Teil 2 der Grundvorlesung

2. Interaktionen

~~2.1 Objektorientierung~~

2.2 Prozesse

2.3 Internet, Vernetzung

2.4 Modellierung

Gliederung des Kapitels

2.2 Prozesse

2.2.1 Stellen-Transitions-Netze

2.2.2 Nachrichtenaustausch

2.2.3 Synchronisation

2.2.4 Parallelität

2.2.5 Nebenläufigkeit in Ada95

2.2.1 Stellen-Transitions-Netze

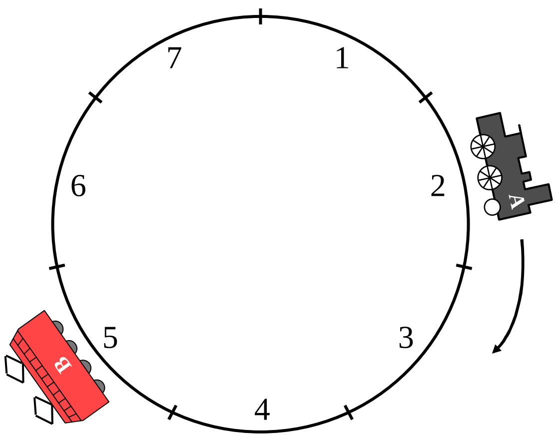
Bisher: Sequentielle Programmierung, d.h., höchstens eine Stelle im Programm wird in jedem Augenblick bearbeitet. Jeder Ablauf wird hierbei in eine Folge nacheinander auszuführender Aktivitäten zerlegt.

Im Folgenden wollen wir unabhängig voneinander ablaufende Programme (Prozesse, Objekte) und deren Kommunikation beschreiben. Man spricht von Nebenläufigkeit (engl.: concurrency) und von nebenläufigen, von verteilten und von parallelen Systemen.

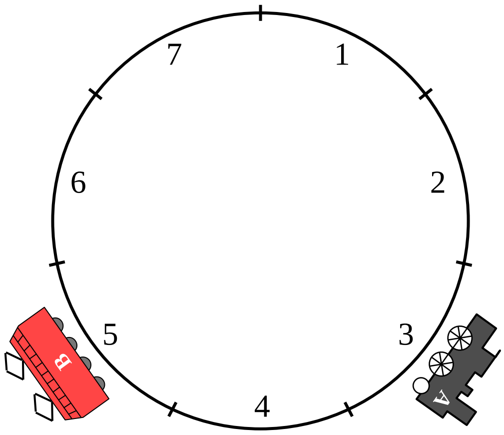
Ansatz: Verallgemeinere endliche Automaten, indem mehrere Zustände gleichzeitig oder nacheinander aktiv sind. Als Beispiel wählen wir Züge auf einer Kreisstrecke.

2.2.1.1 Beispiel: Züge auf einer kreisförmigen Strecke.

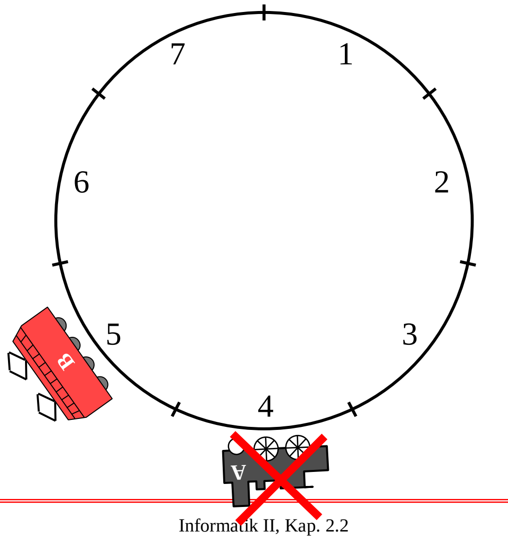
Damit die Züge nicht zusammenstoßen, verlangen wir, dass zwischen ihnen mindestens ein Streckenabschnitt frei bleibt.



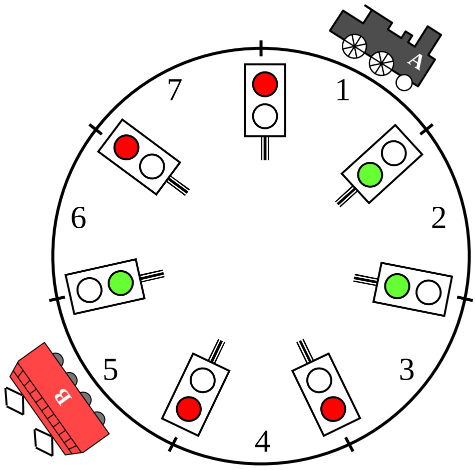
Damit die Züge nicht zusammenstoßen, verlangen wir, dass zwischen ihnen mindestens ein Streckenabschnitt frei bleibt.



Diese Situation darf also nicht eintreten. Wie kann man dies sicherstellen?



