Theoretische Informatik 1

8. April 2013

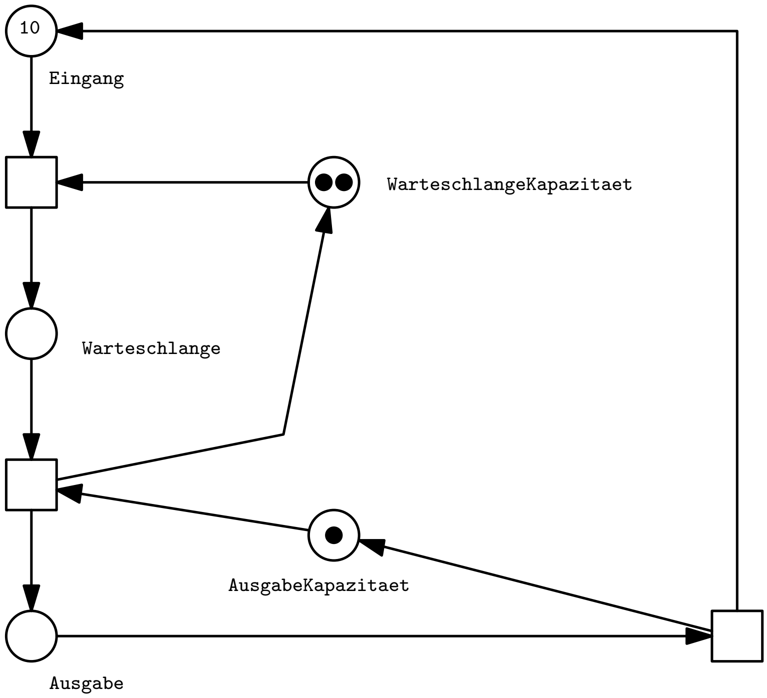
Praktikumsaufgabe 1: Modellierung der Petrinetz-Kantine

