

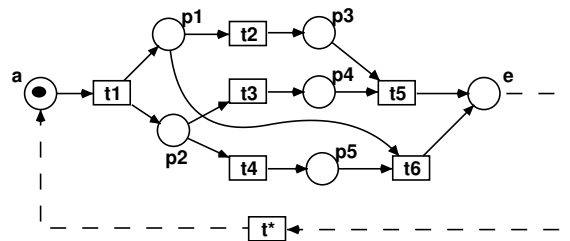
FGI-2 – Formale Grundlagen der Informatik II

Modellierung und Analyse von Informatiksystemen

Präsenzlösung 11: Workflownetze und Gefärbte Petrinetze

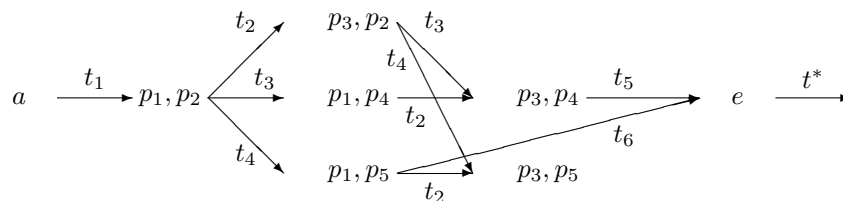
Präsenzteil am 04./05.01. – Abgabe am 11./12.01.2016

Präsenzaufgabe 11.1: Gegeben sei folgendes P/T-Netz \overline{N} . Das Netz N entsteht, indem man die Transition t^* mit den anliegenden Kanten streicht.



1. Konstruieren Sie den Erreichbarkeitsgraphen von \overline{N} .

Lösung:



2. Ist N ein Workflow-Netz? Begründen Sie!

Lösung: Ja. Kantengewicht $W = 1$, Anfangsstelle a , Endstelle e mit $\bullet a = \emptyset = e \bullet$.

3. Ist N ein korrektes Workflow-Netz? Begründen Sie!

Lösung: Nein. Schaltfolge $t_1 t_2 t_4$ führt in den Deadlock $m = p_3 + p_5$.

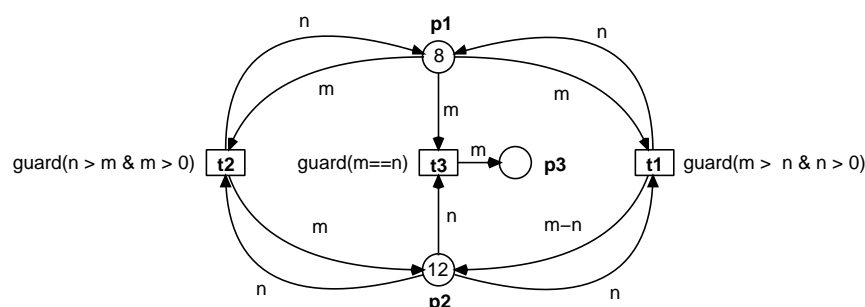
4. Zeigen Sie: In jedem Workflow-Netz ist der Anfangsplatz a eindeutig bestimmt. Ebenso der Endplatz e .

Lösung: Sei a' eine zweite Anfangsstelle. Da nach Definition jede Stelle eines WF Netzes zwischen a und e liegt, also auch a' , hätte a' mindestens eine eingehende Kante. Widerspruch. Analog für einen zweiten Endplatz e' .

5. Beweisen oder widerlegen Sie: Sei N ein korrektes Workflow-Netz und \overline{N} sein Abschluss (s. Skript Abschnitt 8.3.7). Die Initialmarkierung $m_a = a$ ist ein Rücksetzzustand (home state) für \overline{N} .

Lösung: Wenn \overline{N} korrekt ist, dann ist für $m(e) > 0$ das restliche Netz unmarkiert. Außerdem ist $m_e = e$ von jeder erreichbaren Markierung aus erreichbar. Nach dem Schalten von t^* erhalten wir die von jeder erreichbaren Markierung Startmarkierung m_a erneut. Also ist m_a ein Rücksetzzustand.

Präsenzaufgabe 11.2: Betrachten Sie das folgende gefärbte Petrinetz N mit drei Stellen p_1 , p_2 und p_3 , alle vom Typ *integer*.



1. Geben Sie eine Schaltfolge bis zur Termination (d.h. in diesem Fall: bis zur Verklemmung des Netzes) an! Gibt es eine andere?