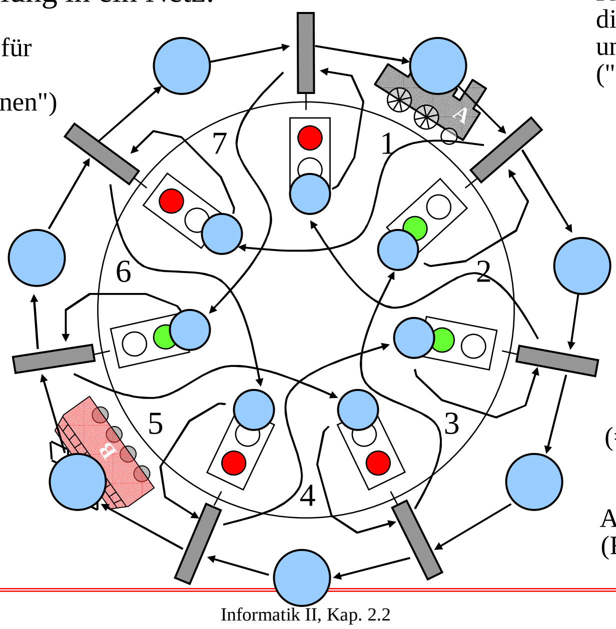
Steuerung durch Ampeln: Grünes Signal bedeutet, dass in den Streckenabschnitt hineingefahren werden darf.



Umwandlung in ein Netz:

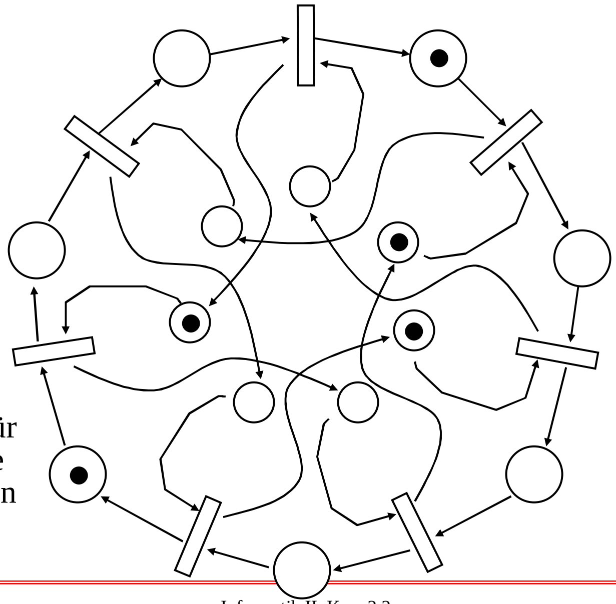
Rechtecke für Übergänge ("Transitionen")

Kreise für die Strecken und Ampeln ("Stellen")



Pfeile (=Kanten) für Voraussetzungen und Auswirkungen (Flussrelation)

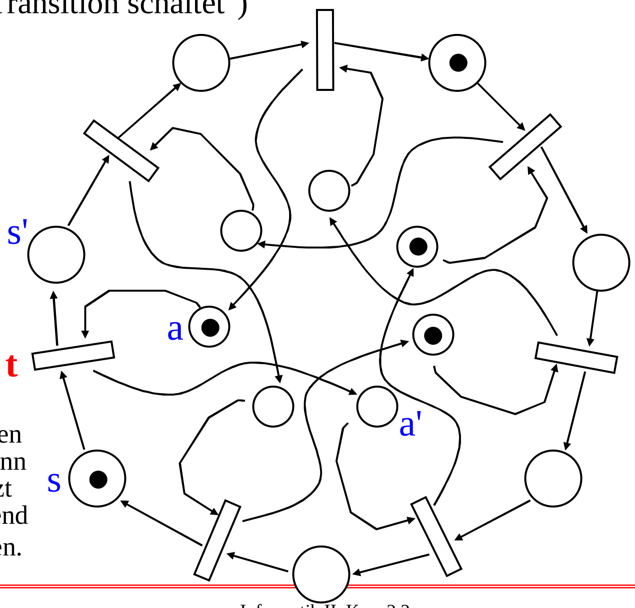
Das resultierende Netz:



Marken für mögliche Aktivitäten einfügen

Eine Aktivität durchführen ("eine Transition schaltet")

