Substitution

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

Renate Klempien-Hinrichs

- Idee der Verifikation
- Syntaktische Ersetzung von Variablen
- Substitutionslemma

Idee der Verifikation

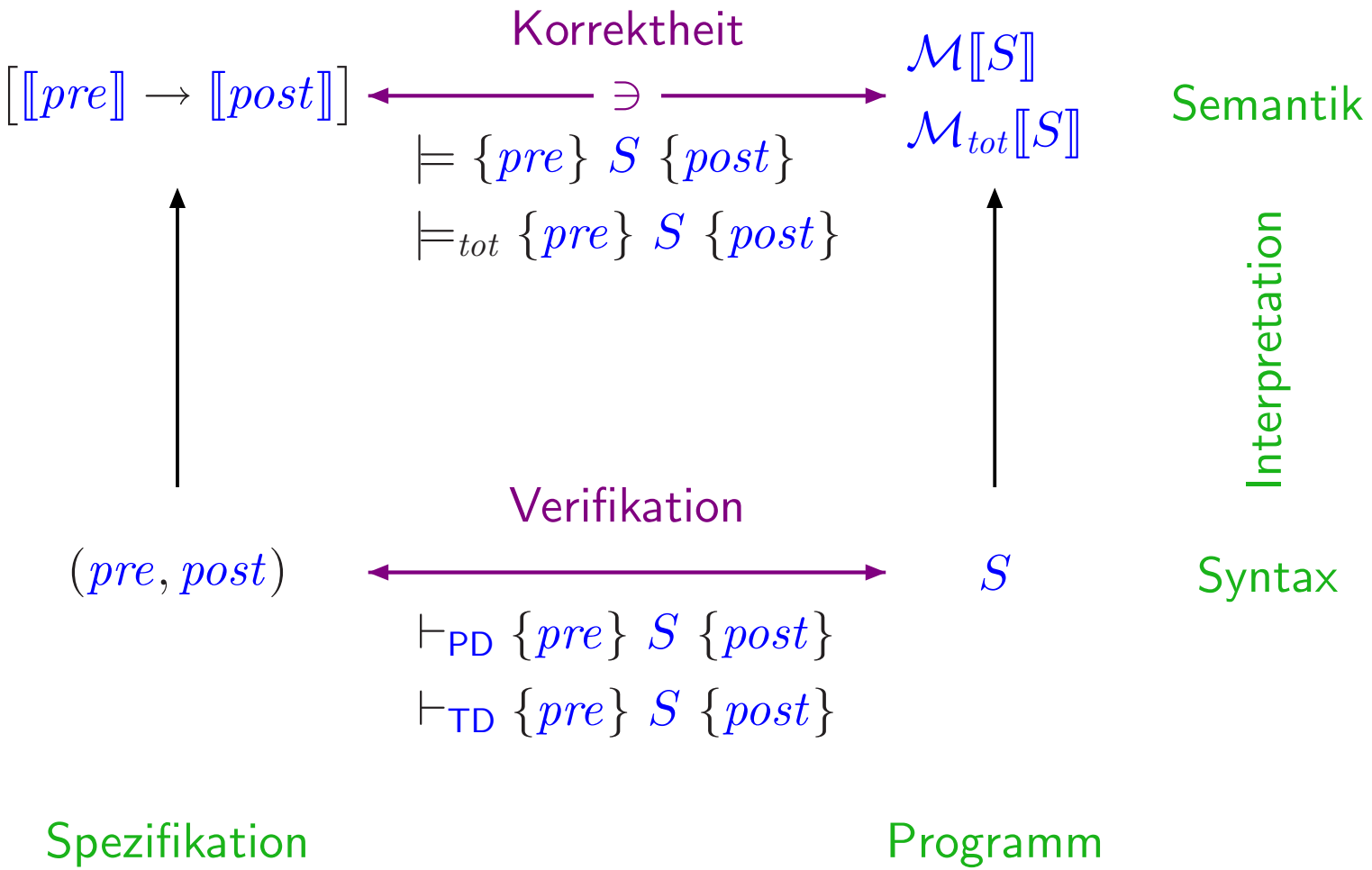
Der Beweis von partieller und totaler Korrektheit
mit Hilfe der Semantik ist mühsam!

