Kommunikation und parallele Prozesse WiSe 2018/19

Übungsblatt 2

Abgabe der Lösungen: 19.11.18

Aufgabe 1 Trace-Äquivalenz (2 Punkte)

Überprüfen Sie, welche der folgenden Paare (P, Q) von Prozessen trace-äquivalent sind:

1. $P = a.\bar{b}.P + \tau.\emptyset$, $Q = a.\bar{b}.Q$.
2. $P = (a.P)\backslash a$, $Q = \tau.Q$.
3. $P = (a.P \mid \bar{a}.P)\backslash a$, $Q = \tau.Q$.

Begründen Sie Ihre Antwort.

Aufgabe 2 Trace-Äquivalenz (4 Punkte)

1. Zeigen Sie, dass bisimilare Prozesse trace-äquivalent sind. **Hinweis:** Verwenden Sie Induktion über die Länge der Spuren.
2. Zeigen Sie, dass die drei CCS-Prozesse aus Aufgabe 1, Übungsblatt 1 paarweise nicht bisimilar sind. **Hinweis:** Verwenden sie Teilaufgabe 1.

Aufgabe 3 Vollständige Traces (6 Punkte)

Ein *vollständiger Trace* eines Prozesses $p \in \mathsf{Proc}$ in einem LTS $(\mathsf{Proc}, \mathsf{Act}, \{\xrightarrow{\alpha}\}_{\alpha \in \mathsf{Act}})$ ist ein Wort $w \in \mathsf{Act}^*$, so dass q existiert mit $p \xrightarrow{w} q$ und $q \not\rightarrow$. Zwei Prozesse heißen *stark trace-äquivalent*, wenn sie trace-äquivalent sind und dieselben vollständigen Traces haben.

1. Die beiden Versionen CM, CM' der aus der Vorlesung bekannten Getränkemaschine sind definiert durch

$$CM = coin.\,(\overline{coffee}.\,CM + \overline{tea}.\,CM)$$
$$CM' = coin.\,\overline{coffee}.\,CM' + coin.\,\overline{tea}.\,CM'.$$

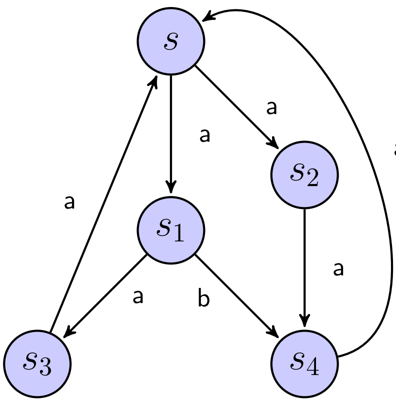
   Sind die beiden Maschinen stark trace-äquivalent?

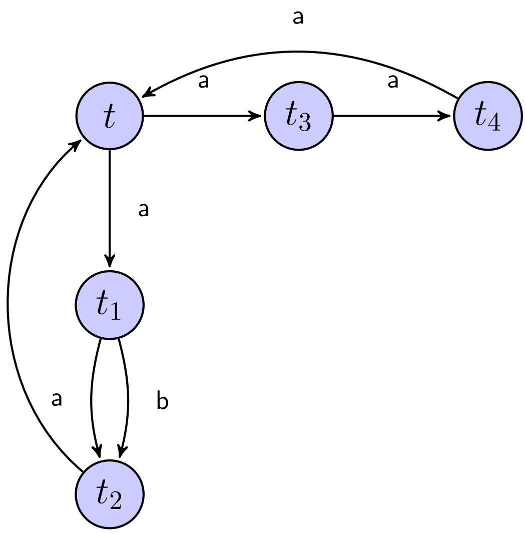
2. Ist starke Trace-Äquivalenz eine Kongruenz bezüglich $+$?
3. Ist starke Trace-Äquivalenz eine Kongruenz bezüglich $\mid$?
4. Ist starke Trace-Äquivalenz eine Kongruenz bezüglich $\backslash$?

(Es ist natürlich immer ein Beweis bzw. ein geeignetes Gegenbeispiel gefragt.)