Lösung: Folgendes ist die einzige Schaltfolge bis zur Termination:

$$(1'8, 1'12, \emptyset) \xrightarrow{t_2} (1'12, 1'8, \emptyset) \xrightarrow{t_1} (1'8, 1'4, \emptyset) \xrightarrow{t_1} (1'4, 1'4, \emptyset) \xrightarrow{t_3} (\emptyset, \emptyset, 1'4)$$

2. Geben Sie für die erste Transition in ihrer Schaltfolge die Belegung β explizit an!

Lösung: $\beta : m \mapsto 8, n \mapsto 12$

3. a) Zeigen Sie, dass das Petrinetz eine Funktion berechnet, d.h. von einer Anfangsmarkierung aus zu einer (abhängig von der Anfangsmarkierung) eindeutig bestimmten Endmarkierung kommt.

b) Um welche Funktion handelt es sich?

Erläuterung: Die Funktion $f(x, y)$ soll berechnet werden, indem die Stellen p_1 und p_2 in der Anfangsmarkierung die Argumente x und y enthalten, d.h. $m_0(p_1) = 1'x$ und $m_0(p_2) = 1'y$. Die Stelle p_3 ist initial unmarkiert. Das Netz soll dann so schalten, dass die Stellen p_1 und p_2 am Ende leer sind und die Stelle p_3 mit $f(x, y)$ markiert ist.

Lösung: Ja, die Schaltfolge ist eindeutig bestimmt. Die Transitionen entsprechen aufgrund der Guards den disjunkten Bedingungen der Fallunterscheidung. Für jede Markierung vor der Endmarkierung ist daher nur genau eine Transition aktiviert. Das Netz verklemmt dann in der gewünschten Endmarkierung.

Alternativ kann das Netz verklemmen, wenn in p_2 oder p_3 ein negativer Wert liegt (das kann aufgrund der Guards nicht während der Berechnung auftreten, sondern nur in der Anfangsmarkierung). Da in diesem Fall die Endmarkierung nicht erreicht wird, ist der Funktionswert undefiniert.

Die Funktion lautet:

$$\text{ggT}(x, y) = \begin{cases} \text{ggT}(y, x - y), & \text{falls } x > y > 0 \\ x, & \text{falls } x = y \\ \text{ggT}(y, x), & \text{falls } y > x \end{cases}$$

Übungsaufgabe 11.3: Modellierung und Analyse mit Petrinetzen

Für diese Aufgabe benötigen Sie zusätzlich zu Renew auch noch das Renew Lola Plugin. Dies können Sie unter den Ressourcen² finden.

von
6

1. Erstellen Sie mit Renew das Cellphone Modell aus Aufgabe 5.3 als P/T-Netz. Benutzen Sie dafür die Vorlage von der FGI2 Web-Seite². Vergessen Sie nicht, die Meta-Daten (@author. . .) einzutragen und das Netz (Datei) entsprechend umzubenennen. Die in der Vorlage bereits vorhandenen Stellen sollen die Prädikate (Etiketten) modellieren. Diese sollen somit also entweder eine Marke oder keine Marke enthalten. Nutzen Sie das Werkzeug für die Virtuellen Stellen um Kopien von den *Prädikatstellen* zu erzeugen. Somit kann das Netz trotz mehrmaligem Zugriff auf diese Stellen aufgeräumt sein.

Anmerkung: Virtuelle Stellen stellen keine semantische Erweiterung der P/T-Netze dar. Sie sind nur ein zeichnerisches Hilfsmittel. In der internen Darstellung (für Lola) tauchen diese nicht mehr auf. Virtuelle Stellen werden dadurch erzeugt, dass das "Virtual Place Tool" ausgewählt wird und dann auf die Originalstelle geklickt wird.

2. Prüfen Sie ihr Modell mit den in Aufgabe 5.3.1 angegebenen Formeln f und g . Die Formeln können als einfache Text-Elemente (Text Tool) zu dem Netz hinzugefügt werden und dann direkt durch Lola ausgewertet werden. Die Lola Integration in Renew liefert dafür eine einfache Benutzerunterstützung. Zum Glück kann Lola überprüfen wie groß eine Markierung ist. So kann mit $on > 0$ geprüft werden, ob sich eine Marke auf der Stelle on befindet.

Anmerkung: Die Ausgaben (Ergebnisse, Fehlermeldungen sowie weitere Informationen) werden vom Lola Plugin auf verschiedene Weise dargestellt. Beachten Sie auf jeden Fall für evtl. Fehlermeldungen (Syntaxfehler) die Kommandozeile. Dazu müssen Sie evtl. Renew von dieser mit dem Befehl `java -jar renew2.5.X/loader.jar gui` starten. Der Parser von Lola ist intolerant gegen zu wenige Leerzeichen. Alle Tokens müssen durch solche getrennt werden. So kann `error = 0` nicht korrekt geparkt werden; es muss `error = 0` heißen. Für Tasks gibt es die Signalfarben für Formeln: grün – `true`, rot – `false`, schwarz – `Syntax Error`.

