Musterlösungen zu den Hausaufgaben auf

Blatt 4

der Übungen zur Vorlesung

“Grundlagen Betriebssysteme und Systemsoftware

G.Groh, 11.11.2008

Aufgabe 1 Lösungsvorschlag

Gegeben sei das (vorerst einmal boolsche) Petri-Netz in Abbildung 1.

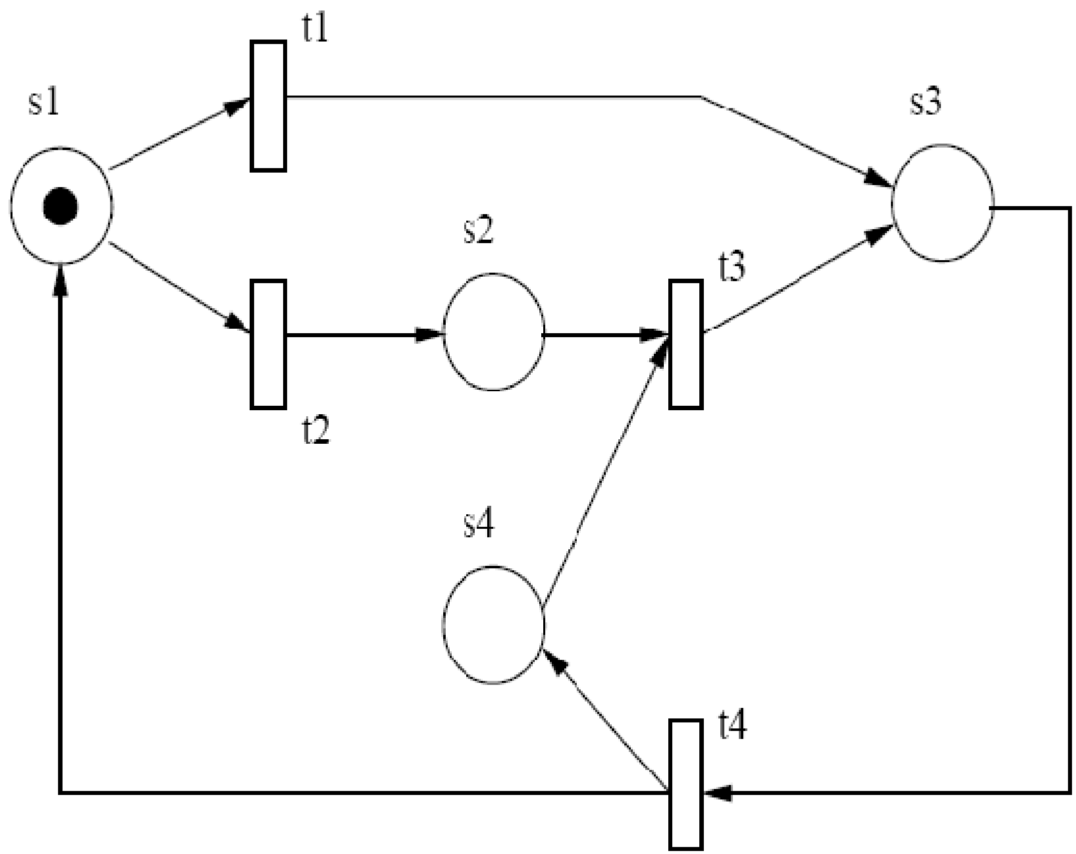
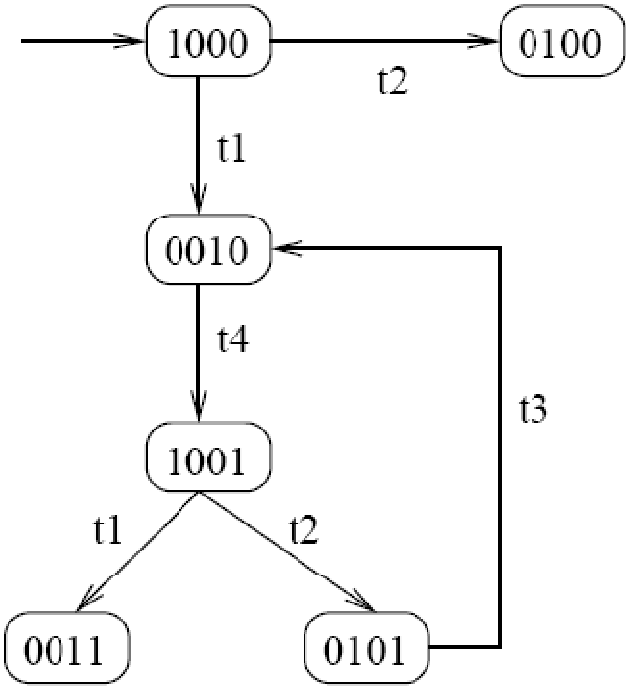


