# Disjunkte parallele Programme

http://www.informatik.uni-bremen.de/theorie/teach/veri

Renate Klempien-Hinrichs

- Syntax
- Semantiken
- Exkurs: Ersetzungssysteme
- Beweissysteme

# Syntax von disjunkten parallelen Programmen



- Zwei deterministische Programme $S_1$ und $S_2$ sind disjunkt, wenn $change(S_1) \cap var(S_2) = \emptyset$ und $var(S_1) \cap change(S_2) = \emptyset$ gilt.

- Disjunkte parallele Programme werden durch die Grammatik für deterministische Programme erzeugt, wobei die folgende Regel für disjunkte parallele Komposition hinzukommt:

$$S ::= [S_1 \parallel \cdots \parallel S_n]$$

Hierbei gilt $n > 1$, und $S_1, \ldots, S_n$ sind paarweise disjunkte deterministische Programme, die (sequentiellen) Komponenten von $S$.

## Semantik von disjunkten parallelen Programmen



- Die Transitionsregeln für deterministische Programme werden um folgende Regel erweitert:

$$\text{(viii)}\quad \frac{\langle S_i\,,\ \sigma\rangle \to \langle T_i\,,\ \tau\rangle}{\langle [S_1 \parallel \cdots \parallel S_i \parallel \cdots \parallel S_n]\,,\ \sigma\rangle \to \langle [S_1 \parallel \cdots \parallel T_i \parallel \cdots \parallel S_n]\,,\ \tau\rangle}$$

wobei $i \in \{1, \ldots, n\}$

- Berechnungen, Termination und Divergenz sind wie bei deterministischen Programmen definiert; dabei gelte $[E \parallel \cdots \parallel E] \equiv E$.

- **Lemma** (keine Blockierung)

  Für jedes disjunkte parallele Programm $S \not\equiv E$ und jeden Zustand $\sigma$ gibt es eine Konfiguration $\langle S'\,,\ \sigma'\rangle$ mit $\langle S\,,\ \sigma\rangle \to \langle S'\,,\ \sigma'\rangle$.

- Die Semantiken der partiellen und totalen Korrektheit disjunkter paralleler Programme sind wie bei deterministischen Programmen definiert.

# Exkurs: Ersetzungssysteme

- Ein Ersetzungssystem $(A, \longrightarrow)$ besteht aus einer Menge $A$ und einer binären Relation $\longrightarrow$ auf $A$ (also $\longrightarrow \subseteq A \times A$).

- Die reflexive und transitive Hülle von $\longrightarrow$ wird mit $\longrightarrow^*$ bezeichnet.

- Die Relation $\longrightarrow$ hat die Diamant-Eigenschaft, wenn für alle $a, b, c \in A$ gilt:

  Falls $\begin{array}{ccc} & a & \\ \swarrow & & \searrow \\ b & & c \end{array}$ mit $b \neq c$, so gibt es ein $d \in A$ mit $\begin{array}{ccc} b & & c \\ \searrow & & \swarrow \\ & d & \end{array}$ .

- Die Relation $\longrightarrow$ ist konfluent, wenn für alle $a, b, c \in A$ gilt:

  Falls $\begin{array}{ccc} & a & \\ {}^*\swarrow & & \searrow^* \\ b & & c \end{array}$ , so gibt es ein $d \in A$ mit $\begin{array}{ccc} b & & c \\ {}_*\searrow & & \swarrow_* \\ & d & \end{array}$ .

- **Lemma** (Konfluenz)

  Wenn $\longrightarrow$ die Diamant-Eigenschaft hat, so ist $\longrightarrow$ auch konfluent.

# Normalformen



Für $a, b \in A$ ist $b$ eine Normalform von $a$, wenn gilt: $a \longrightarrow^* b$ und es gibt kein $c \in A$ mit $b \to c$.

**Lemma** (Eindeutigkeit von Normalformen)

Wenn $\longrightarrow$ konfluent ist, so besitzt jedes Element in $A$ höchstens eine Normalform.

**Lemma** (Nicht-Existenz von Normalformen)

Sei $a \in A$ so, dass es eine unendliche Folge $a \longrightarrow a_1 \longrightarrow a_2 \longrightarrow \ldots$ gibt. Wenn $\longrightarrow$ die Diamant-Eigenschaft hat, so besitzt $a$ keine Normalform.