*Die Transition **t** "schaltet":*

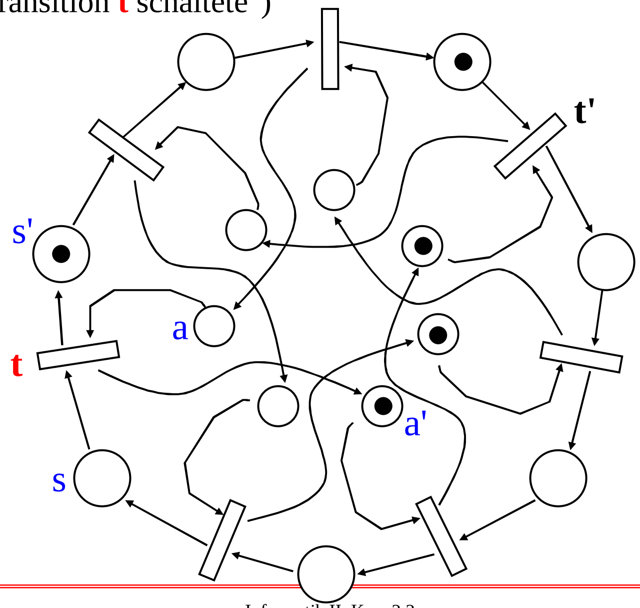


Ein Zug kann von Strecke s nach Strecke s' fahren (= in Stelle s liegt eine Marke). Die Ampel steht auf grün (= in der Stelle a liegt eine Marke). Der Zug fährt nun von s nach s' (= die Transition t schaltet).

Die Marken werden dann umgesetzt entsprechend den Kanten.

Eine Aktivität durchführen
("die Transition **t** schaltete")

Ergebnis des Schaltens:



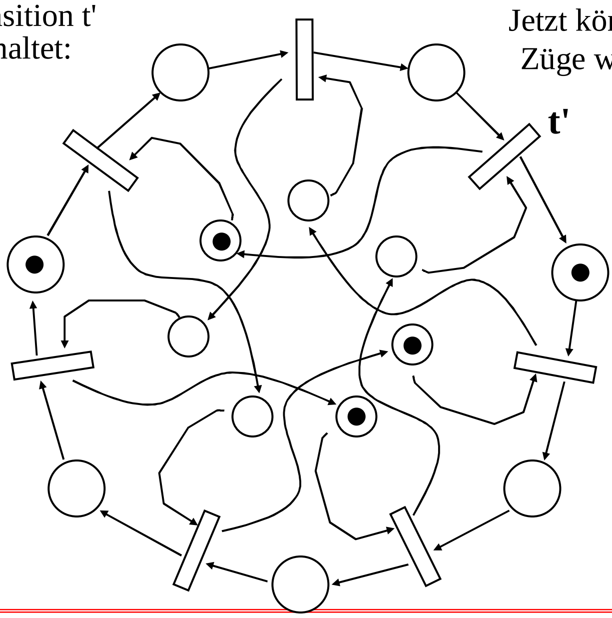
Der Zug ist nun auf der Strecke s' (= in Stelle s' liegt eine Marke). Die Ampel a steht auf rot (= in der Stelle a liegt keine Marke). Dafür wird aber a' auf grün gestellt (= in Stelle a' liegt eine Marke).

Eine Transition t schaltet bedeutet also:
Voraussetzung: Auf allen Stellen, von denen eine Kante nach t führt, muss mindestens eine Marke liegen.
Aktion: Von jeder dieser Stellen wird eine Marke abgezogen. Danach wird zu jeder Stelle, zu der eine Kante von t führt, eine Marke hinzugefügt.

Man nennt dies die "Schaltregel". Wir werden sie im Folgenden exakt definieren.

In unserem Beispiel: Der linke Zug kann nicht weiterfahren, weil die zugehörige Ampel keine Marke enthält. Aber der rechte Zug kann weiterfahren, d.h., die Transition t' kann jetzt schalten. Das Ergebnis (= die neue Verteilung der Marken) finden Sie auf der nächsten Folie.

Die Transition t'
hat geschaltet:



Jetzt könnten beide
Züge weiterfahren
usw.

Definition 2.2.1.2: Stellen-Transitionsnetz (S/T-Netz)

N = (S, T, F, K, W, M_0) heißt Stellen-Transitions-Netz $\Leftrightarrow$

(1) S ist eine endliche Menge (Menge der "Stellen"),

(2) T ist eine endliche Menge (Menge der "Transitionen"),

(3) $F \subseteq (S \times T) \cup (T \times S)$ ist die "Flussrelation" (Kantenmenge),

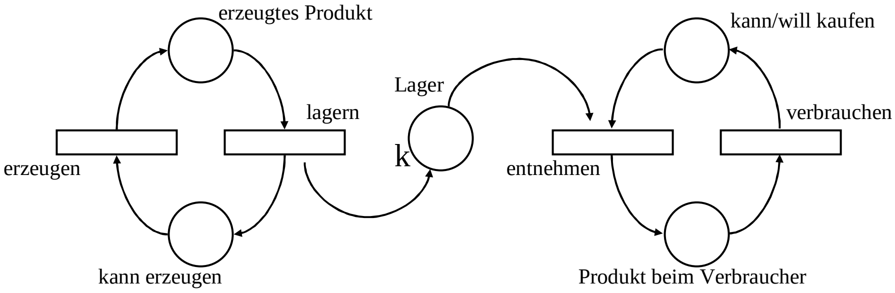
(4) K: $S \rightarrow \mathbb{N} \cup \{\infty\}$ ist die Kapazität für jede Stelle,