Figure 1: Petri-Netz

Aufgabe 1.1 Lösungsvorschlag

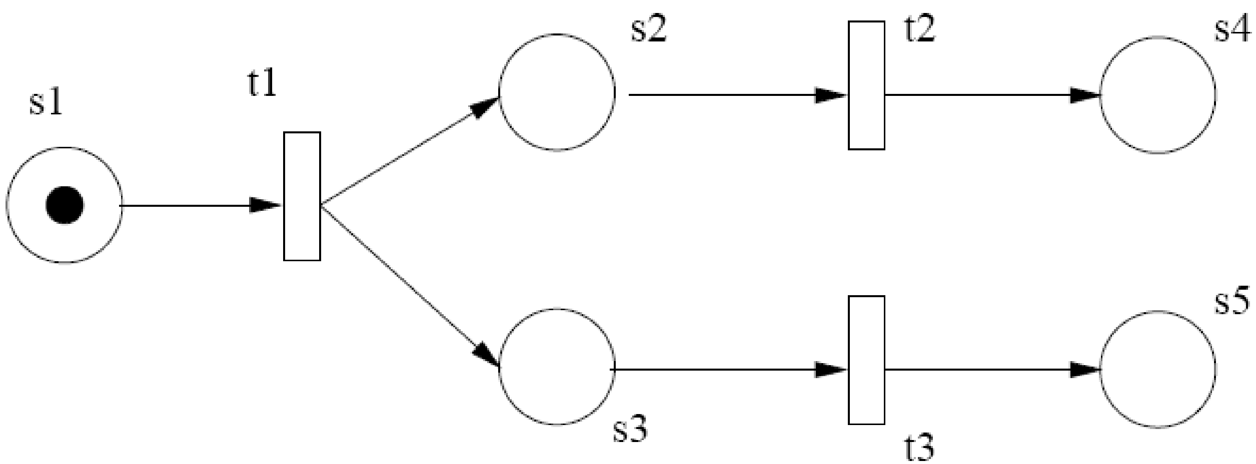
Um alle erreichbaren Belegungen zu ermitteln, sind alle zulässigen Schaltvorgänge von der Startbelegung aus zu untersuchen und diese Betrachtung dann für alle Folgebelegungen fortzusetzen. Diese Untersuchung ist gleichwertig mit dem Aufstellen eines Erreichbarkeitsgraphen (oder Zustands-Übergangs-Automaten) für das Petri-Netz. Wir benutzen dafür die Bezeichnung der Stellen des Petri-Netzes. Boolsche Belegungen können nun durch ein 4-Tupel $b \in bool$ repräsentiert werden, wobei ein 4-Tupel $b = (b_1, b_2, b_3, b_4)$ bedeutet: Die Stelle si ist genau dann belegt, wenn $b_i = 1$ (`TRUE`). Das ergibt folgenden Erreichbarkeitsgraphen:



Aufgabe 1.2 Lösungsvorschlag

Ein nicht-sequentieller Ablauf existiert genau dann, wenn zwei Transitionen gleichzeitig schalten können. In dem in dieser Aufgabe betrachteten Petri-Netz ist dies nicht der Fall, es existiert also kein nebenläufiger Ablauf.

