




Übung zur Vorlesung Angewandte Informatik I
WS 2004/2005

Übung 10: Petrinetze
- Lösungsvorschläge -

Übung 9 Wiederholung

Petrinetze

- **Stellen:** beschreiben passive (statische) Systemkomponenten:
Bedingungen, Medien, Materialbehälter, etc.
Symbol: 
- **Transitionen:** beschreiben aktive (dynamische) Systemkomponenten:
Aktionen, Handlungen, Transporte, Transformationen, etc.
Symbol: 
- **Kanten:** beschreiben (je nach Transition):
Vor-/Nachbedingungen von Aktivitäten, Start und Ziel von
Transporten, Eingabe und Ausgabe von Programmen, etc.
Symbol: 
- Keine Aussage über Schaltreihenfolge

Übung 9 Wiederholung

Definition: Struktur von Petrinetzen

Ein Petrinetz ist ein Tripel $N = (S, T, F)$ mit

- S, T sind endliche Mengen
- $S \cap T = \emptyset$
- $S \cup T \neq \emptyset$
- $F \subseteq (S \times T) \cup (T \times S)$ ist eine binäre Relation über $S \cup T$

→ Ein Petrinetz ist ein bipartiter Graph

Ein Graph heißt *bipartit*, wenn sich seine Knotenmenge in zwei Klassen zerlegen läßt, so dass jeder Knoten der einen Klasse nur mit Knoten der anderen Klasse durch eine Kante verbunden ist (und umgekehrt)

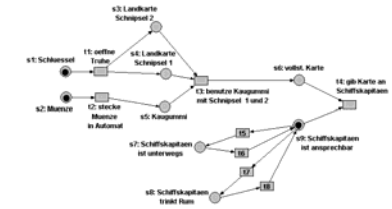
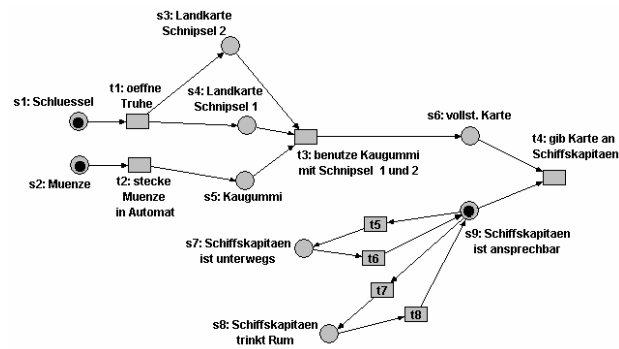
Übung 9 Aufgabe 1

Nach Ihrem Studium in Karlsruhe beschließen Sie bei einer Firma anzufangen, die Computerspiele herstellt. In einem Projektmeeting zum Adventure-Spiel „Donkey Island“ sind viele der „Kreativen“ mit einer Modellierung des Spielverlaufs überfordert:

„Der Benutzer soll eine Spielfigur steuern können, die sich am Anfang auf einer Pirateninsel befindet. Dabei sollen nur bestimmte Aktionen ein Fortkommen ermöglichen, während bei allen anderen ein Text angezeigt wird. Die Spielfigur besitzt am Anfang einen Schlüssel und eine Münze. Mit Hilfe des Schlüssels ist die Spielfigur in der Lage, eine Truhe zu öffnen, die erst bei näherem Hinschauen auf dem Bildschirm erkennbar ist. Nach dem Öffnen der Truhe mit dem Schlüssel findet die Spielfigur zwei Papierschnipsel, die zur selben Landkarte gehören. Die Spielfigur kann auch die Münze in einen Kaugummiautomaten stecken, der auch auf dem Bildschirm sichtbar ist. Mit Hilfe des Kaugummis kann die Spielfigur die zwei Papierschnipsel zu einer vollständigen Landkarte zusammenfügen. Ab und zu soll ein Schiffskapitän auf dem Bildschirm erscheinen. Gelegentlich trinkt er Rum und ist nicht ansprechbar. Erst wenn der Schiffskapitän ansprechbar ist (d.h. wenn er nicht unterwegs ist und nicht Rum trinkt), kann ihm die Spielfigur die vollständige Landkarte überreichen. In diesem Moment kommt dann das nächste Level“.