Lucas Jenss und Tommy Redel in Gruppe 1

1 Aufgabenteil 1

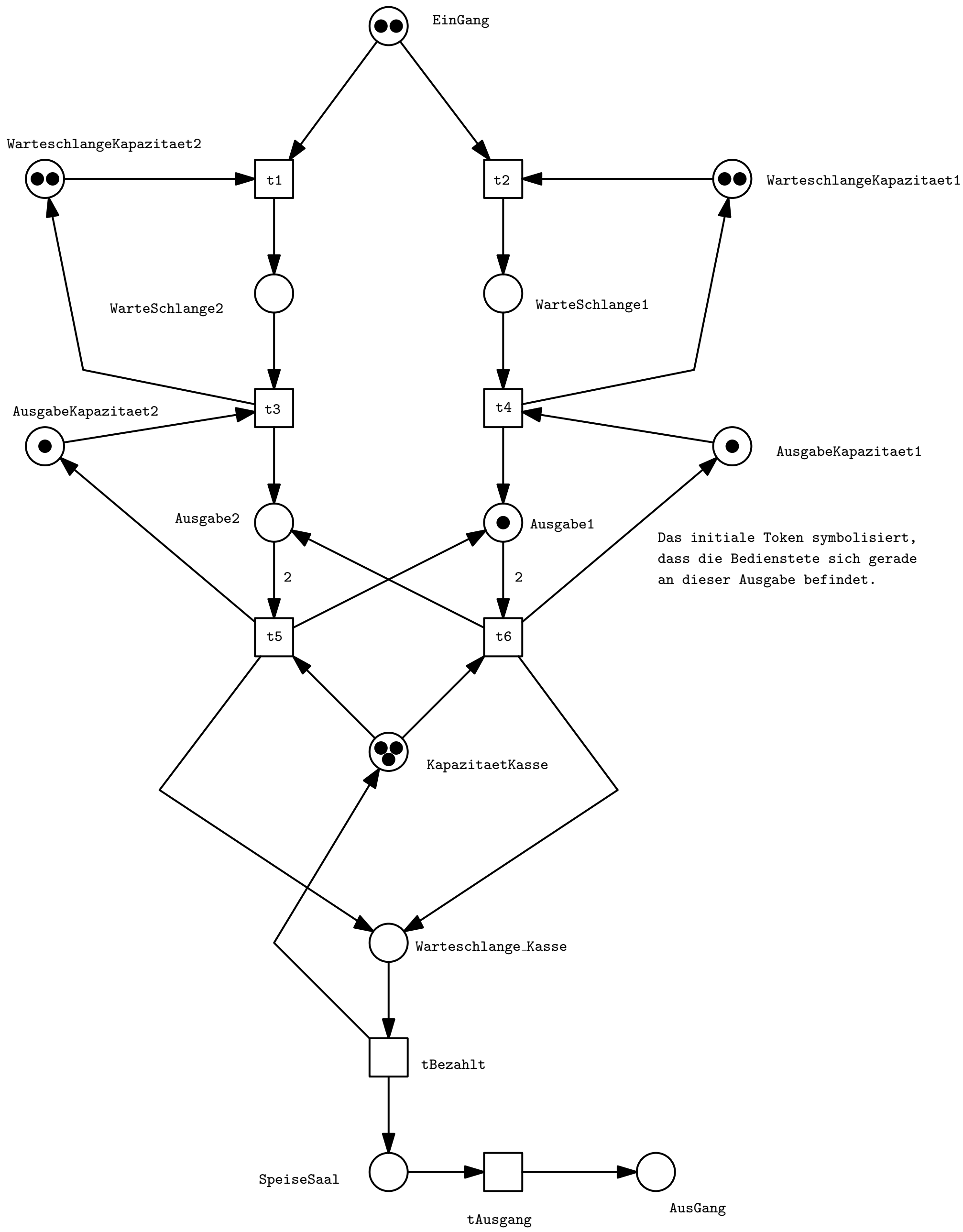
1.1 Erläuterung der Modellierung

Die erste zu treffende Entscheidung war die Art und Weise der Warteschlangen-Modellierung. Hierfür wurde folgendes System verwendet:



Um sicherzustellen, dass sich in der Stelle **Warteschlange** immer max. zwei Besucher befinden, wird eine zusätzliche Stelle **Warteschlange-Kapazitaet** hinzugefügt, deren Tokenanzahl der Kapazität der Warteschlange entspricht (im Beispiel zwei). Die Transition $\mathrm{Eingang} \rightarrow \mathrm{Warteschlange}$ wird nun mit einer zusätzlichen Kante $\mathrm{WarteschlangeKapazitaet} \rightarrow \mathrm{Warteschlange}$ versehen, sodass diese nur schalten kann, wenn mindestens ein Kapazitätstoken vorhanden ist. Die Transition, welche aus der Warteschlange hinausführt, wird analog mit einer Kante versehen, welche ein Token in die "WarteschlangeKapazitaet"-Stelle hinzufügt. Das selbe Verfahren wird eingesetzt, um sicherzugehen, dass sich maximal eine Person zur Zeit in der Ausgabe befindet.

In der Gesamtlösung verwenden wir diese Technik ebenfalls, um die Warteschlange an der Kasse auf drei Personen zu begrenzen. Die Bedienstete, welche für die Essensausgabe verantwortlich ist, wurde als zusätzliches Token modelliert, welches zwischen den beiden Ausgaben für die Gerichte alterniert. Eine erfolgreiche Ausgabe und anschließende Transition zur Kasse ist immer nur dann möglich, wenn sich zwei Token in der Ausgabe befinden, welche die Person sowie die Bedienstete repräsentieren:



1.2 Formale Angabe des Netzes

Ein S/T-Netz wird durch ein 4-Tupel $N = (P, T, W, M_0)$ definiert. Die einzelnen Bestandteile werden im Folgenden aufgeführt:

$$
\begin{aligned}
P = \{ & \\
& \mathrm{EG, WK2, WK1, WS2, WS1} \\
& \mathrm{AK2, AK1, A2, A1, KK} \\
& \mathrm{W\_K, SS, AG} \\
\} &
\end{aligned}
$$

$$
\begin{aligned}
T = \{ & \\
& \mathrm{t1, t2, t3, t4, t5, t6, tB, tA} \\
\} &
\end{aligned}
$$

$$
\begin{aligned}
W_1 = \{ & \\
& \mathrm{(EG, t2), (t2, WS1), (WS1, t4), (t4, A1),} \\
& \mathrm{(WK1, t2), (t4, WK1), (t1, WS2), (WS2, t3),} \\
& \mathrm{(EG, t1), (WK2, t1), (t3, WK2), (t3, A2),} \\
& \mathrm{(AK1, t4), (A1, t6), (t6, A2), (t6, AK1),} \\
& \mathrm{(A2, t5), (t5, A1), (AK2, t3), (t5, AK2),} \\
& \mathrm{(KK, t5), (KK, t6), (t5, W_K), (t6, W\_K),} \\
& \mathrm{(W\_K, tB), (tB, KK), (tB, SS), (SS, tA),} \\
& \mathrm{(tA, AG)} \\
\} &
\end{aligned}
$$

$$
W(x,y) = \begin{cases} 2 & ; falls\ (x,y) \in \{(Ausgabe2, t5), (Ausgabe, t6)\} \\ 1 & ; falls\ (x,y) \in W_1 \\ 0 & ; sonst \end{cases}
$$

$$
M_0(x) = \begin{cases} 3 & ; falls\ x = KK \\ 2 & ; falls\ x \in \{EG, WK2, WK1\} \\ 1 & ; falls\ x \in \{AK2, AK1, A1\} \\ 0 & ; sonst \end{cases}
$$