(5) W: $F \rightarrow \mathbb{N}$ ist die Gewichtsfunktion ("weight") der Kanten,

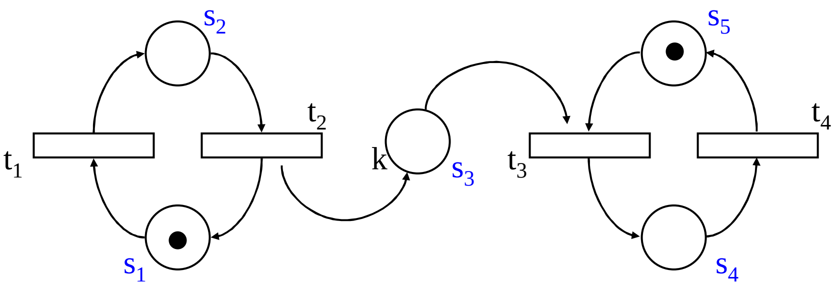
(6) M_0: $S \rightarrow \mathbb{N}_0 \cup \{\infty\}$ ist die Anfangsmarkierung, für die gelten muss: $\forall s \in S$: $M_0(s) \leq K(s)$, d.h., in keiner Stelle dürfen mehr Marken liegen, als die Kapazität zulässt (die Markierungen schreibt man in der Regel als Vektoren).

Beispiel 2.2.1.3: Erzeuger-Verbraucher-Kreislauf (Producer-Consumer-Cycle)

Ein Erzeuger erzeugt ein Produkt, legt dieses in einem Lager, das maximal $k \geq 1$ Stellplätze besitzt, ab und wiederholt diesen Prozess.
Ein Verbraucher entnimmt ein Produkt aus dem Lager, konsumiert dieses und wiederholt diesen Prozess.



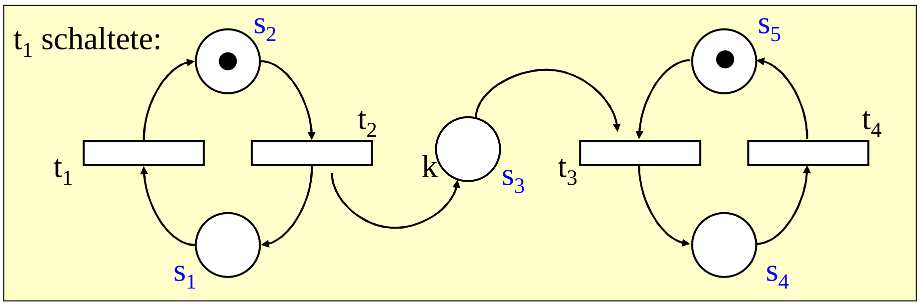
Hinweis: Das Lager hat die Kapazität k, alle anderen Stellen haben die Kapazität ∞.

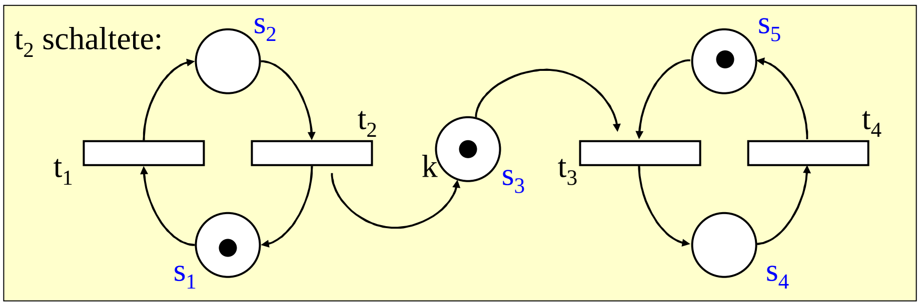


Verbraucher-Erzeuger-System
mit Anfangsmarkierung $M_0 = (1,0,0,0,1)$.

Formal: $S = \{s_1, s_2, s_3, s_4, s_5\}$, $T = \{t_1, t_2, t_3, t_4\}$, $M_0 = (1,0,0,0,1)$,
$F=\{(s_1,t_1),(s_2,t_2),(t_1,s_2),(t_2,s_1),(t_2,s_3),(s_3,t_3),(s_5,t_3),(t_3,s_4),(s_4,t_4),(t_4,s_5)\}$,
$W((x,y))=1$ für alle Kanten (x,y), $K(s_1)=K(s_2)=K(s_4)=K(s_5)=\infty$, $K(s_3)=k$.

