TH2 Übungszettel 4

Bearbeitet von Jan Strothmann, Matthies Becker,
Vitalij Kagadij & Lotte Steenbrink 8. Juni 2015

Projekt: Modellierung von Fehlern

a)

Mögliche Fehler:

1. Sensor sendet falsche Sensordaten (Fehlalarm, kein Alarm)
2. Sensor reagiert nicht auf Poll (Totalausfall, Kabeldefekt)
3. Sensor meldet Defekt
4. Sensor lässt sich nicht resetten
5. Controller fällt aus
6. Controller wertet Daten falsch aus

b)

Modellierte Fehler:

1. Da wir keine modllierte Umwelt haben, können wir ziwschen echtem und falschem Alarm nicht unterscheiden.
2. Ohne Zeitmodellierung nicht modellierbar (Deadlock)
3. Siehe Code
4. Siehe Code
5. Ohne Zeitmodellierung nicht modellierbar (Deadlock)
6. Nicht modellierbar bzw nicht sinnvoll

c)

Was können wir erkennen?

1. Nur mit Redundanzsensor
2. Nur mit Zeitmodellierung
3. Siehe Code
4. Nur mit reset ACK und Zeitmodellierung
5. Nur mit zeit UND watchdog-prozess ODER erkennung in den sensoren: zb: melder leuchten rot wenn controller nicht zurück ACKt
6. Nicht zur Laufzeit feststellbar

Projekt: Fehlertoleranz

a)

Siehe 1. c)

b)

1. Redundanz im Sensor: extra internal choice
2. Mit timeout, bei timeout Service-Warnung
3. Nachricht auf dem Kommunikationskanal -> Service-Warnung
4. Mit Ack/timeout, bei timeout Service-Warnung
5. timeout & watchdog oder Erkennung durch Sensoren, bei Fehler Service-Warnung
6. Gar nicht

Theorie: Aufgabe 1

Siehe CSP-Datei.

Theorie: Aufgabe 2

a)

Wissen laut traces-Definition für den Interupt:
$traces(Stop \bigtriangleup P) = traces(Stop) \cup \{tr_1 \frown tr_2 | tr_1 \in traces(Stop) \wedge \checkmark \notin \sigma(tr_1) \wedge tr_2 \in traces(P)\}$

$$\begin{aligned}
&\text{Unsere Werte einsetzen:} \\
&= \{<>\} \cup \{<>, P\} \\
&\text{Nach Absorbptionsgesetz:} \\
&= \{<>, P\} \\
&\text{Zusamenfassung nach prefix-Abgeschlossenheit } (<> \in \text{traces(P)}): \\
&= \{tr(p)\}
\end{aligned}$$

Somit wurde durch Umformen bewiesen, dass $Stop \bigtriangleup P =_T P$.

b)

Durch Einsetzen erhalten wir:
$\{tr | tr \in (traces(P) \cup traces(Q)) \wedge \checkmark \notin \sigma(tr)\} \cup \{tr_1 \frown tr_2 | tr_1 \frown \checkmark \in (traces(P) \cup traces(Q)) \wedge tr_2 \in traces(R)\} = (\{tr | tr \in traces(P) \wedge \checkmark \notin \sigma(tr)\} \cup \{tr_1 \frown tr_2 | tr_1 \frown \checkmark \in traces(P) \wedge tr_1 \in traces(R)\}) \cup (\{tr | tr \in traces(Q) \wedge \checkmark \in \sigma(tr)\} \cup \{tr_1 \frown tr_2 | tr_1 \frown \checkmark \in traces(Q) \wedge tr_1 \in traces(R)\})$
Formen nun die linke Seite um, bis wir die rechte Seite bekommen.
$\{tr | tr \in traces(P) \wedge \checkmark \notin \sigma(tr)\} \cup \{tr | tr \in traces(Q) \wedge \checkmark \notin \sigma(tr)\} \cup \{tr_1 \frown tr_2 | tr_1 \frown \checkmark \in (traces(P) \cup (traces(Q)) \wedge tr_2 \in traces(R)\}$
$= \{tr | tr \in traces(P) \wedge \checkmark \notin \sigma(tr)\} \cup \{tr | tr \in traces(Q) \wedge \checkmark \notin \sigma(tr)\} \cup \{tr_1 \frown tr_2 | tr_1 \frown \checkmark \in traces(P) \wedge tr_2 \in traces(R)\} \cup \{tr_1 \frown tr_2 | r_1 \frown \checkmark \in traces(Q) \wedge tr_2 \in traces(R)\}$ Dies lässt sich nun noch umsortieren, um den oberen rechten Term zu erhalten:
$= (\{tr | tr \in traces(P) \wedge \checkmark \notin \sigma(tr)\} \cup \{tr_1 \frown tr_2 | tr_1 \frown \checkmark \in traces(P) \wedge tr_1 \in traces(R)\}) \cup (\{tr | tr \in traces(Q) \wedge \checkmark \in \sigma(tr)\} \cup \{tr_1 \frown tr_2 | tr_1 \frown \checkmark \in traces(Q) \wedge tr_1 \in traces(R)\})$
Somit wurde durch Umformen bewiesen, dass die Behauptung gilt.