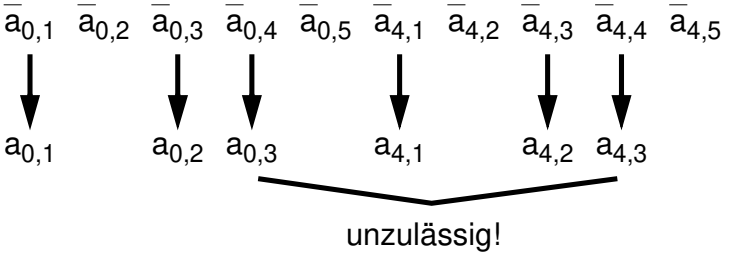
Vollständig?

$a_{i,1}$ 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:



unzulässig!

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors unzulässig

D1.12 Definition: Verklemmungskonsistent, prompt, prozeßtreu, behinderungsfrei

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

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

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

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

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

3

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.

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors unzulässig

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

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

besitzt.

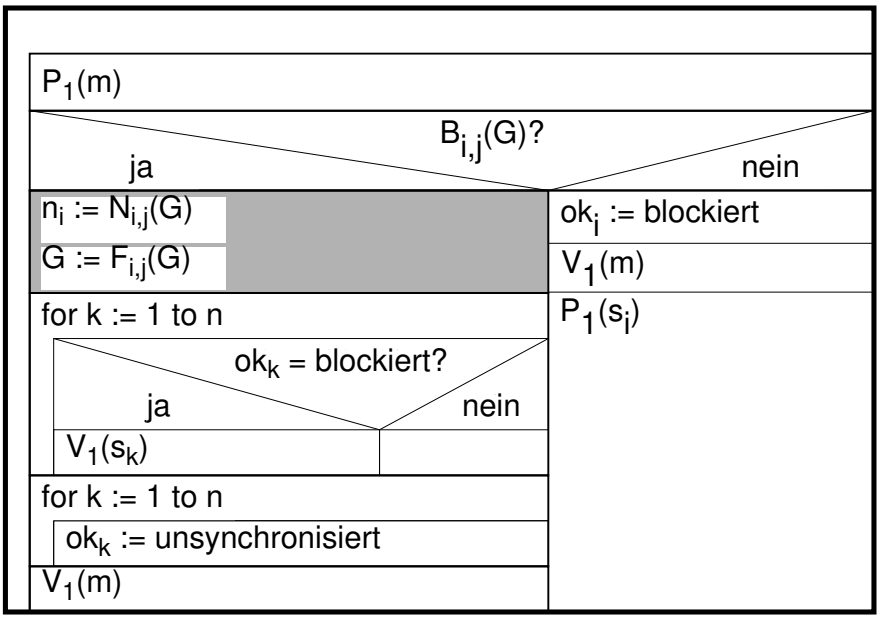
Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors unzulässig

❑ Prompte und verklemmungskonsistente Implementation mit binären Semaphoren

- Aktionen der Form

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

  werden ersetzt durch

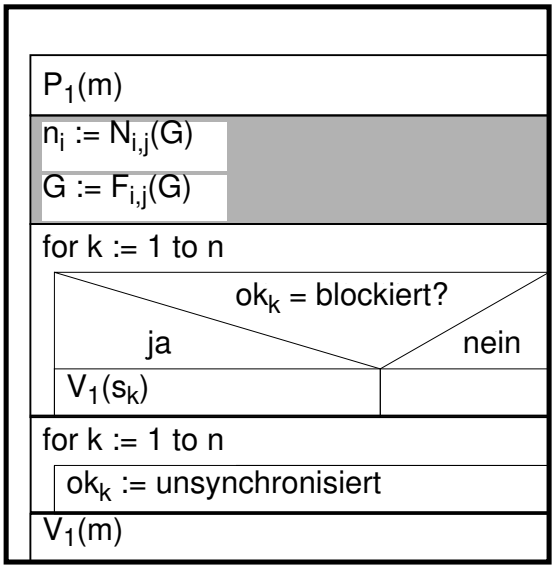


Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors unzulässig

- Aktionen der Form

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

  werden ersetzt durch



Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors unzulässig