# Das Ersetzungssystem der Transitionen 

**Lemma** (Diamant-Eigenschaft)

Sei $S$ ein disjunktes paralleles Programm. Dann gibt es für je zwei Transitionen der Form

$$\begin{tikzcd}
 & \langle S\,,\ \sigma\rangle \arrow[dl] \arrow[dr] & \\
\langle S_1\,,\ \sigma_1\rangle & \neq & \langle S_2\,,\ \sigma_2\rangle
\end{tikzcd}$$

eine Konfiguration $\langle T\,,\ \tau\rangle$ mit

$$\begin{tikzcd}
\langle S_1\,,\ \sigma_1\rangle \arrow[dr] & & \langle S_2\,,\ \sigma_2\rangle \arrow[dl] \\
 & \langle T\,,\ \tau\rangle. &
\end{tikzcd}$$

# Determiniertheit der Semantiken



**Lemma** (Determiniertheit)

Für jedes disjunkte parallele Programm $S$ und jeden Zustand $\sigma$ besitzt $\mathcal{M}[\![S]\!](\sigma)$ höchstens ein Element und es gilt:

$$\mathcal{M}_{tot}[\![S]\!](\sigma) = \{\bot\} \qquad \text{gdw.} \qquad \mathcal{M}[\![S]\!](\sigma) = \emptyset.$$

**Lemma** (Sequentialisierung)

Für paarweise disjunkte deterministische Programme $S_1, \ldots, S_n$ gilt

$$\mathcal{M}[\![[S_1 \| \cdots \| S_n]]\!] = \mathcal{M}[\![S_1; \ldots ;S_n]\!]$$

und

$$\mathcal{M}_{tot}[\![[S_1 \| \cdots \| S_n]]\!] = \mathcal{M}_{tot}[\![S_1; \ldots ;S_n]\!].$$

# Korrektheit von disjunkten parallelen Programmen



Partielle und totale Korrektheit von disjunkten parallelen Programmen ist wie für deterministische Programme definiert:

(1) $\models \{p\}\ S\ \{q\}$, wenn $\mathcal{M}[\![S]\!]([\![p]\!]) \subseteq [\![q]\!]$.

(2) $\models_{tot} \{p\}\ S\ \{q\}$, wenn $\mathcal{M}_{tot}[\![S]\!]([\![p]\!]) \subseteq [\![q]\!]$.

# Beweisregeln für disjunkte parallele Programme



(viii) $$\frac{\{p\}\ S_1; \ldots; S_n\ \{q\}}{\{p\}\ [S_1 \parallel \cdots \parallel S_n]\ \{q\}}$$ Sequentialisierung

(ix) $$\frac{\{p_i\}\ S_i\ \{q_i\},\ i = 1, \ldots, n}{\{\bigwedge_{i=1}^{n} p_i\}\ [S_1 \parallel \cdots \parallel S_n]\ \{\bigwedge_{i=1}^{n} q_i\}}$$ disjunkter Parallelismus

wobei $\mathit{free}(p_i, q_i) \cap \mathit{change}(S_j) = \emptyset$ für $i \neq j$

(x) $$\frac{\{p\}\ S\ \{q\}}{\{p\}\ S_0\ \{q\}}$$ Hilfsvariablen

wobei $S_0$ aus $S$ durch das Streichen von Zuweisungen $z := t$ bzw. durch Ersetzen von $z := t$ durch $\mathit{skip}$ entsteht, derart dass $z$ eine einfache Variable ist, die weder in $S_0$ noch in $\mathit{free}(q)$ vorkommt

# Beweissysteme für disjunkte parallele Programme



- **Beweissystem PP**: Regeln (i)–(vi), (ix)–(x) und A1–A5
- **Beweissystem TP**: Regeln (i)–(iv), (vi)–(vii), (ix)–(x) und A2–A5

**Satz** (Korrektheit von PP und TP)

(1) PP ist korrekt für die partielle Korrektheit von disjunkten parallelen Programmen.

(2) TP ist korrekt für die totale Korrektheit von disjunkten parallelen Programmen.