Ein Beispiel, bei dem zwei Transitionen t2 und t3 gleichzeitig schalten können, ist folgendes Petri-Netz:

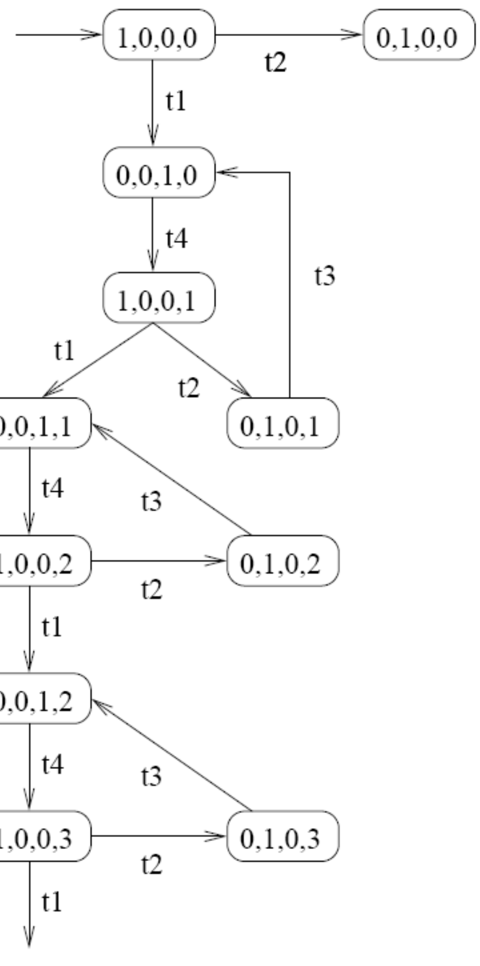


Aufgabe 1.3 Lösungsvorschlag

Ja. Es sind zwei Belegungen erreichbar (`0100`, `0011`), in denen keine Transition mehr schalten kann. Dies erkennt man, indem man aus dem Erreichbarkeitsgraphen aus Teilaufgabe 3.3 Knoten sucht, aus denen keine Kanten mehr herausführen.

Aufgabe 1.4 Lösungsvorschlag

Ja. Wenn man natürlichzahlige Belegungen betrachtet, kann bei der Belegung `0011` die Transition t4 schalten (was vorher durch den belegten Ausgangsplatz verhindert wurde). Dies ergibt die Zustandsübergänge in folgendem Erreichbarkeitsgraphen:



Aufgabe 1.5 Lösungsvorschlag

Die Transitionen t2 und t3 können ausgehungert werden, da (unendliche) Abläufe möglich sind, in denen diese Transitionen nicht mehr enthalten sind:

$$t1 \longrightarrow t4 \longrightarrow t1 \longrightarrow t4 \longrightarrow t1 \longrightarrow t4 \longrightarrow \ldots$$