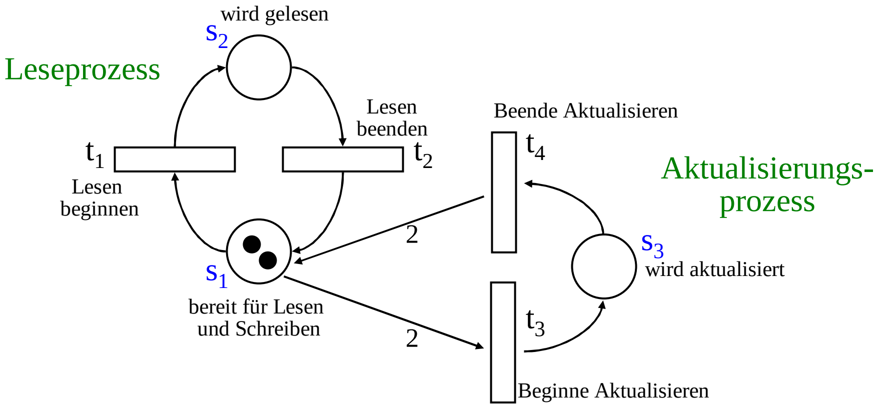
In dieser Anfangssituation kann nur die Transition t_1 schalten.
Aus der Anfangsmarkierung (1,0,0,0,1) ensteht dann die Folge-Markierung (0,1,0,0,1). Nun kann t_2 schalten und es entsteht die Markierung (1,0,1,0,1). Jetzt können t_1 oder t_3 schalten, wobei die Markierungen (0,1,1,0,1) bzw. (1,0,0,1,0) entstehen usw.

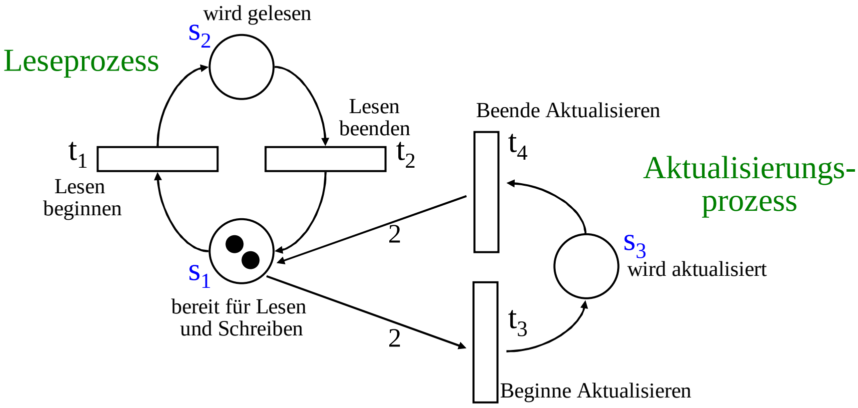




Beispiel 2.2.1.4: Lese-Schreib-Konflikt

Eine Datenbank steht zwei Benutzern zum Lesen zur Verfügung. Ab und zu sollen die Inalte von einem Autor aktualisiert werden; zu diesem Zeitpunkt darf nur der Autor auf die Datenbank zugreifen können. Modell hierzu?



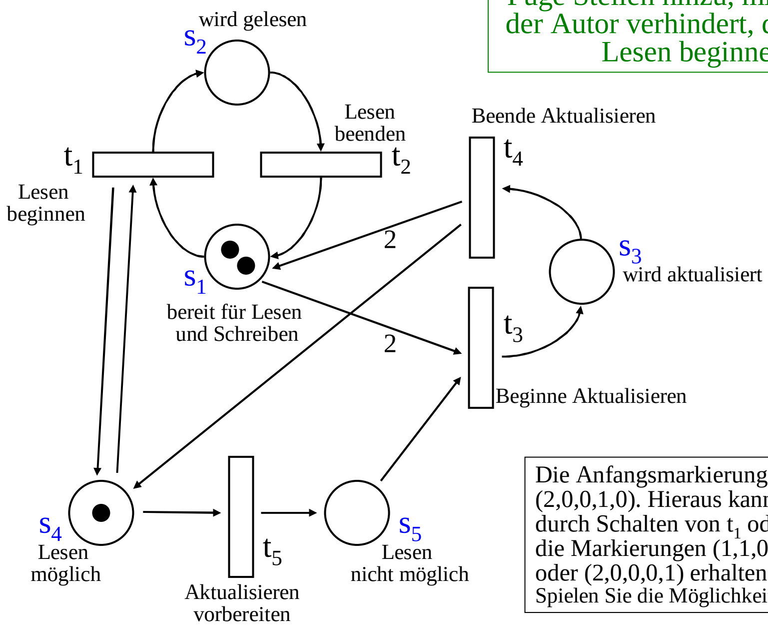


Nun kann eine "unfaire Aktionsfolge" auftreten:

$t_1\ t_1\ t_2\ t_1\ t_2\ t_1\ t_2\ t_1\ t_2\ t_1\ t_2\ \ldots$

Hierbei ist stets mindestens eine Marke in der Stelle s_2, so dass niemals aktualisiert werden kann. Wie kann man erzwingen, dass der Autor das Lesen unterbrechen kann?

