

Übungen zur Vorlesung Einführung in die Informatik III

Aufgabe 12 Agenten, Abläufe

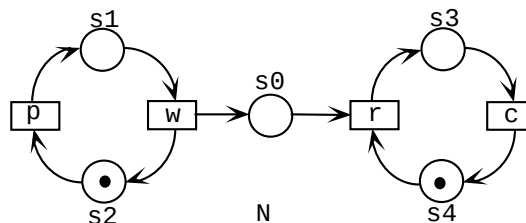
(a) Bestimmen Sie die Abläufe folgender Agenten über der Aktionenmenge $A = \{a, b, c, d\}$:

- (i) $t_0 =_{def} a; (b \text{ or } c)$
- (ii) $t_1 =_{def} a; (b \parallel c)$
- (iii) $t_2 =_{def} (x :: a; b; x) \parallel_{\{b\}} (y :: b; a; y)$
- (iv) $t_3 =_{def} (x :: a; (b \text{ or } c); x) \parallel_{\{b, c\}} (y :: (b \text{ or } c); a; y)$
- (v) $t_4 =_{def} (x :: a; ((b; c) \text{ or } (c; d)); x) \parallel_{\{b, c, d\}} (y :: a; (b \text{ or } c); (c \text{ or } d); y)$

Welche dieser Agenten sind verklemmungsfrei? Besitzen die beiden Teilagenten von t_4 dieselben Ablaufmengen?

(b) Geben Sie einen Agenten an, der den Prozess P aus Aufgabe 2 beschreibt.

Aufgabe 13 Petri-Netze, Agenten, Prozessprädikate



Gegeben sei noch einmal das obenstehende Boolesche Petri-Netz N . Dargestellt ist ein Erzeuger/Verbraucher-Problem:

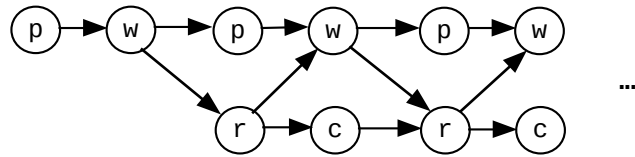
Liest man

p als produce,
w als write,
r als read,
c als consume,

dann beschreibt der linke Kreis einen Produzenten und der rechte einen Konsumenten.

Beachte: Aufgrund der Kapazität Boolescher Petrinetze ist w nicht schaltbereit, wenn s_0 mit $true$ belegt ist. Die Stelle s_0 wirkt wie ein *Puffer* der Kapazität 1.

Gegeben sei der Prozess A (siehe Skizze), ein vollständiger Ablauf von N mit der Eigenschaft, dass alle anderen vollständigen Abläufe von N Sequentialisierungen von A sind.



- Geben Sie ein natürlichzahliges Petrinetz N1 an, das ablaufäquivalent zu N ist, das insbesondere also die Puffereigenschaft von s_0 modelliert.
- Beschreiben Sie N1 durch einen Agenten t , der die selbe Menge von Abläufen besitzt, der also Erzeuger, Verbraucher und Puffer über diejenigen Ereignisse koordiniert, die mit w oder r markiert sind.
- Geben Sie ein Petrinetz N2 zu einem Erzeuger/Verbraucher-Problem mit Puffergröße 2 an. Geben Sie zu N2 einen vollständigen Ablauf A2 an. Kann auf den Puffer gleichzeitig lesend und schreibend zugegriffen werden?
- Bezeichne q_e den kleinsten Präfix eines Prozesses q , der das Ereignis e enthält, und $\#(a, q_e)$ die Anzahl der mit a markierten Ereignisse von q_e .

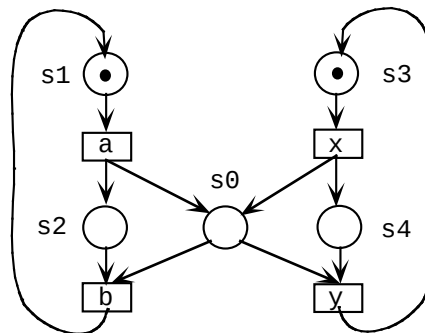
Formulieren Sie durch Prädikate die Beziehungen, die zwischen der Anzahl der mit w und der Anzahl der mit r markierten Ereignisse in q_e gelten, wenn e mit w oder mit r markiert ist, und zwar

(1) für $q=A$

(2) für $q=A2$

Aufgabe 14 (H) Petri-Netze

Vorgelegt sei folgendes Petri-Netz:



- Welche Belegungen sind von der Anfangsbelegung aus erreichbar? Geben Sie einen entsprechenden Zustandsübergangsgraphen an.
- Eine Stelle eines Booleschen Petri-Netzes heißt *blockierend*, wenn sie unter der Anfangsbelegung oder unter einer Folgebelegung mit *true* belegt und zugleich Ausgabestelle einer schaltbereiten Transition ist. Bestimmen Sie die blockierenden Stellen unseres Petri-Netzes.
- Geben Sie ein ablaufäquivalentes natürlichzahliges Petri-Netz an, das also insbesondere die Rolle der blockierenden Stellen modelliert. Lässt sich bei Ihrer Lösung eine Stelle (mit ihren Kanten) einsparen?

Aufgabe 15 (P) Erreichbarkeit von Zuständen in Petri-Netzen

In dieser Aufgabe soll der Simulator für boolesche Petri-Netze von Übungsblatt 3 so erweitert werden, dass ausgehend von einem initialen Zustand die Menge aller erreichbaren Zustände analysiert werden kann.

- a) Implementieren Sie eine Hilfsklasse `BoolState`, die es Ihnen erlaubt den Zustand des Netzes in einem booleschen Feld zu speichern, Zustände auf Äquivalenz zu überprüfen (`equals`) und einen Zustand in einen String zu konvertieren (`toString`).

– P –

Erweitern Sie die Klasse `BPetriNet` um folgende Methoden:

- b) `BoolState getState()` - Ermitteln der aktuellen Belegung
- d) `setState(BoolState)` - Setzen der Belegung des Petri-Netzes
- c) `Vector reachable()` - Analyse der erreichbaren Zustände Informelle Beschreibung:
- Die Menge E der erreichbaren Zustände ist initial leer
 - Die Menge Z der noch zu untersuchenden Zustände ist initial $\{\text{akt_Zustand}\}$
 - für jedes $z \in Z$
 - für jede schaltbereite Transition t
 - * belege Petri-Netz mit z
 - * schalte t
 - * falls Folgezustand $z' \notin Z \cup E : Z = Z \cup \{z'\}$
 - $E = E \cup \{z\}$
 - $Z = Z \setminus \{z\}$

Analysieren Sie mit Ihrem Programm alle erreichbaren Zustände für das Erzeuger-Verbraucher Problem von Übungsblatt 3.

Hinweis: Für die Realisierung von Mengen empfiehlt es sich hier die Java Klasse `Vector` aus `java.util.Vector` einzusetzen; siehe <http://java.sun.com/j2se/1.4.1/docs/api/>.