1.3 Formale Schaltfolge für zwei Personen mit unterschiedlichen Gerichten

Ausgehend vom im vorherigen Abschnitt beschriebenen Netz ist folgende Schaltfolge möglich, sodass zwei Personen unterschiedliche Gerichte einnehmen, beginnend mit einem M welches M_0 gleicht:

$$M(x) = \begin{cases} 3 & ; falls\ x = KK \\ 2 & ; falls\ x \in \{WK2, WK1, EG\} \\ 1 & ; falls\ x \in \{AK2, AK1, A1\} \\ 0 & ; sonst \end{cases}$$

Nun wird die Transitionfolge $t1 \rightarrow t3 \rightarrow t2 \rightarrow t4 \rightarrow t6 \rightarrow t5 \rightarrow tB \rightarrow tb \rightarrow tA \rightarrow tA$ geschaltet:

$$\bullet\, t1 = \{EG, WK2\}$$
$$M(WK) = 2 \geq 1 = W(WK, t1)$$
$$M(WK2) = 1 \geq 1 = W(WK2, t1)$$
$$M[t1\rangle M' \text{ und } M' = \begin{cases} 3 & ; falls\ x = KK \\ 2 & ; falls\ x \in \{WK1\} \\ 1 & ; falls\ x \in \{EG, WS2, WK2, AK2, AK1, A1\} \\ 0 & ; sonst \end{cases}$$

$$\bullet\, t3 = \{WS2, AK2\}$$
$$M'(WS2) = 1 \geq 1 = W(WS2, t3)$$
$$M'(AK2) = 1 \geq 1 = W(AK2, t3)$$

$$M'[t3\rangle M'' \text{ und } M'' = \begin{cases} 3 & ; falls\ x = KK \\ 2 & ; falls\ x \in \{WK1, WK2\} \\ 1 & ; falls\ x \in \{EG, A2, A1\} \\ 0 & ; sonst \end{cases}$$

$$\bullet\, t2 = \{EG, WK1\}$$
$$M''(EG) = 1 \geq 1 = W(EG, t2)$$
$$M''(WK1) = 2 \geq 1 = W(WK1, t2)$$

$$M''[t2\rangle M''' \text{ und } M''' = \begin{cases} 3 & ; falls\ x = KK \\ 2 & ; falls\ x \in \{WK2\} \\ 1 & ; falls\ x \in \{WK1, WS1, A2, A1\} \\ 0 & ; sonst \end{cases}$$

$$\bullet\, t4 = \{WS1, AK1\}$$
$$M'''(WS1) = 1 \geq 1 = W(WS1, t4)$$
$$M'''(AK1) = 1 \geq 1 = W(AK1, t4)$$

$$M'''[t4\rangle M'''' \text{ und } M'''' = \begin{cases} 3 & ; falls\ x = KK \\ 2 & ; falls\ x \in \{WK1, WK2, A1\} \\ 1 & ; falls\ x \in \{A2\} \\ 0 & ; sonst \end{cases}$$

$$\bullet\, t6 = \{A1, KK\}$$
$$M''''(A1) = 2 \geq 2 = W(A1, t6)$$
$$M''''(KK) = 3 \geq 1 = W(KK, t6)$$