Füge Stellen hinzu, mit denen der Autor verhindert, dass das Lesen beginnen kann.



Die Anfangsmarkierung lautet: (2,0,0,1,0). Hieraus kann man durch Schalten von t_1 oder von t_5 die Markierungen (1,1,0,1,0) oder (2,0,0,0,1) erhalten usw. Spielen Sie die Möglichkeiten durch!

Fragen 2.2.1.5 a:

1. Erreichbarkeitsproblem: Wie kann man feststellen, ob eine angestrebte Situation (= Markierung) von einer gegebenen Situation (= Markierung) aus erreicht werden kann?
2. Beschränktheit: Wie kann man nachweisen, dass die Zahl der Marken in allen Stellen beschränkt bleibt?
3. Fairness: Wie kann man ermitteln und sicherstellen, dass keine "unfairen Aktionsfolgen" (= unfaire Folge von schaltenden Transitionen) auftreten kann?
4. Wir kann man beweisen, dass das Netz nicht in "Verklemmungen" gerät, also in eine Markierung, von der aus es nicht mehr weitergeht?
5. Wie stellt man sicher, dass das Netz nicht in "sinnlose Schleifen" (= Iteration von schaltenden Transitionen, die aus Sicht des Problems keinen Sinn machen) gerät?

Um diese Fragen beantworten zu können, müssen wir zuerst die Arbeitsweise und die erforderlichen Begriffe exakt definieren.

Definition 2.2.1.5 b: Begriffe und Schreibweisen

$N = (S, T, F, K, W, M_0)$ sei ein Stellen-Transitions-Netz.

(1) Jede Abbildung $M: S \to \mathbb{N}_0 \cup \{\infty\}$ heißt Markierung von N. M heißt zulässig, wenn für alle $s \in S$ gilt: $M(s) \leq K(s)$.

(2) Eine Markierung schreibt man in der Regel als Spaltenvektor (oder in Texten auch als Zeilenvektor). Hierbei wird vorausgesetzt, dass die Menge der Stellen S geordnet ist: $S = \{s_1, s_2, s_3, \ldots, s_n\}$ mit $s_1 < s_2 < s_3 < \ldots < s_n\}$

(3) Für $x \in S \cup T$ heißen ${}^\bullet x = \{(y,x) \mid (y,x) \in F\}$ der Vorbereich von x und $x^\bullet = \{(x,y) \mid (x,y) \in F\}$ der Nachbereich von x.

(4) Die Flussrelation $F \subseteq (S \times T) \cup (T \times S)$ zusammen mit der Gewichtsfunktion W wird meist auf die gesamte Menge $(S \times T) \cup (T \times S)$ wie folgt fortgesetzt zu $W': S \to \mathbb{N}_0$
$W'((x,y)) :=$ <u>if</u> $(x,y) \in F$ <u>then</u> $W((x,y))$ <u>else</u> 0 <u>fi</u>.
(W' beschreibt F und W eindeutig.)

Definition 2.2.1.6: Arbeitsweise von S/T-Netzen

$N = (S, T, F, K, W, M_0)$ sei ein Stellen-Transitions-Netz.

(1) Eine Transition $t \in T$ heißt unter der Markierung M aktiviert
$\Leftrightarrow \forall s \in {}^\bullet t: W((s,t)) \leq M(s)$ und $\forall s \in t^\bullet: W((t,s)) + M(s) \leq K(s)$.
Man schreibt hierfür auch: $M[\,t>$

(2) *Schaltregel:* Es sei t eine Transition, die unter der zulässigen Markierung M aktiviert ist. Dann kann t schalten und es entsteht aus M die Folge-Markierung M' mit:

$$M'(s) = \begin{cases} M(s) & s \notin {}^\bullet t \text{ und } s \notin t^\bullet, \\ M(s) - W((s,t)) & s \in {}^\bullet t \text{ und } s \notin t^\bullet, \\ M(s) + W((t,s)) & s \notin {}^\bullet t \text{ und } s \in t^\bullet, \\ M(s) - W((s,t)) + W((t,s)) & s \in {}^\bullet t \text{ und } s \in t^\bullet. \end{cases}$$

[Wir hätten auch $M'(s) = M(s) - W'((s,t)) + W'((t,s)), \forall s \in S$ schreiben können, siehe Definition 2.2.1.5 (4).]

noch Definition 2.2.1.6:

(3) Schreibweise: Wenn t unter M aktiviert ist und nach dem Schalten von t die Markierung M' entsteht, so schreibt man $M\,[\,t>M'$ oder $M \xrightarrow{t} M'$.

