Theorie reaktiver Systeme

Spezifikationsbasiertes Testen

Refinement

Safety

Requirement Coverage

Robustness

Hennessy test cases

Korrigierte Fassung 09.01.2007

Spezifikationsbasiertes Testen

Gegeben: Spezifikation P (abstrakt) und Implementierung Q (konkret)

Ziel: Alle Testfälle können alleine auf Grundlage der Spezifikation P konstruiert werden, um die Äquivalenz (hier: $\sim_{TE}$) zwischen P und Implementierung Q nachzuweisen.

Refinement

Eine *Verfeinerung* ist ein Transformationsschritt von einem (abstrakten) Programm (Spezifikation) $SPEC$ zu einer konkreteren Implementierung IMP.

Dabei wird i.d.R. das Verhalten auf dem gleichen Alphabet betrachtet:

$$\alpha(SPEC) = \alpha(IMP) = I.$$

(Tatsächlich gilt häufig $IMP = X \setminus (\alpha(X) - I)$.)

Safety

Safety $\hat{=}$ "Nothing bad happens" [Lamport]

Die Implementierung darf nur solche Berechnungen ausführen, die die Spezifikation auch ausführen kann.

$$traces(IMP) \subseteq traces(SPEC)$$

Trace Refinement:

$$SPEC \sqsubseteq_T IMP \Leftrightarrow traces(IMP) \subseteq traces(SPEC)$$

Anmerkung: $\forall P : P \sqsubseteq_T STOP$

Requirement Coverage: Nach einer Berechnung mit Ereignisfolge s von IMP und $SPEC$ darf IMP keine Fortsetzung verweigern, die $SPEC$ nicht verweigern kann.

Robustness: Jede Berechnung die $SPEC$ erlaubt, kann von IMP ausgeführt werden:
$traces(SPEC) \subseteq traces(IMP)$.

Safety Tests

Menge von Tests, die entscheidet, ob $traces(Q) \subseteq traces(P)$.

Tests der Form $U(s, a)$ mit $s \in traces(P)$, $a \in \alpha(P)$ und $a \notin first(P/s)$.

$$
\begin{aligned}
U_S(s, a) = \quad & \textbf{if } s = \langle\,\rangle \\
& \textbf{then } \omega \rightarrow \mathrm{STOP} \;\Box\; a \rightarrow \mathrm{STOP} \\
& \textbf{else } \omega \rightarrow \mathrm{STOP} \;\Box\; head(s) \rightarrow U_S(tail(s), a)
\end{aligned}
$$

$\mathcal{H}_{Trace}(P) = \{U_S(s,a) \mid s \in traces(P) \wedge a \in \alpha(P) \wedge a \notin first(P/s)\}$.

P $\underline{must}$ $U_S(s,a)$ für alle $U_S(s,a) \in \mathcal{H}_{Trace}(P)$.

Falls Q $\underline{must}$ $U_S(s,a)$ für alle $U_S(s,a) \in \mathcal{H}_{Trace}(P)$,
so folgt $traces(Q) \subseteq traces(P)$.

$\mathcal{H}_{Trace}(P)$ ist minimal.

Konstruktion von $\mathcal{H}_{Trace}(P)$ über *acceptance tree*.

Wurzel: P

Knoten: $\{P/s \mid s \in traces(P)\}$

Kanten: $P/s \xrightarrow{a} P/s^\frown\langle a\rangle$

– Deterministischer Graph