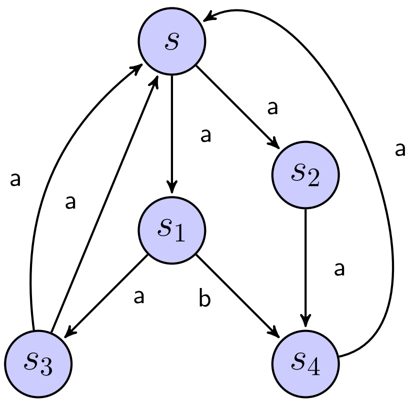
Aufgabe 4 Bisimulation und Bisimilarität (3 Punkte)

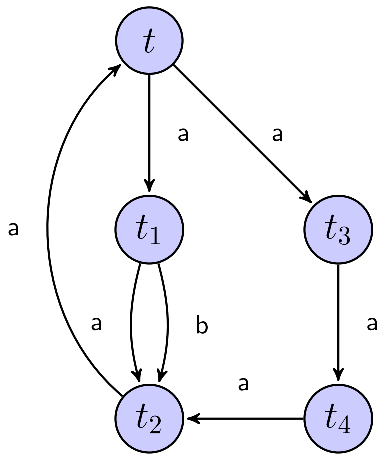
Zeigen Sie, dass für die folgenden Transitionssysteme $s \sim t$ gilt, indem sie eine Bisimulation R finden, so dass $(s, t) \in R$.





Beweisen Sie ferner, dass für die folgenden Systeme $s \sim t$ nicht gilt, d.h. dass keine Bisimulation R mit $(s, t) \in R$ existiert.





Aufgabe 5 Ready Simulation (5 Punkte)

Eine Relation $R \subseteq \mathsf{Proc} \times \mathsf{Proc}$ über einem LTS $(\mathsf{Proc}, \mathsf{Act}, (\rightarrow_\alpha))$ heißt *ready simulation* wenn

1. $s \xrightarrow{a} s'$ und $(s, t) \in R$ impliziert, dass ein t' existiert, so dass $t \xrightarrow{a} t'$ und $(s', t') \in R$, und
2. $s \not\rightarrow$ und $(s, t) \in R$ impliziert, dass $t \not\rightarrow$.

Wir schreiben $s \sqsubseteq_{RS} t$ wenn es eine ready simulation R gibt, so dass $(s, t) \in R$ und $s \rightleftharpoons t$ wenn $s \sqsubseteq_{RS} t$ und $t \sqsubseteq_{RS} s$. In letzterem Fall werden s und t *ready similar* genannt.

1. Geben Sie ein Paar von Prozessen an, die stark trace-äquivalent sind (siehe Aufgabe 3), und dennoch nicht ready similar.

Die Relation $\rightleftharpoons$ wird oft als eine Alternative zu $\sim$ betrachtet, da $\rightleftharpoons$ die feinste Äquivalenz ist, die nur solche Paare von Prozessen unterscheidet, die in allen möglichen Kontexten (wobei Kontexte noch allgemeiner als CCS-Kontexte definiert sind) die gleichen vollständigen Traces ausführen (eine genaue Formulierung dieser Tatsache benötigt gewisse Vorbereitungen und wird hier weggelassen).

Dennoch ist $\rightleftharpoons$ streng schwächer als $\sim$, wie das folgende Beispiel zeigt.

Das *Lossy two-stage link Protokoll* spezifiziert einen Prozess, der periodisch einen Eingabewert v oder ein Signal $\Box$ bekommt, wobei das letztere besagt, dass der Wert verloren gegangen ist. Man kann dieses Verhalten in CCS auf zwei Arten implementieren: der Wert wird immer erfolgreich empfangen, aber kann während einer Pause d verloren gehen; oder der Wert kann sowohl unterwegs als auch während einer Pause verloren gehen:

$$\begin{aligned} LL_1 &= \sum\nolimits_v v.(d.\bar{v}.LL_1 + d.\Box.LL_1), \\ LL_2 &= \sum\nolimits_v (v.(d.\bar{v}.LL_2 + d.\Box.LL_2) + v.d.\Box.LL_2). \end{aligned}$$

2. Beweisen Sie, dass $LL_1 \rightleftharpoons LL_2$, aber $LL_1 \not\sim LL_2$ (Das verdeutlicht einen grundlegenden Unterschied zwischen $\rightleftharpoons$ und $\sim$, der auch unter dem Slogan *"Bisimulation can't be traced"* bekannt ist.).