TH2 - Übung 2

Andreas Krohn, Benjamin Jochheim, Theodor Nolte, Benjamin Vetter

November 1, 2011

1 CSP Basis

1.1 Alphabete

$$P = (a \to b \to \mathit{Skip}) \,\Box\, (b \to d \to \mathit{Stop}) \tag{1.1}$$

$$\begin{aligned}
\alpha(P) &= \alpha(a \to b \to \mathit{Skip}) \cup \alpha(b \to d \to \mathit{Stop}) \\
&= \alpha(b \to \mathit{Skip}) \cup \{a\} \cup \alpha(d \to \mathit{Stop}) \cup \{b\} \\
&= \alpha(\mathit{Skip}) \cup \{b\} \cup \{a\} \cup \alpha(\mathit{Stop}) \cup \{d\} \cup \{b\} \\
&= \emptyset \cup \{b, a\} \cup \emptyset \cup \{b, d\} \\
&= \{a, b, d\}
\end{aligned}$$

$$Q = (x \to y \to \mathit{Stop}) \sqcap (u \to \mathit{Stop}) \tag{1.2}$$

$$\begin{aligned}
\alpha(Q) &= \alpha(x \to y \to \mathit{Stop}) \cup \alpha(u \to \mathit{Stop}) \\
&= \alpha(y \to \mathit{Stop}) \cup \{x\} \cup \alpha(\mathit{Stop}) \cup \{u\} \\
&= \alpha(\mathit{Stop}) \cup \{y\} \cup \{x\} \cup \emptyset \cup \{u\} \\
&= \emptyset \cup \{x, y, u\} \\
&= \{u, x, y\}
\end{aligned}$$

$$R = (Q;\ P) \setminus \{x, y\} \tag{1.3}$$

$$\begin{aligned}
\alpha(R) &= \alpha(Q;\ P) \setminus \{x, y\} \\
&= (\alpha(Q) \cup \alpha(P)) \setminus \{x, y\} \\
&= (\{u, x, y\} \cup \{a, b, d\}) \setminus \{x, y\} \\
&= \{a, b, d, u, x, y\} \setminus \{x, y\} \\
&= \{a, b, d, u\}
\end{aligned}$$

$$S = (P \ |[\,\{a, b\}\,|\,\{x, y\}\,]|\ R) \triangle Q \tag{1.4}$$

$$
\begin{array}{rcl}
\alpha(S) & = & \alpha(P \ |[\,\{a,b\} \,|\, \{x,y\}\,]|\ R) \cup \alpha(Q) \\
& = & \{a,b\} \cup \{x,y\} \cup \{u,x,y\} \\
& = & \{a,b,u,x,y\}
\end{array}
$$

2 CSP Modelle

2.1 System Zusammensetzen

Betrachten Sie das Fragment einer CSP-Spezifikation für eine Variante des Peterson-Algorithmus für beliebig viele Prozesse in der Datei `peterson-skeleton.csp`. Ergänzen Sie die Spezifikation durch

- a) durch passende Kanaldeklarationen für die verwendeten Kanäle
- b) durch eine Deklaration für das System `SYS` für 3 Prozesse mit der Variablen `turn` und dem Mengen-Prozess `interested`
- c) durch geeignete Prüfanweisungen, die prüfen, ob das System den wechselseitigen Ausschluss garantieren.
- d) durch geeignete Prüfanweisungen, die belegen, dass Petersons Lösung keine strict alternation realisiert.

Demonstrieren Sie die Prüfungen und die Spezifikation im Praktikum.

2.2 Debugging bei CSP

In der Datei rauchmelder-problems.csp finden Sie ein CSP Prozess-System, das einen Rauchmelder mit zwei Sensorkomponenten und einer zentralen Meldeeinheit versucht entsprechend der folgenden informellen Spezikfaiton zu modellieren:

- Die Sensorkomponenten können per `reset`-Kommando zurückgesetzt werden.
- Die Meldeeinheit fragt bei den Sensorkomponenten regelmäßig mit dem Kommando `check` an, ob Rauch detektiert wird.
- Auf diese Anfrage der Zentraleinheit liefert eine Sensorkomponente zurück, ob sie Rauch detektiert (`smoke`) oder nicht (`no smoke`).
- Nachdem eine Sensorkomponente smoke gemeldet hat, tut sie dies auf alle folgenden Anfragen bis zum Zurücksetzen.
- Die zentrale Meldeeinheit geht erst bei einer Positivmeldung von beiden Sensoren davon aus, dass eine Gefahrensituation vorliegt. Nach Empfang eines `smoke` Signals fragt sie daher bei dem zweiten Sensor an, ob dieser auch Rauch detektiert.

- In dem Fall, in dem beide Sensoreinheiten smoke melden, wird von der Meldeeinheit ein `alarm` Signal ausgelöst.
- Nachdem die Meldeeinheit auf das alarm Signal ein `ack` Signal aus der Umgebung erhalten hat, setzt sie die Sensoreinheiten zurück.
- In dem Fall, dass nur eine Sensoreinheit `smoke` meldet, wird dies als Fehler angesehen und die Komponente zurückgesetzt.

Die Spezifikation wie sie vorliegt ist fehlerhaft und zwar sowohl bzgl der Syntax, der statischen Semantik, der Modellumsetzung und der Prüfeigenschaften. Finden Sie die Fehler mit FDR und dokumentieren und korrigieren Sie sie. Bei der Dokumentation der Fehler geben Sie bitte an

- wie sie den Fehler entdeckt haben
- wie sich der Fehler bemerkbar macht
- wie er behoben wurde

Lösung s. Ausdruck

2.3 Verifikation mit Refinement

In Restaurantküchen gibt es die Vorschrift, dass zwischen der Verarbeitung von rohem und gebratenem Fleisch die Hände gewaschen werden müssen. Es wurde versucht, diese Regel in der folgenden CSP-Spezifikation zu erfassen:

$$\begin{aligned} ROH &= roh \rightarrow waschen \rightarrow ROH \\ BRATEN &= waschen \rightarrow braten \rightarrow BRATEN \\ SYS &= ROH \,|[\,\{waschen\}\,]|\, BRATEN \end{aligned}$$

Prüfen Sie ob dieser Prozess die beschriebene Regel wiedergibt mit Hilfe von CSP-Refinement, indem Sie mit einen geeigneten Prozess X eine Refinementcheck im Failuresmodell durchführt. Wie muss X aussehen und wie die Verfeinerungsrelation? Demonstrieren Sie im Praktikum die Spezifikation und die Prüfung.

$$X = (roh \rightarrow waschen \rightarrow X) \sqcap (braten \rightarrow waschen \rightarrow X)$$

$$X \sqsubseteq_F SYS$$

$x \sqsubseteq_F SYS$ ist falsch. SYS erlaubt bspw. $roh \rightarrow waschen \rightarrow braten \rightarrow roh$, was offensichtlich ein Verstoß gegen die Vorschrift ist, dass zwischen der Verarbeitung von rohem und gebratenem Fleisch die Hände gewaschen werden müssen. Bezogen auf die Failures ist zu sagen, dass X nach $roh \rightarrow waschen \rightarrow braten$ das Event roh verweigert, SYS jedoch nicht.