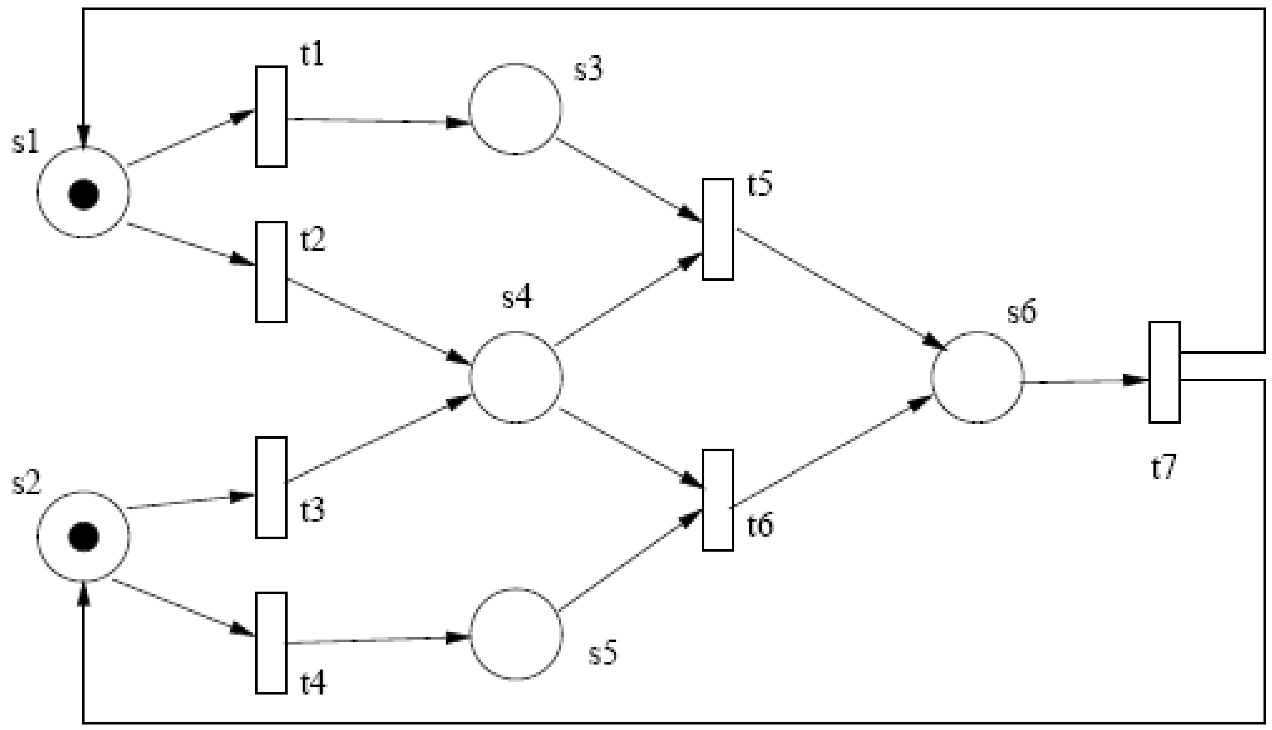
Dies gilt allerdings nur, wenn wir die Kapazität von s4 auf ∞ setzen, was laut Def. möglich ist. . Da ∞ aber keine natürliche Zahl ist (wie in der Aufgabe gefordert), ist das Aushungern von t2 und t3 also eigentlich nicht möglich.

Man kann zwar t1 "pseudo-aushungern" mit der Kette t1→t4→t2→t3→t4→t2→t3→... aber laut Def im Skript ist dies kein echtes Aushungern, da t1 einmal vorkommt.

Die Antwort lautet insgesamt also nein.