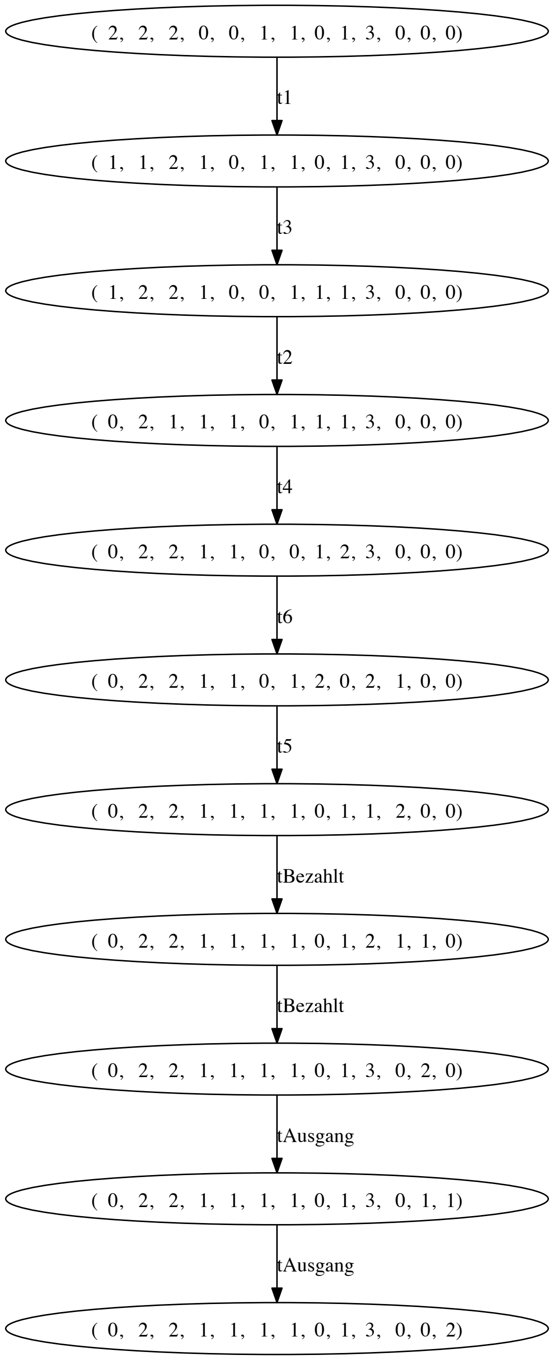
$$M''''[t6\rangle M''''' \text{ und } M''''' = \begin{cases} 2 & ; falls\ x \in \{KK, WK1, WK2, A2\} \\ 1 & ; falls\ x \in \{AK1, W\_K\} \\ 0 & ; sonst \end{cases}$$

usw.

1.4 Erreichbarkeitsgraph der Schaltfolge

Die Reihenfolge der Stellen lautet wie folgt:

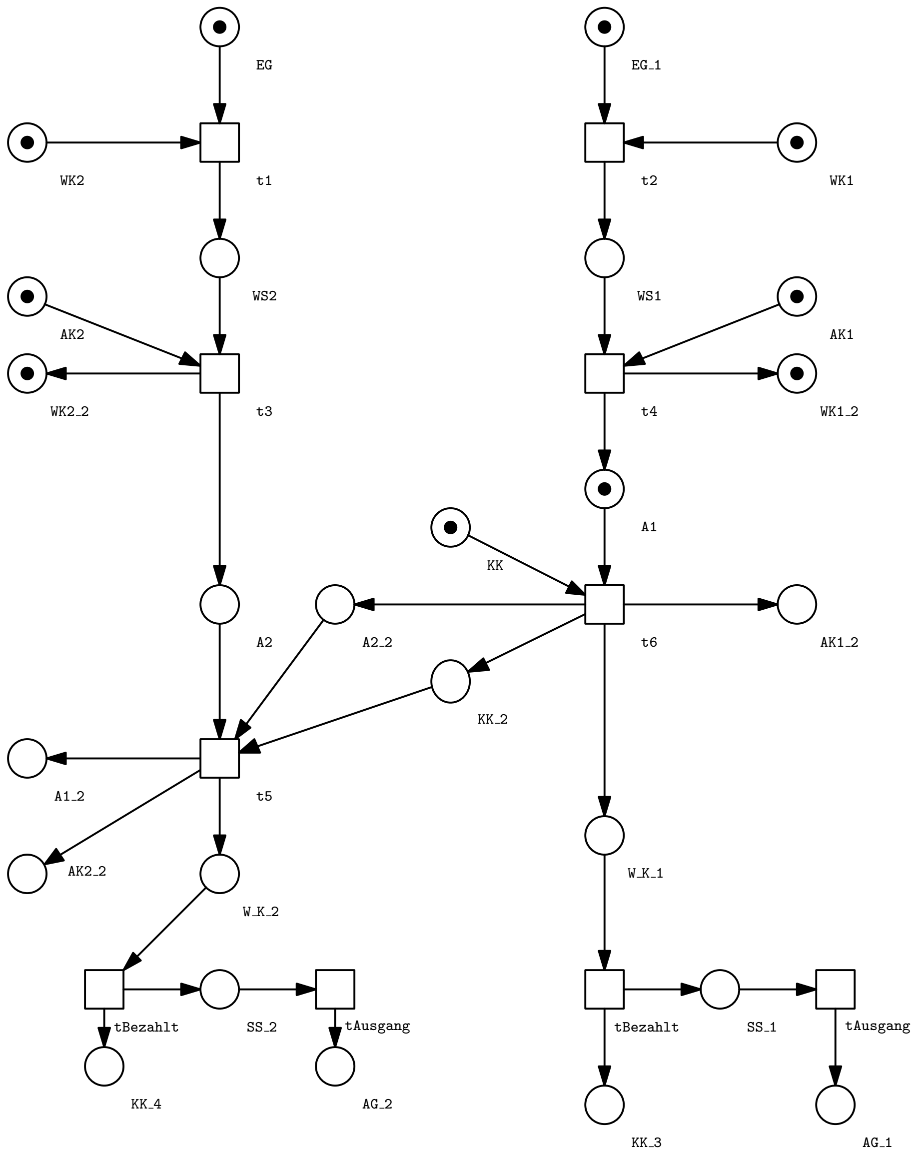
`EG, WK2, WK1, WS2, WS1, AK2, AK1, A2, A1, KK, W_K, SS, AG`