(4) Fortsetzung der Relation $\cdot[\,>$ auf Folgen von Transitionen (also auf T*; Wörter über T nennen wir auch Schaltfolgen):
$M[\,t_1 t_2 t_3 \ldots t_r> \;\Leftrightarrow$
es gibt Markierungen $M_1, M_2, \ldots, M_{r-1}$ mit $M[\,t_1>M_1$,
$M_1[\,t_2>M_2$, $M_2[\,t_3>M_3$, ..., $M_{r-2}[\,t_{r-1}>M_{r-1}$, $M_{r-1}[\,t_r>$.

noch Definition 2.2.1.6: Aktiviertheit und Erreichbarkeit

(5) Fortsetzung der Relation $\cdot[>\cdot$ auf Schaltfolgen (also auf Folgen von Transitionen):
$M[t_1t_2t_3...t_r > M'$ ⇔ es gibt Markierungen $M_1, M_2, ..., M_{r-1}$ mit $M[t_1 > M_1$, $M_1[t_2 > M_2$, $M_2[t_3 > M_3$, ..., $M_{r-1}[t_r > M'$.
(Im Falle r=0 muss M=M' sein.)

(6) ERR(M) = $\{M' \mid$ es existiert $t_1t_2t_3...t_r$ mit $M[t_1t_2t_3...t_r > M'\}$ heißt Erreichbarkeitsmenge bzgl. M.
(Beachte: $M \in$ ERR(M), insbesondere ist ERR(M) nie leer.)

ERR(N) = $\{M' \mid$ es existiert $t_1t_2t_3...t_r$ mit $M_0[t_1t_2t_3...t_r > M'\}$ heißt Erreichbarkeitsmenge des Netzes N.

Wenn M' in ERR(M) liegt, so sagt man auch, M' ist von M aus erreichbar.

Die Erreichbarkeit stellt die Bedeutung der S/T-Netze dar. Kennt man also den Erreichbarkeitsgraphen (siehe unten) eines S/T-Netzes im Detail, so kann man hieraus alle seine Eigenschaften ableiten. Wir werden dies an den Begriffen Beschränktheit, Lebendigkeit und Fairness demonstrieren.

Hinweise:

Oft schreibt man S/T-Netze nur in der Form N = (S, T, F, M_0).
In diesem Fall ist $K(s) = \infty$ für alle Stellen s und $W((x,y)) = 1$ für alle Kanten (x,y) einzusetzen.
Dies gilt auch für Zeichnungen: Fehlen die Angaben für K oder W an einer Stelle bzw. Kante, so ist unbeschränkte Kapazität bzw. Kantengewicht 1 gemeint.

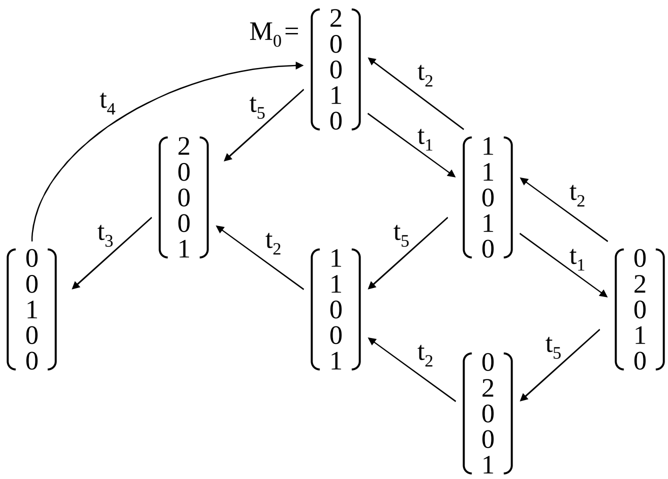
In diesem Abschnitt sprechen wir oftmals nur von einem "Netz" und meinen damit stets ein S/T-Netz.

Um die Arbeitsweise eines Netzes im Ganzen zu verstehen, konstruiert man schrittweise alle Markierungen, die man von der Anfangsmarkierung M_0 aus erreichen kann. Das Ergebnis ist der "Erreichbarkeitsgraph" des Netzes.

Dieser Graph kann unendlich oder endlich sein. Unter den endlichen Graphen interessieren vor allem diejenigen, die nicht allzu groß werden; also: Wenn d die Länge der Darstellung eines Netzes ist, dann soll die Länge der Darstellung des Erreichbarkeitsgraphen höchstens polynomiell bzgl. d sein.

Leider ist dies jedoch nur selten der Fall. Wir betrachten hierzu einige Beispiele und definieren den Erreichsbarkeitsgraphen eines Netzes formal.