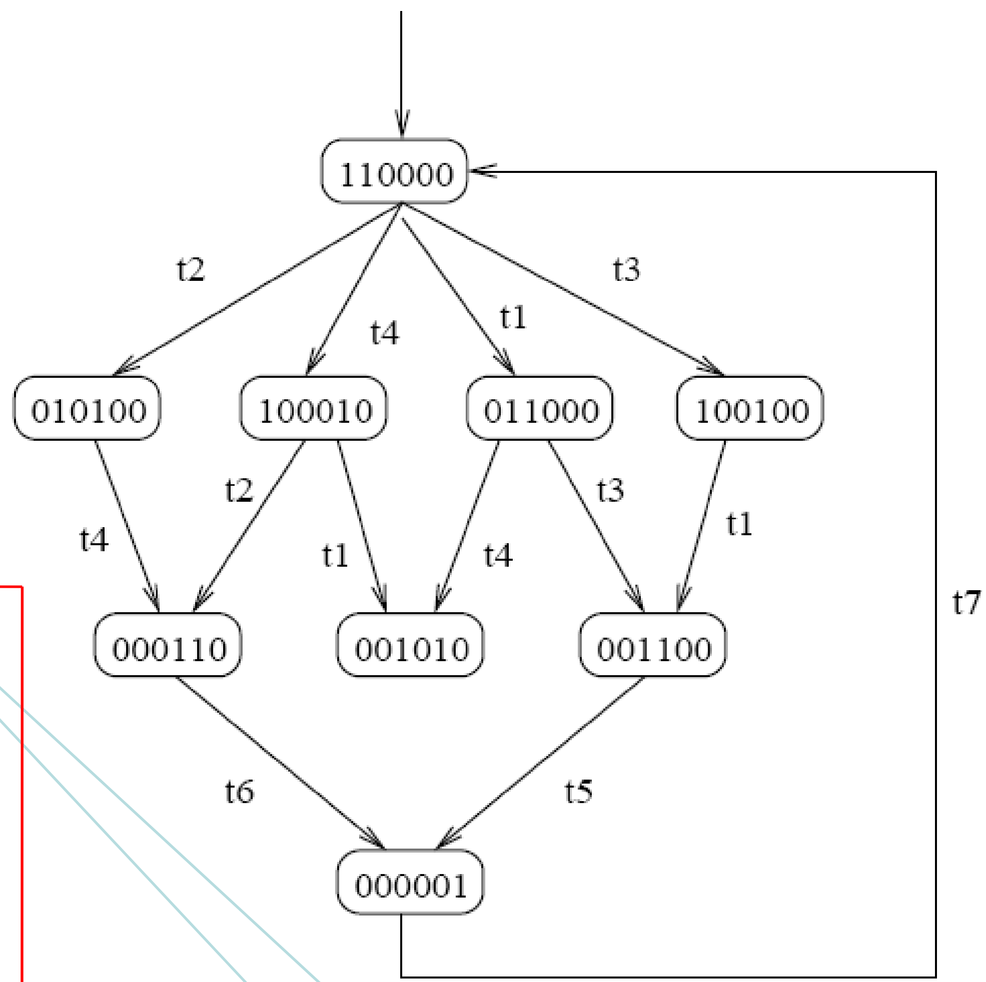
Aufgabe 2 Lösungsvorschlag

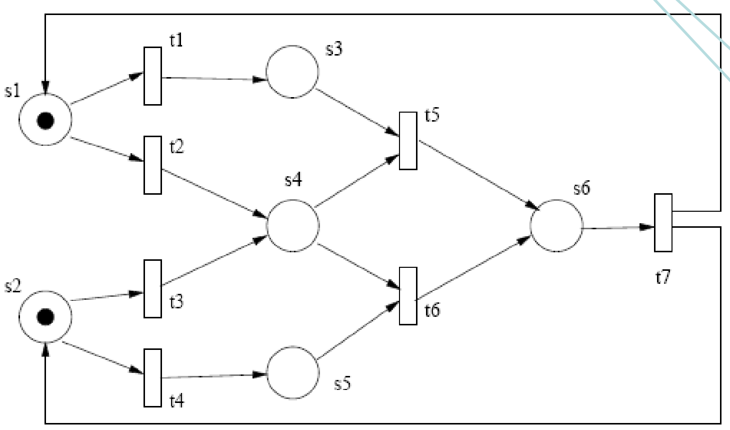