(2, 2, 2, 0, 0, 1, 1, 0, 1, 3, 0, 0, 0)
t1
(1, 1, 2, 1, 0, 1, 1, 0, 1, 3, 0, 0, 0)
t3
(1, 2, 2, 1, 0, 0, 1, 1, 1, 3, 0, 0, 0)
t2
(0, 2, 1, 1, 1, 0, 1, 1, 1, 3, 0, 0, 0)
t4
(0, 2, 2, 1, 1, 0, 0, 1, 2, 3, 0, 0, 0)
t6
(0, 2, 2, 1, 1, 0, 1, 2, 0, 2, 1, 0, 0)
t5
(0, 2, 2, 1, 1, 1, 1, 0, 1, 1, 2, 0, 0)
tBezahlt
(0, 2, 2, 1, 1, 1, 1, 0, 1, 2, 1, 1, 0)
tBezahlt
(0, 2, 2, 1, 1, 1, 1, 0, 1, 3, 0, 2, 0)
tAusgang
(0, 2, 2, 1, 1, 1, 1, 0, 1, 3, 0, 1, 1)
tAusgang
(0, 2, 2, 1, 1, 1, 1, 0, 1, 3, 0, 0, 2)

1.5 Netzkomplement

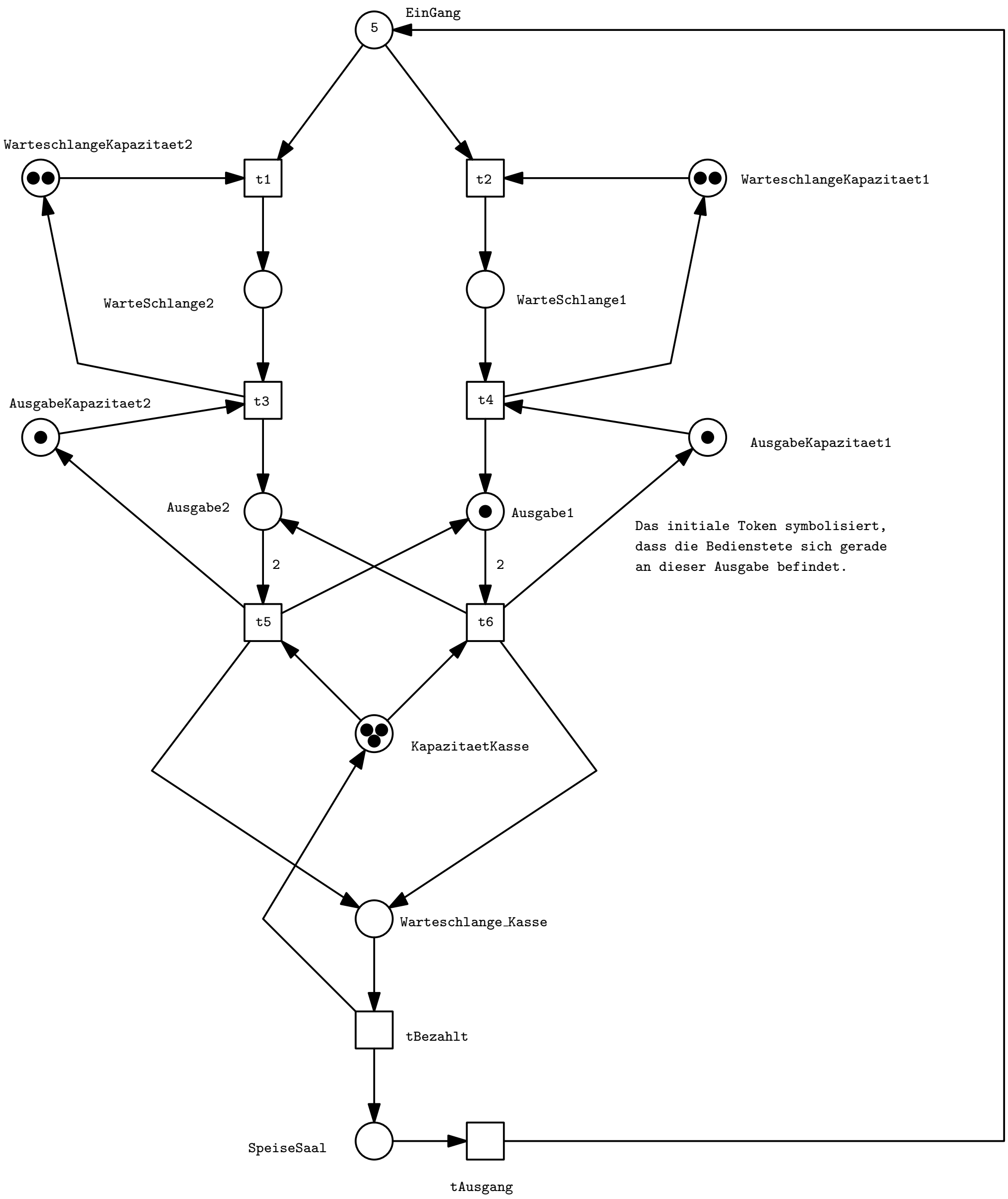
Da Snoopy keine Kapazitäten unterstützt, wurde das Netz von Anfang an ohne diese modelliert. Das Netzkomplement ist somit nicht zu berechnen, bzw. es ist identisch mit der bereits gezeigten Lösung.

