

## Temporale Logik und Zustandssysteme

### Aufgabe 13-1

### Event-Regel

$$\begin{array}{l}
 \text{(event)} \quad A \text{ invof } Act_{\Gamma} \setminus Act_h \quad (\text{mit } Act_h = \{\lambda_1, \dots, \lambda_m\} \subseteq Act_{\Gamma}) \\
 \text{exec } \lambda \wedge A \rightarrow \circ B \quad \text{für alle } \lambda \in Act_h \\
 \square A \rightarrow \diamond(\text{enabled}_{\lambda_1} \vee \dots \vee \text{enabled}_{\lambda_m}) \\
 \vdash A \rightarrow \diamond B
 \end{array}$$

Die Regel erlaubt den Beweis von Lebendigkeitseigenschaften auf der Basis von Fairnessannahmen, also ohne Verwendung fundierter Ordnungen. Wir leiten aus den Annahmen die Formel  $A \wedge \square \neg B \rightarrow \diamond B$  her; die Idee ist, dass  $A$  immer gelten muss und keine der "hilfreichen Aktionen" in  $Act_h$  ausgeführt werden kann, wenn  $B$  nie zutrifft. Wir zeigen zunächst  $\neg B \vdash A \rightarrow \diamond B$ .

- (1)  $\neg B$  (Ann.)
- (2)  $A \text{ invof } \lambda$  für alle  $\lambda \in Act_{\Gamma} \setminus Act_h$  (Ann.)
- (3)  $\text{exec } \lambda \wedge A \rightarrow \circ B$  für alle  $\lambda \in Act_h$  (Ann.)
- (4)  $\square A \rightarrow \diamond \bigvee_{i=1}^m \text{enabled}_{\lambda_i}$  (Ann.)
- (5)  $\neg \circ B$  (nex)(1)(T1)
- (6)  $A \wedge \text{exec } \lambda \rightarrow \circ A$  für alle  $\lambda \in Act_h$  (3)(5)
- (7)  $\text{nil}_{\Gamma} \wedge A \rightarrow A$  (taut)
- (8)  $A \rightarrow \circ A$  (trans)(2)(6)(7)
- (9)  $A \rightarrow \square A$  (ind1)(8)
- (10)  $\square A \rightarrow \square \diamond \bigvee_{i=1}^m \text{enabled}_{\lambda_i}$  (4)(T30)
- (11)  $\square A \rightarrow \bigvee_{i=1}^m \square \diamond \text{enabled}_{\lambda_i}$  (10)(T32)
- (12)  $A \rightarrow \bigvee_{i=1}^m \square \diamond \text{enabled}_{\lambda_i}$  (9)(11)
- (13)  $A \rightarrow \bigvee_{i=1}^m \diamond \text{exec } \lambda_i$  (fair)(12)
- (14)  $A \rightarrow \bigvee_{i=1}^m \diamond(A \wedge \text{exec } \lambda_i)$  (9)(13)(T29)
- (15)  $A \rightarrow \diamond \circ B$  (14)(3)(T26)
- (16)  $A \rightarrow \diamond B$  (15)(T13)(T24)

Mit dem Deduktionstheorem (anwendbar, weil die Regel (init) nicht verwendet wurde) folgt

$$\vdash A \wedge \square \neg B \rightarrow \diamond B$$

und daraus folgt aussagenlogisch die Behauptung

$$\vdash A \rightarrow \diamond B$$

Man kann (event) stattdessen auch als Spezialfall der Regel (well) her, wobei die Sorte  $wf$  beliebig und  $\preceq$  durch  $=$  interpretiert wird (dies ist offensichtlich eine fundierte Ordnung).

### Aufgabe 13-2

### Aussagen über PAR-Programme

In dieser Aufgabe beweisen wir einige einfache Konsequenzen der Programmmaxiome, die bei der Programmverifikation ständig gebraucht werden.

(at- $\Pi_i$ ): Die  $i$ -te Komponente wird nie verlassen.