Gegeben sei das boolsche Petri-Netz



Aufgabe 2.1 Lösungsvorschlag

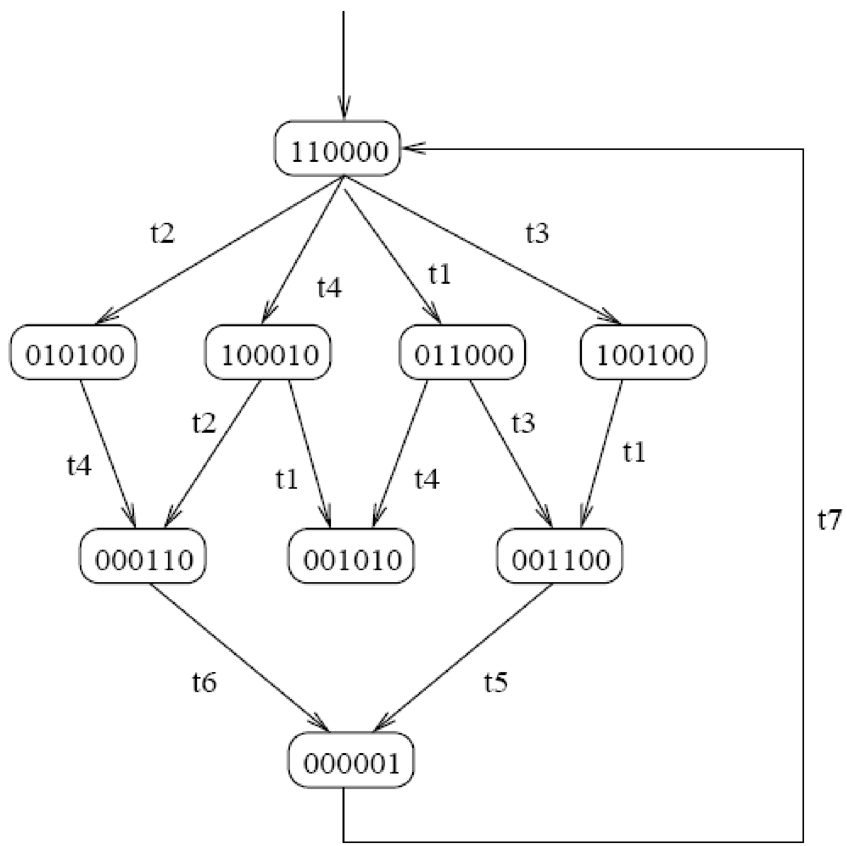
Erreichbarkeitsgraph:

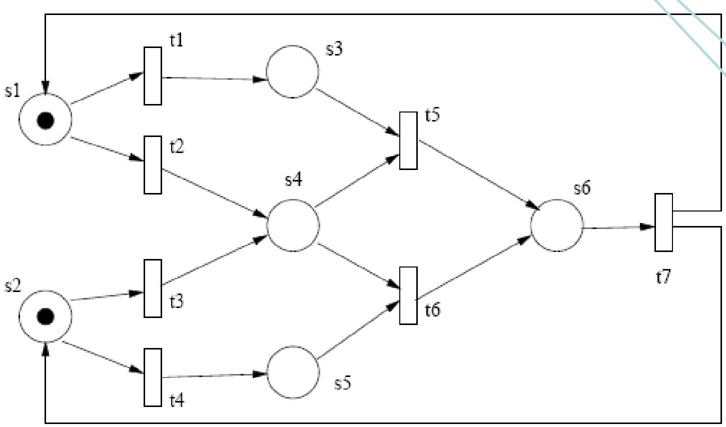




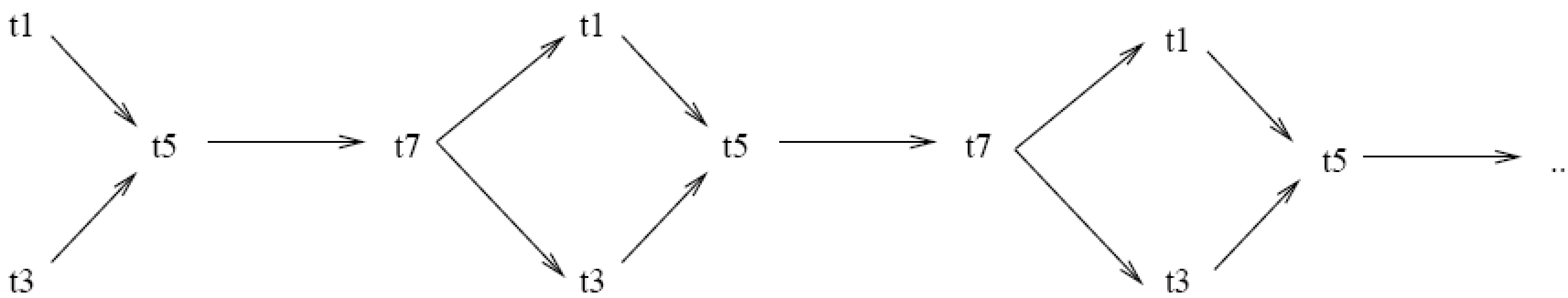
Aufgabe 2.2 Lösungsvorschlag

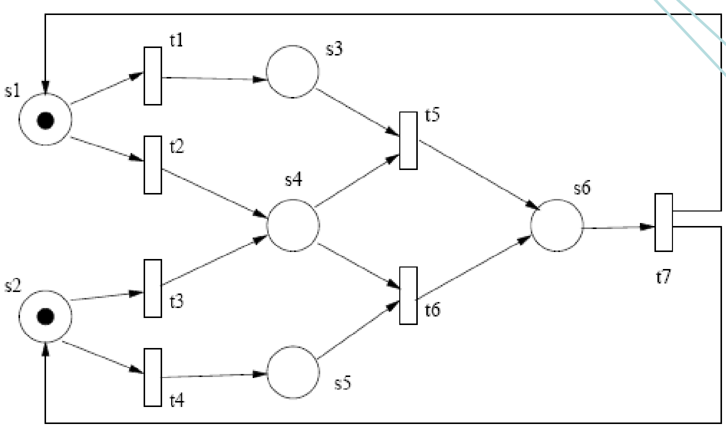
Ja. Die Belegung 001010 entspricht einer Verklemmung, da keine Transition mehr schalten kann.