- Modellieren Sie den logischen Spielverlauf mit Hilfe eines Petrinetzes!
- Welche Eigenschaften hat das Netz in a) und was bedeutet das für den Spielverlauf?

- Petrinetz zum Spielverlauf



Ein S/T-Netz (N, m_0)

✖ **terminiert**,

1) wenn die Menge der möglichen Schaltfolgen endlich ist.

Anders formuliert:

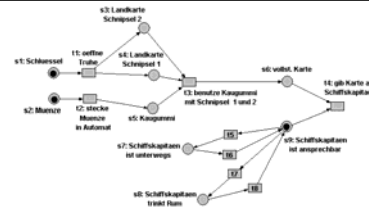
2) der zugehörige Markierungsgraph ist endlich und hat keine Zyklen.

Hier: Markierungsgraph hat Zyklen, z.B. kommt man von $(0,0,0,0,0,1,1,0,0,1)$ nach $(0,0,0,0,0,1,1,0,0)$ und umgekehrt. D.h. vollst. Karte kann evtl. nicht an Kapitän übergeben werden, so dass evtl. kein Übergang zum nächsten Level stattfinden kann!

✓ ist **verklemmungsfrei**,

wenn jede erreichbare Markierung eine Transition aktiviert.

Sonst würde im Spiel nichts mehr passieren.



Ein S/T-Netz (N, m_0)

✖ ist **lebendig**,

1) wenn von jeder erreichbaren Markierung ausgehend eine Schaltfolge existiert, die alle Transitionen umfaßt.

Anders formuliert

2) Jede Transition t kann – von jedem $m \in [m_0]$ ausgehend – immer wieder aktiviert werden.

Wenn der Spieler erst einmal die vollständige Karte hat, kann er z.B. nicht mehr die Truhe öffnen (d.h. t_1 aktivieren)

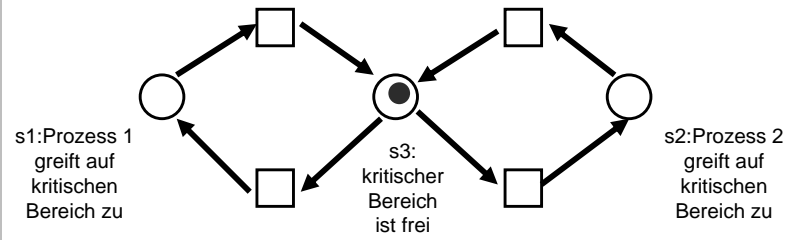
✖ ist **rücksetzbar**,

wenn von jeder erreichbaren Markierung wieder die Anfangsmarkierung m_0 erreichbar ist.

Während des Spiels kann der Spieler nicht mehr zur Anfangssituation zurückkehren.

In einem Betriebssystem sollen zwei Prozesse gleichzeitig ausführbar sein. Sie tauschen Daten untereinander aus, indem sie im selben Speicherbereich lesend und schreibend zugreifen. Dabei soll verhindert werden, dass beide Prozesse gleichzeitig auf diesen (kritischen) Speicherbereich zugreifen, da dies zu inkonsistenten Zuständen führen könnte (z.B. ein Prozess liest schon, während der andere noch schreibt).

Modellieren Sie mit einem Stellen/Transitions-Netz diesen gegenseitigen Ausschluss. Das Netz soll zwei Stellen besitzen, die jeweils den Zustand „Prozess i greift auf kritischen Bereich zu“ darstellen. Zusätzlich darf nur eine weitere (dritte) Stelle verwendet werden.



Dieses Konzept des gegenseitigen Ausschlusses wird in vielen Systemen angewendet. Es kann als ein Entwurfsmuster (Design Pattern) angesehen werden.