▷ Idee von Hoare$^{(*)}$:

Korrektheit direkt auf der Ebene von Formeln beweisen,
mit Hilfe eines syntaxorientierten Beweissystems.

$^{(*)}$ C.A.R. Hoare: An axiomatic basis for computer programming.
Communications of the ACM 12 : 576-583, 1969.

Korrektheit und Verifikation



Substitution: Ersetzung von Variablen

Sei x einfache oder indizierte Variable und t Ausdruck vom selben Typ.

(1) $c[x := t] \equiv c$ für eine Konstante c von einem Basistyp.

(2) $y[x := t] \equiv \begin{cases} t & \text{falls } x \equiv y \\ y & \text{sonst} \end{cases}$ für eine einfache Variable y.

(3) $op(s_1, \ldots, s_n)[x := t] \equiv op(s_1[x := t], \ldots, s_n[x := t])$
für eine Konstante op von einem höheren Typ und Ausdrücke $s_1, \ldots, s_n$.

(4) $a[s_1, \ldots, s_n][x := t] \equiv a[s_1[x := t], \ldots, s_n[x := t]]$
für eine indizierte Variable $a[s_1, \ldots, s_n]$, falls x eine einfache Variable ist oder $x \equiv b[t_1, \ldots, t_m]$ mit $a \not\equiv b$.

(5) $a[s_1, \ldots, s_n][x := t] \equiv$

$$\textbf{if } \bigwedge_{i=1}^{n} s_i[x := t] = t_i \textbf{ then } t \textbf{ else } a[s_1[x := t], \ldots, s_n[x := t]] \textbf{ fi}$$

für eine indizierte Variable $a[s_1, \ldots, s_n]$, falls $x \equiv a[t_1, \ldots, t_n]$.

Substitution (Fortsetzung)

(6) $\textbf{if } B \textbf{ then } s_1 \textbf{ else } s_2 \textbf{ fi}\,[x := t] \equiv$
$\textbf{if } B[x := t] \textbf{ then } s_1[x := t] \textbf{ else } s_2[x := t] \textbf{ fi}$
für einen bedingten Ausdruck $\textbf{if } B \textbf{ then } s_1 \textbf{ else } s_2 \textbf{ fi}$.

(7) $(\neg p)[x := t] \equiv \neg(p[x := t])$ für eine Formel $\neg p$.

(8) $(p \wedge q)[x := t] \equiv p[x := t] \wedge q[x := t]$ für eine Formel $p \wedge q$
(und entsprechend für $\vee, \rightarrow, \leftrightarrow$).

(9) $(\forall y : p)[x := t] \equiv \forall z : (p[y := z][x := t])$ für eine Formel $\forall y : p$,
wobei z nicht in p, x, t vorkommt und y und z denselben Typ haben
(und entsprechend für $\exists$).

Substitutionslemma

Lemma (Koinzidenzlemma)

Für alle Ausdrücke t, Formeln p, Zustände σ, τ gilt:

(1) $\sigma[var(t)] = \tau[var(t)]$ impliziert $\hat{\sigma}(t) = \hat{\tau}(t)$.

(2) $\sigma[free(p)] = \tau[free(p)]$ impliziert: $\sigma \models p$ genau dann, wenn $\tau \models p$.

Lemma (Substitutionslemma)

Für alle Ausdrücke s, t, Variablen x desselben Typs wie t, Formeln p und Zustände σ gilt:

(1) $\hat{\sigma}(s[x := t]) = \hat{\sigma}[x := \hat{\sigma}(t)](s)$.

(2) $\sigma \models p[x := t]$ genau dann, wenn $\sigma[x := \hat{\sigma}(t)] \models p$.