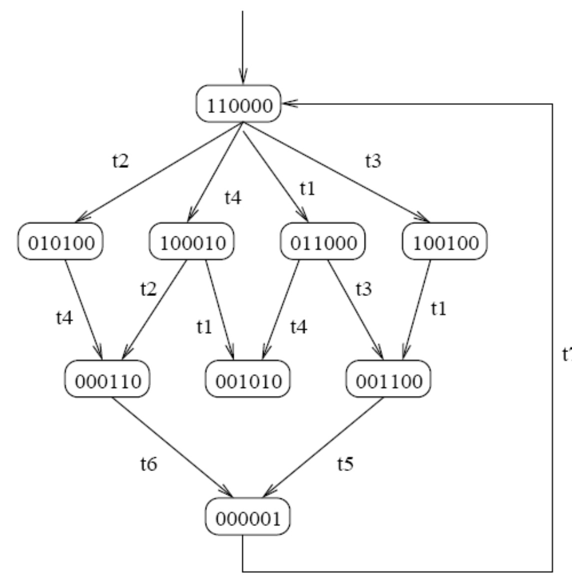




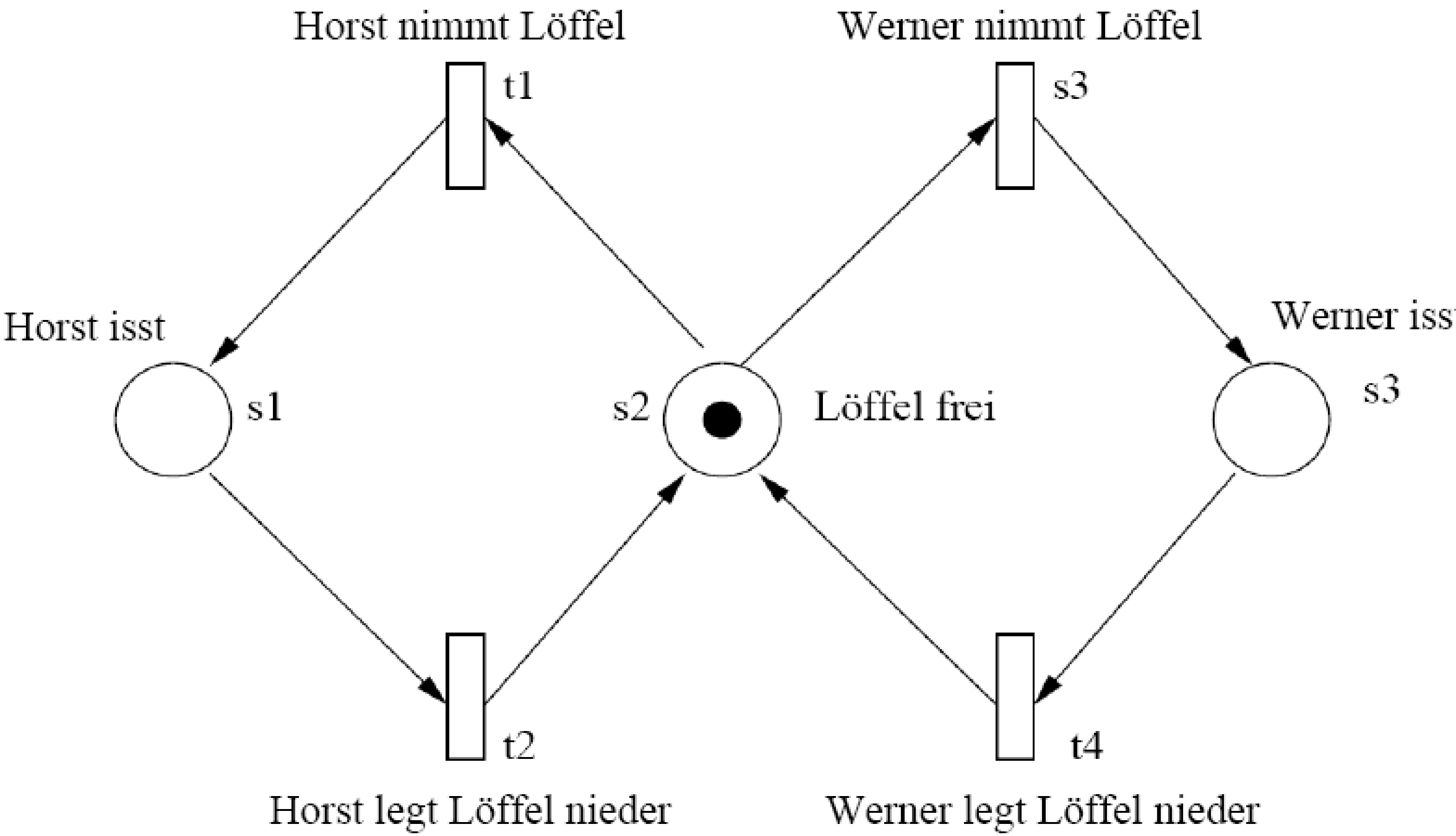
Folgender unendlicher Ablauf existiert:



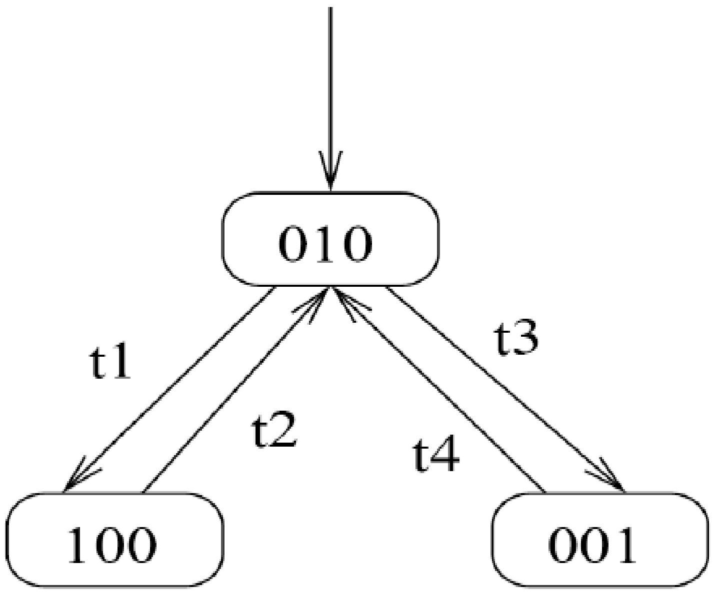




Aufgabe 3.1 Lösungsvorschlag



Aufgabe 3.2 Lösungsvorschlag



In diesem Erreichbarkeitsgraphen gibt es keinen Zustand in dem sowohl Horst als auch Werner essen.