1.6 Prozessnetz

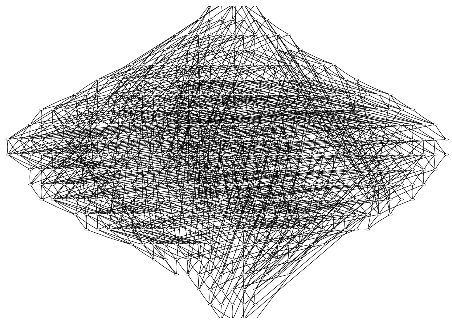


1.7 Geschlossenes System mit fünf Kunden

Das geschlossene Netz für fünf Kunden wurde wie folgt modelliert:



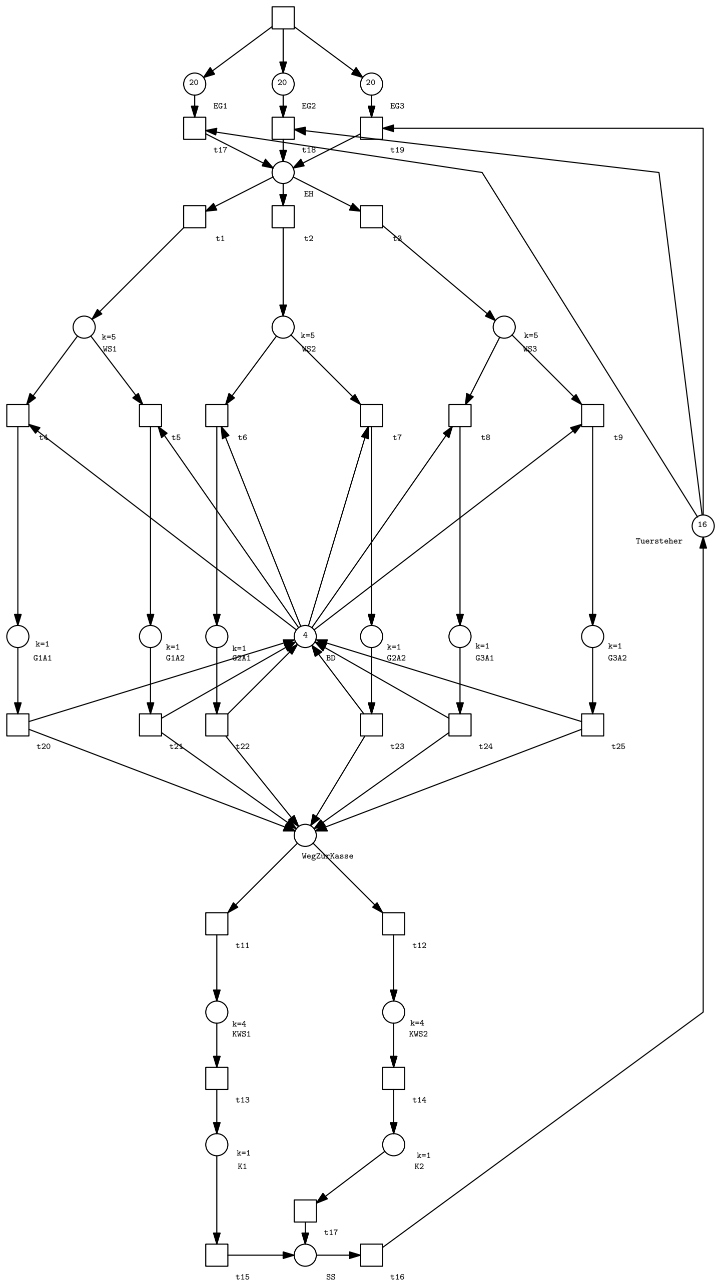
Der daraus resultierende Erreichbarkeitsgraph hat 484 Knoten und 1518 Kanten. Die kombinatorische Explosion (nur fünf Personen, 1500 Kanten) ist das Resultat der Tatsache, dass, ausgehend von jedem Knoten, fast alle Transitionen schalten können, da das Netz viele Transitionen, aber nur wenige Limitierungen für das Schaltverhalten dieser hat. Wenn man den Graph von Charlie mit dem “TreeLayout” rendern lässt, sieht man, dass er bei dem Knoten “Alle Personen befinden sich am Eingang” anfängt, und dann immer breiter wird. Die Breite lässt sich dadurch erklären, dass die Anzahl der Möglichkeiten, welche Transition als nächstes schaltet, wächst, je mehr Personen/Token sich innerhalb des Netztes befinden. Nachdem der Graph seine maximale Breite erreicht hat, fängt er wieder an “zusammenzuschrumpfen”, bis er im Knoten “Alle Personen befinden sich im Speisesaal” wieder nur eine Breite von einem Knoten hat. Wenn sich also nun alle Personen im Speisesaal versammelt haben, haben diese nur noch die Möglichkeit wieder zum Eingang zurückzukehren und wieder ein Gericht zu wählen. Da an dieser Stelle aber bereits alle möglichen Durchläufe der fünf Personen durch das Netz im Graph vorhanden sind, werden keine neuen Knoten erzeugt, sondern lediglich Kanten zu den bereits bestehenden. Hier zur Referenz noch einmal die Form des Graphen (bei dem leider die oberen und unteren Endknoten abgeschnitten sind):