Es sei  $\mathcal{A}_{\Pi_i} = \{\lambda_0^{(i)}, \dots, \lambda_n^{(i)}\}$ . (NB: die Syntax von PAR garantiert, dass  $\mathcal{A}_{\Pi_i}$  nichtleer und endlich ist.) Je nach Art der Anweisung, die mit  $\lambda_j^{(i)}$  markiert ist, trifft Axiom (K1) oder (K2) zu, und wir können folgern  $\text{exec } \lambda_j^{(i)} \rightarrow \circ \text{at } \mathcal{A}_{\Pi_i}$ , da keine „Sprünge in andere Komponenten“ erfolgen.

- |      |   |                 |
|------|---|-----------------|
| (1)  | $\mathbf{start}_{\Pi} \rightarrow \text{at } \lambda_0^{(i)}$   | (taut)          |
| (2)  | $\text{at } \lambda_0^{(i)} \rightarrow \text{at } \mathcal{A}_{\Pi_i}$   | (taut)          |
| (3)  | $\text{exec } \lambda_j^{(i)} \rightarrow \text{Oat } \mathcal{A}_{\Pi_i} \quad (j = 0, \dots, n)$                                    | (K1) bzw. (K2)  |
| (4)  | $\text{at } \lambda_j^{(i)} \wedge \neg \text{exec } \lambda_j^{(i)} \rightarrow \text{Oat } \lambda_j^{(i)} \quad (j = 0, \dots, n)$ | (K3)            |
| (5)  | $\text{Oat } \lambda_j^{(i)} \rightarrow \text{Oat } \mathcal{A}_{\Pi_i} \quad (j = 0, \dots, n)$                                     | (taut)(T25)     |
| (6)  | $\text{at } \lambda_j^{(i)} \rightarrow \text{Oat } \mathcal{A}_{\Pi_i} \quad (j = 0, \dots, n)$                                      | (3)(5)          |
| (7)  | $\text{at } \mathcal{A}_{\Pi_i} \rightarrow \bigvee_{j=0}^n \text{at } \lambda_j^{(i)}$   | (taut)          |
| (8)  | $\text{at } \mathcal{A}_{\Pi_i} \rightarrow \text{Oat } \mathcal{A}_{\Pi_i}$  | (6)(7)          |
| (9)  | $\text{at } \mathcal{A}_{\Pi_i} \rightarrow \Box \text{at } \mathcal{A}_{\Pi_i}$  | (ind1)(7)       |
| (10) | $\mathbf{init} \rightarrow \Box \text{at } \mathcal{A}_{\Pi_i}$   | (1)(2)(9)(root) |
| (11) | $\text{at } \mathcal{A}_{\Pi_i}$  | (init)(10)      |

(disjoint): Anweisungen einer Komponente beeinflussen nicht den Kontrollfluss in anderen Komponenten.

- |     |   |                |
|-----|---|----------------|
| (1) | $\text{exec } \lambda \rightarrow \neg \text{exec } \mu \quad (\text{für alle } \mu \in L)$   | (I) bzw. (II)  |
| (2) | $\text{at } \mu \wedge \neg \text{exec } \mu \rightarrow \text{Oat } \mu \quad (\text{für alle } \mu \in L)$                                | (K3)           |
| (3) | $\text{exec } \lambda \rightarrow (\text{at } L \rightarrow \text{Oat } L)$   | (1)(2)         |
| (4) | $\text{exec } \lambda \rightarrow (\text{at } (\mathcal{A}_{\Pi_j} \setminus L) \rightarrow \text{Oat } (\mathcal{A}_{\Pi_j} \setminus L))$ | (genauso)      |
| (5) | $\text{at } \mathcal{A}_{\Pi_j}$  | (at- $\Pi_j$ ) |
| (6) | $\text{at } (\mathcal{A}_{\Pi_j} \setminus L) \leftrightarrow \neg \text{at } L$  | (5)            |
| (7) | $\text{exec } \lambda \rightarrow (\neg \text{at } L \rightarrow \text{O} \neg \text{at } L)$   | (4)(6)(T25)    |
| (8) | $\text{exec } \lambda \rightarrow (\text{at } L \leftrightarrow \text{Oat } L)$   | (3)(7)         |

