

Übungsblatt 1

Abgabe der Lösungen: 05.11.2018

Aufgabe 1 All the King's Media

(5 Punkte)

Gegeben sei eine Spezifikation des Ein-Platz-Puffers $Cell$ wie folgt:

$$Cell = in(x).Cell_1(x)$$

$$Cell_1(x) = \overline{out}(x).Cell$$

Geben Sie auf der Basis von $Cell$ rekursive Definitionen für die **drei** folgenden Varianten eines Zwei-Platz-Puffers an. Verwenden Sie dabei nur Aufrufe von $Cell$, ohne es neu zu implementieren.

1. LIFO (Last-in-First-out): die zuletzt eingegangenen Werte werden zuerst ausgegeben;
2. FIFO (First-in-First-out) die zuerst eingegangenen Werte werden auch zuerst ausgegeben;
3. Multimenge: Die Übertragungsreihenfolge ist un spezifiziert; Hauptsache ist: Alle eingegangenen Werte werden auf jeden Fall irgendwann weitergeleitet, und keine neue Werte werden erzeugt.

Zeichnen Sie in allen drei Fällen entsprechende beschriftete Transitionssysteme auf, unter der Voraussetzung, dass für x nur zwei Werte möglich sind: 0 und 1.

Aufgabe 2 Mutex

(5 Punkte)

Mutex (mutual exclusion) bezeichnet das Problem der Zugangsregelung zu *kritischen Abschnitten* für nebenläufige Prozesse.

Kurz gesagt geht es um zwei oder mehr nebenläufige Prozesse, die gewisse Programmabschnitte nicht gleichzeitig ausführen dürfen (wie z.B. Schreiben in den gemeinsamen Speicher). Dafür wird ein als *Semaphor* bezeichneter besonderer Prozess verwendet, der sicherstellt, dass sich zu jedem Zeitpunkt höchstens ein Prozess in einem kritischen Abschnitt befindet.

Betrachten Sie folgende Definitionen von Prozessen $User$ und Sem :

$$User = \bar{p}.enter.exit.\bar{v}.User$$

$$Sem = p.v.Sem$$

die ein Benutzerprogramm bzw. ein Semaphor repräsentieren.

1. Leiten Sie gemäß der Regeln der CCS-Semantik das LTS für den Prozess $Mutex_1 = (User \mid Sem) \setminus \{p, v\}$ her und stellen Sie es graphisch dar.
2. Verfahren Sie ebenso für den Prozess $Mutex_2 = ((User \mid Sem) \mid User) \setminus \{p, v\}$.
3. Argumentieren Sie, warum $Mutex_2$ das Mutex-Problem löst.
4. Argumentieren Sie, warum die folgende Änderung der Definition von $User$

$$User = \bar{p}.enter.\bar{v}.exit.User$$

keine korrekte Lösung des Mutex-Problems ist.

Aufgabe 3 Parallelität**(4 Punkte)**

Definiere

$$Q = a.b.Q.$$

Für $n \geq 0$ sei P_n der Prozeß

$$\underbrace{Q | \dots | Q}_{n \text{ mal}}.$$

Geben Sie die Größe des gemäß der CCS-Semantik von P_n dargestellten LTS als Funktion von n an, und beweisen Sie die Formel durch Induktion über n , mittels der Regeln der Semantik.

Aufgabe 4 Von der Semantik zur Syntax**(6 Punkte)**

Wir bezeichnen das Fragment von CCS, das nur Präfix und Summen zulässt, als *Basic Process Algebra* (BPA).

1. Konstruieren Sie zu einem gegebenen LTS T eine Prozessdefinition in BPA, deren Semantik gerade T ist.
2. Folgern Sie, dass eine semantikerhaltende (was heißt das?) Übersetzung von Prozessdefinitionen in CCS in Prozessdefinitionen in BPA existiert.
3. Zeigen Sie, dass das durch eine Prozessdefinition in BPA beschriebene LTS höchstens linear groß ist. *Hinweis:* Dies folgert man aus einer genaueren Beschreibung dieses LTS, die man per Induktion über Herleitungen in der Semantik beweist.
4. Wir bezeichnen mit CCS/BPA das Fragment von CCS, in dem als rechte Seiten von *rekursiven* Definitionen nur BPA-Terme erlaubt sind (während in nicht-rekursiven Definitionen beliebige CCS-Terme zugelassen sind). Was lässt sich nach Aufgabe 3 und den vorigen Teilaufgaben über die in schlechten Fällen mindestens benötigte Größe einer von CCS/BPA nach BPA übersetzten Prozessdefinition sagen?