2 Aufgabenteil 2

2.1 Modellierung

Das Netz der umgebauten Kantine wurde wie folgt modelliert:



2.2 Konflikte und Nebenläufigkeit

Nach der in der Vorlesung behandelten Definition treten Konflikte zwischen zwei Transitionen t_1 und t_2 immer dann auf, wenn $t1$ und $t2$ M-aktiviert sind und sich im Vorbereich von $t1$ und $t2$ eine **gemeinsame** Stelle p befindet, dessen Markierung die Summe der Kantengewichte $W(p, t_1)$ und $W(p, t_2)$ unterschreitet.

Eine Nebenläufigkeit zwischen zwei Transitionen t_1 und t_2 besteht, wenn $t1$ und $t2$ M-aktiviert sind und sich im Vorbereich von $t1$ und $t2$ eine **gemeinsame** Stelle p befindet, dessen Markierung gleich der Summe der Kantengewichte $W(p, t_1)$ und $W(p, t_2)$ ist oder diese sogar überschreitet.

2.2.1 Teil (a)

Zu Beginn der Simulation existieren noch keine Konflikte, da lediglich die Transitionen $WS1$, $WS1$ und $WS1$ schalten können und auf den Stellen in den Vorbereichen genügend Token liegen.

Ein erster Konflikt ergibt sich im Laufe der Simulation jeweils an den Warteschlangen $WS1$, $WS2$ und $WS3$. Der Konflikt soll im Folgenden am Beispiel von $WS1$ erläutert werden. Die Markierung $M(WS1)$ soll 1 betragen.

$$\bullet t4 \cap \bullet t5 = \{WS1\}$$
$$M(WS1) = 1 < W(WS1, t4) + W(WS1, t5) = 2$$
$$1 < 2$$

$t4$ und $t5$ stehen also in Konflikt miteinander, da 2 Token benötigt werden, um beide Transitionen zu schalten, jedoch nur 1 Token vorhanden ist. Dieser Beweis lässt sich analog auf die Transitionspaare $(t6, t7)$ und $(t8, t9)$ sowie dessen Warteschlangen $WS2$ und $WS3$ übertragen.

Ein weiterer Konflikt befindet sich im Bereich der Ausgaben, da es insgesamt 6 Essensausgaben und nur 4 Bedienstete gibt.

$$\bullet t4 \cap \bullet t5 \cap \bullet t6 \cap \bullet t7 \cap \bullet t8 \cap \bullet t9 = \{BD\}$$
$$M(BD) = 4$$
$$W(BD, t4) + W(BD, t5) + W(BD, t6) + W(BD, t7) +$$
$$W(BD, t8) + W(BD, t9) = 6$$
$$4 < 6$$

Demnach stehen alle sechs Transitionen jeweils miteinander in Konflikt bezüglich der Stelle BD.

Im Bereich der Eingangshalle EH sowie beim $Tuersteher$ kann es ebenfalls zu Konflikten kommen, sofern auf den Stellen weniger als drei Token liegen.