### Aufgabe 13-3

### Klausur SS 2002

Die Beweisidee besteht darin, die Aussage

$$\text{at } \mathcal{M}_0 \rightarrow \Diamond \text{at } \alpha_0$$

(mit  $\mathcal{M}_0 = \{\alpha_0, \alpha_1, \alpha_2\}$ ) herzuleiten. Diesen Beweis zerlegen wir in die Teilbehauptungen

- |     |  |
|-----|--|
| (1) | $\text{at } \alpha_0 \rightarrow \Diamond \text{at } \alpha_0$ |
| (2) | $\text{at } \alpha_2 \rightarrow \Diamond \text{at } \alpha_0$ |
| (3) | $\text{at } \alpha_1 \rightarrow \Diamond \text{at } \alpha_2$ |

Nur die Herleitung von Aussage (3) ist nicht unmittelbar klar; die Regel (event) mit hilfreicher Aktion  $\alpha_1$  erfordert den Beweis von

$$\Box \text{at } \alpha_1 \rightarrow \Diamond(x = y)$$

Dazu beweisen wir wiederum die Aussage

$$(A) \quad \Diamond(x = y)$$

durch Herleitung von

$$\text{at } \mathcal{M}_1 \rightarrow \Diamond(x = y)$$

(mit  $\mathcal{M}_1 = \{\beta_0, \beta_1, \beta_2\}$ ); dieser Beweis ist ähnlich strukturiert wie der der Hauptaussage.

- |       |  |                  |
|-------|--|------------------|
| (A.1) | $\text{at } \beta_0 \mathbf{invof} \{\alpha_0, \alpha_1, \alpha_2\}$ | (prop)(disjoint) |
| (A.2) | $\text{at } \beta_0 \rightarrow \neg \text{at } \beta_1$             | (PC)             |
| (A.3) | $\text{at } \beta_0 \rightarrow \neg \text{exec } \beta_1$           | (A.2)(action)    |
| (A.4) | $\text{at } \beta_0 \mathbf{invof} \beta_1$                          | (A.3)            |
| (A.5) | $\text{at } \beta_0 \mathbf{invof} \beta_2$                          | (genauso)        |
| (A.6) | $\text{exec } \beta_0 \rightarrow \text{O}(x = y)$                   | (assign)         |

|        |   |                                     |
|--------|---|-------------------------------------|
| (A.7)  | $\Box \text{at } \beta_0 \rightarrow \Diamond \text{enabled}_{\beta_0}$ | (T4)(T5)                            |
| (A.8)  | $\text{at } \beta_0 \rightarrow \Diamond(x = y)$                        | (event)(A.1)–(A.7)                  |
| (A.9)  | $\text{at } \beta_1 \mathbf{invol} \{\alpha_0, \alpha_1, \alpha_2\}$    | (prop)(disjoint)                    |
| (A.10) | $\text{at } \beta_1 \mathbf{invol} \{\beta_0, \beta_2\}$                | (wie A.4, A.5)                      |
| (A.11) | $\text{exec } \beta_1 \rightarrow \circ \text{at } \beta_2$             | (K1)                                |
| (A.12) | $\Box \text{at } \beta_1 \rightarrow \Diamond \text{enabled}_{\beta_1}$ | (*)(T29)                            |
| (A.13) | $\text{at } \beta_1 \rightarrow \Diamond \text{at } \beta_2$            | (event)(A.9)–(A.12)                 |
| (A.14) | $\text{at } \beta_2 \mathbf{invol} \{\alpha_0, \alpha_1, \alpha_2\}$    | (prop)(disjoint)                    |
| (A.15) | $\text{at } \beta_2 \mathbf{invol} \{\beta_0, \beta_1\}$                | (wie A.4, A.5)                      |
| (A.16) | $\text{exec } \beta_2 \rightarrow \circ \text{at } \beta_0$             | (K1)                                |
| (A.17) | $\Box \text{at } \beta_2 \rightarrow \Diamond \text{enabled}_{\beta_2}$ | (T4)(T5)                            |
| (A.18) | $\text{at } \beta_2 \rightarrow \Diamond \text{at } \beta_0$            | (event)(A.14)–(A.17)                |
| (A.19) | $\text{at } \beta_2 \rightarrow \Diamond(x = y)$                        | ( $\Diamond \Diamond$ )(A.18)(A.8)  |
| (A.20) | $\text{at } \beta_1 \rightarrow \Diamond(x = y)$                        | ( $\Diamond \Diamond$ )(A.19)(A.13) |
| (A.21) | $\text{at } \mathcal{M}_1 \rightarrow \Diamond(x = y)$                  | (A.8)(A.19)(A.20)                   |
| (A.22) | $\text{at } \mathcal{M}_1$  | (at- $\mathcal{M}_1$ )              |
| (A.23) | $\Diamond(x = y)$   | (mp)(A.22)(A.21)                    |