Siehe auch: <http://www.paose.net/wiki/Publications>

3. Geben Sie drei weitere CTL Formeln an, fügen Sie diese dem Netz hinzu. Geben sie außerdem die Bedeutung der Formeln als sprachliche Übersetzung an.

Schicken Sie das Netz an Ihren Übungsgruppenleiter in der üblichen Form (ZIP, benannt,...).

Übungsaufgabe 11.4: Gefärbte Netze: Farben, Datenstrukturen und Berechnungen

Ähnlich wie in Präsenzaufgabe 11.2 können gefärbte Petrinetze genutzt werden um Algorithmen zu modellieren. Hier wird nun ein Algorithmus für die Summe der Knoten in einem Baum modelliert.

Erstellen Sie ein gefärbtes Petrinetz mit Renew, dass die Summe der Knoten eines Binärbaumes bestimmt. Eingabe ist ein Binärbaum in der Form von geschachtelten Tupeln (in Renew, z.B.: $[1, [2, 3]]$). Ausgabe soll ein Tupel aus dem Original-Tupel und der Anzahl der Knoten sein (ohne

von
5

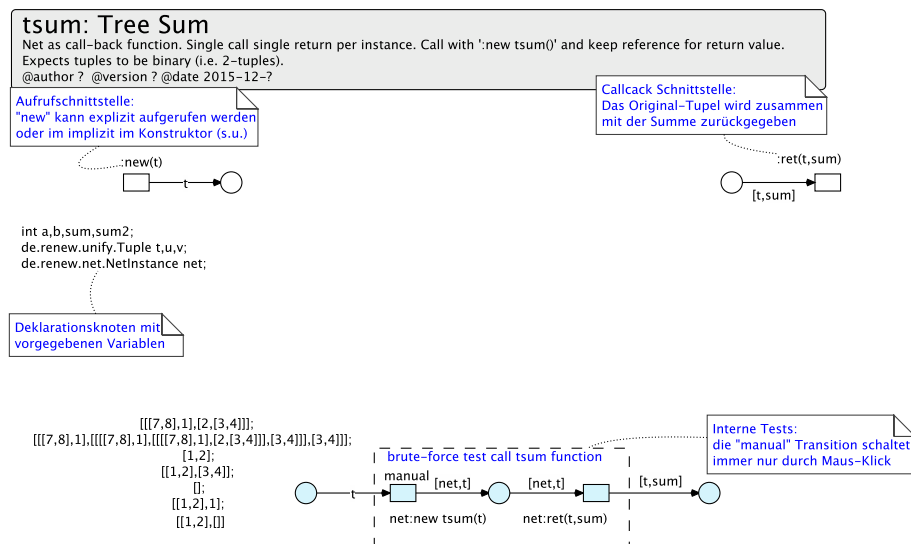
²Ressourcen: <http://www.informatik.uni-hamburg.de/TGI/F2/sec/ressourcen.html>

Blätter). Wir gehen davon aus, dass die Blätter des Baumes aus natürlichen Zahlen (*int*) bestehen und dass alle Bäume wohlgeformt sind. Die Bäume müssen allerdings nicht vollständig sein. Blätter (also Zahlen) werden aufsummiert (Beispiel: $tsum([1[2, 3]]) = 6$).

Die funktionale Schnittstelle geben wir vor. Dabei wird die Eingabe dem Netz mit dem Konstruktor (*new*) übergeben (in Renew: `tc: new tsum(tuple)`). Das Ergebnis wird dann über ein call-back Aufruf (`:ret(.,.)`) zurückgegeben (in Renew: `tc:ret(tuple,sum)`). Somit besteht die Schnittstelle aus synchronen Kanälen, wobei das anbietende Netz (*tsum*) die beiden folgenden *Up-links* implementieren muss: `:new(t)` und `:ret(t,sum)`. Die Variable *t* ist vom Typ *Tuple* und die Variable *sum* vom Typ *int*. Sie brauchen typischerweise fünf Fallunterscheidungen, die Sie durch Pattern-Matching erreichen können.

Hinweis: (Netz-)Intern können die Kanäle (:new(·) und :ret(·,·)) natürlich auch benutzt werden. Hier eignet sich das Schlüsselwort this, um die Netzinstanz referenzieren zu können (z.B.: this:new(t)).

Eine Vorlage für das Netz gibt es unter den Ressourcen² zum download. Verfahren Sie mit ihren Ergebnissen wie gewohnt (s.o.).



Übungsaufgabe 11.5: Erstellen Sie zwei neue Olat-Fragen für den aktuellen Lesestoff der 11. Woche entsprechend den bisherigen Anforderungen.

von
1

Bisher erreichbare Punktzahl: 132