2.2.2 Teil (b)

Unter der Annahme, dass sich genügend Kunden in den jeweiligen Warteschlangen befinden, ist die gleichzeitige Ausgabe von Essen abhängig von der Anzahl der Bediensteten. Da die Stelle BD hier vier Markierungen aufweist, können also maximal vier Kunden gleichzeitig ihre Mahlzeit bekommen. Als Beweis für diese Behauptung kann wieder der obige Konflikt zwischen $t4$, $t5$, $t6$, $t7$, $t8$ und $t9$ herangezogen werden. In diesem Konflikt war $M(BD)$ kleiner als die Summe der Kantengewichte der genannten Transitionen. Demnach ist die maximale gleichzeitige Bearbeitung von Kunden abhängig von der Anzahl der Bediensteten. Daraus resultiert, dass maximal vier der oben genannten Transitionen gleichzeitig schalten können.

3 Definition: Netzmorphismus

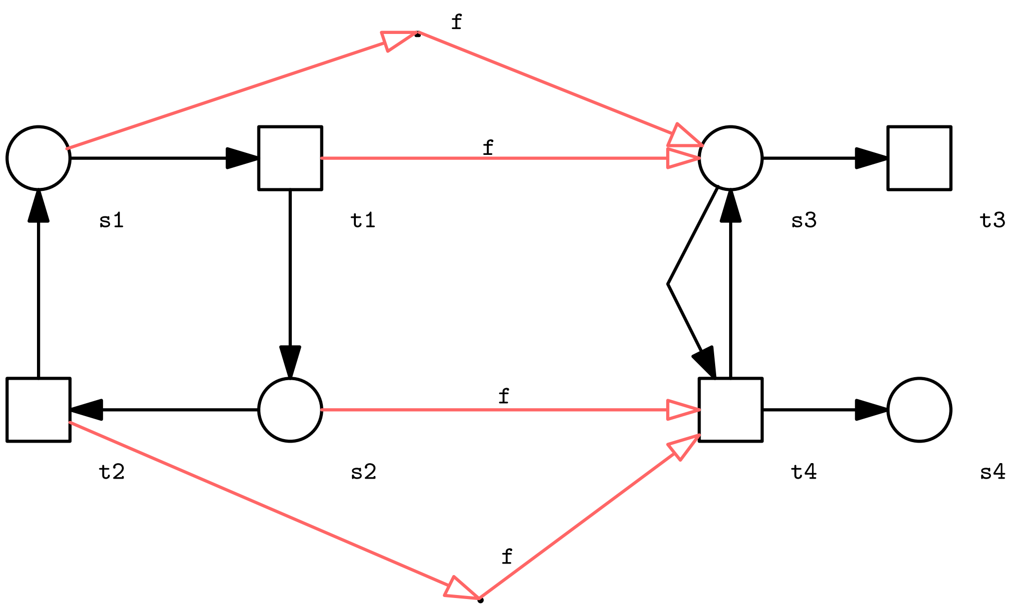
In der Vorlesung haben wir eine Definition von Netzmorphismus kennengelernt, welche ein Tupel $f = (f_P, t_T)$ beschreibt, wobei f_P die Stellenabbildungsfunktion und f_T die Transitionsabbildungsfunktion ist. Zusätzlich wird verlangt, dass die Vorbereiche und Nachbereiche einer Transition gewahrt werden müssen, oder mit anderen Worten, dass die Summe der eingehenden und ausgehenden Kantengewichte im neuen Netz erhalten bleiben muss. “Alles andere wäre doof!” (Padberg, 2013).

3.1 Alternative Definition

Eine Netzabbildung f von N_1 in N_2 bildet $S_1 \cup T_1$ in $S_2 \cup T_2$ ab. Wir nennen sie Netzmorphismus, wenn sie keine Kanten zerreißt, d.h. wenn:

$$(x, y) \in F_1 \Rightarrow f(x) = f(y) \vee (f(x), f(y)) \in F_2$$

Quelle: Petri-Netze - Grundlagen und Anwendung (Bernd Baumgarten), BI-Wiss.-Verl., 1990, ISBN 3–411–14291-X



Das gezeigte Beispiel ist laut der alternativen Definition ein Netzmorphismus, mit der Funktion

$$f(x) = \begin{cases} (s3, s3) & ; falls\ x = (s1, t1) \\ (s3, t4) & ; falls\ x = (t1, s2) \\ (t4, t4) & ; falls\ x = (s2, t2) \\ (t4, s3) & ; falls\ x = (t2, s1) \end{cases}$$

Diese deckt sich nicht mit der Definition aus der Vorlesung, da sie

1. die Abbildung von Stellen auf Transitionen und umgekehrt erlaubt
2. die Kantengewichte in Vor- und Nachbereichen nicht wahrt. Im Beispiel haben t_2 und s_2 jeweils eine eingehende Kante, deren Abbildung t_4 hingegen nur eine.