Der Beweis der Behauptungen (1)–(3) folgt nun demselben Schema:

|       |   |                    |
|-------|---|--------------------|
| (1)   | $\text{at } \alpha_0 \rightarrow \Diamond \text{at } \alpha_0$            | (T5)               |
| (2.1) | $\text{at } \alpha_2 \mathbf{invol} \{\beta_0, \beta_1, \beta_2\}$        | (prop)(disjoint)   |
| (2.2) | $\text{at } \alpha_2 \mathbf{invol} \{\alpha_0, \alpha_1\}$               | (wie A.4, A.5)     |
| (2.3) | $\text{exec } \alpha_2 \rightarrow \circ \text{at } \alpha_0$             | (K1)               |
| (2.4) | $\Box \text{at } \alpha_2 \rightarrow \Diamond \text{enabled}_{\alpha_2}$ | (T4)(T5)           |
| (2.5) | $\text{at } \alpha_2 \rightarrow \Diamond \text{at } \alpha_0$            | (event)(2.1)–(2.4) |
| (3.1) | $\text{at } \alpha_1 \mathbf{invol} \{\beta_0, \beta_1, \beta_2\}$        | (prop)(disjoint)   |
| (3.2) | $\text{at } \alpha_1 \mathbf{invol} \{\alpha_1, \alpha_2\}$               | (wie A.4, A.5)     |
| (3.3) | $\text{exec } \alpha_1 \rightarrow \circ \text{at } \alpha_2$             | (K1)               |
| (3.4) | $\Box \text{at } \alpha_2 \rightarrow \Diamond \text{enabled}_{\alpha_2}$ | (A)(T29)           |
| (3.5) | $\text{at } \alpha_2 \rightarrow \Diamond \text{at } \alpha_0$            | (event)(3.1)–(3.4) |

Nun noch die Herleitung der eigentlichen Behauptung:

|     |   |                               |
|-----|---|-------------------------------|
| (4) | $\text{at } \alpha_1 \rightarrow \Diamond \text{at } \alpha_0$      | ( $\Diamond \Diamond$ )(3)(2) |
| (5) | $\text{at } \mathcal{M}_0 \rightarrow \Diamond \text{at } \alpha_0$ | (prop)(1)(2)(4)               |
| (6) | $\text{at } \mathcal{M}_0$  | (at- $\mathcal{M}_0$ )        |
| (7) | $\Diamond \text{at } \alpha_0$                                      | (mp)(6)(5)                    |
| (8) | $\Box \Diamond \text{at } \alpha_0$                                 | (alw)(7)                      |

**Besprechung:** Montag, 1.2.2004, in der Übung.