2.2.1.7: Konstruktion des Erreichbarkeitsgraphen des letzten S/T-Netzes aus Beispiel 2.2.1.4: Ausgehend von M_0 werden suksessiv alle im nächsten Schritt erreichbaren Markierungen notiert:

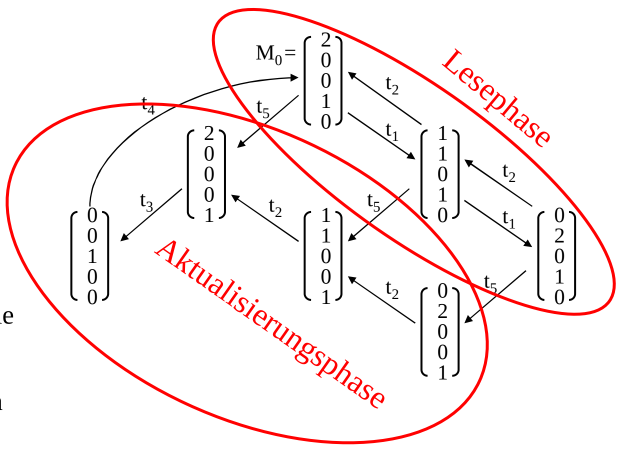


Dieser Graph besitzt einige Eigenschaften, die uns Hinweise geben, ob das angegebene Netz unser gestelltes Lese-Schreib-Problem tatsächlich löst:

1. Man erkennt die (korrekte) Trennung zwischen Lesen und Schreiben

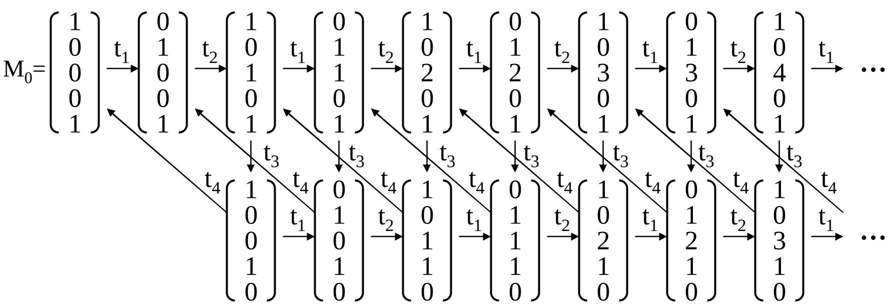
2. Man entdeckt eine "Invariante" der erreichbaren Markierungen M: $(1,1,3,1,1)\cdot M = 3$.

3. Man sieht, dass die Transitionenfolgen $(t_1t_2)^*$ und $(t_5t_3t_4)^*$ korrekt von M_0 nach M_0 führen.



4. Man erkennt, dass jede Transition irgendwann noch einmal schalten kann, dass also keine Transition irgendwann überflüssig wird.

2.2.1.8: Konstruktion des Erreichbarkeitsgraphen aus 2.2.1.3



Falls die in 2.2.1.3 angegebene Größe k eine natürliche Zahl ist, so besitzt der Erreichbarkeitsgraph genau 4k+2 Markierungen. Ist $k = \infty$, so ist der Ereichbarkeitsgraph unendlich groß.

Definition 2.2.1.9: Erreichbarkeitsgraph.
Es sei $N = (S, T, F, K, W, M_0)$ ein Stellen-Transitions-Netz mit der Erreichbarkeitsmenge ERR(N).

Der Erreichbarkeitsgraph G(N) des Netzes N ist ein gerichteter Graph mit der Knotenmenge ERR(N). Er besitzt in der Regel Mehrfachkanten, die dann aber mit *verschiedenen* Transitionen beschriftet sind. Eine Kante (M, M') mit der Beschriftung t existiert genau dann, wenn $M[t>M'$ gilt.

Formal: $G(N) = \big(ERR(N), \{(M, t, M') \mid M[t>M'\}\big)$, wobei (M, t, M') eine Kante von der Markierung M zur Markierung M' mit Beschriftung t ist.

(M, t, M') zeichnet man in der Form $M \xrightarrow{t} M'$

[Hinweis: Zu jedem Netz gibt es genau einen Erreichbarkeitsgraphen. Es ist klar, wie man ihn schrittweise aufbaut.]

2.2.1.10: Beschränktheit eines Netzes

Ein Netz soll beschränkt heißen, wenn es eine natürliche Zahl k gibt, so dass keine Markierung im Netz erreicht werden, in der eine Stelle mehr als k Marken besitzt.

Definition: Sei $N = (S, T, F, K, W, M_0)$ ein S/T-Netz.

(1) Es sei k eine natürliche Zahl. Eine Stelle s des Netzes N heißt k-beschränkt, wenn für jede von M_0 aus erreichbare Markierung M gilt, M(s) ist nicht größer als k, d.h., $\forall M \in ERR(N): M(s) \leq k$.
(2) N heißt k-beschränkt, wenn jede Stelle k-beschränkt ist, d.h., $\forall s \in S\ \forall M \in ERR(N): M(s) \leq k$.
(3) s heißt beschränkt, wenn es eine natürliche Zahl k gibt, so dass s k-beschränkt ist, d.h., $\exists k \in \mathbb{N}\ \forall M \in ERR(N): M(s) \leq k$.
(4) N heißt beschränkt, wenn jede Stelle von N beschränkt ist, d.h., $\exists k \in \mathbb{N}\ \forall s \in S\ \forall M \in ERR(N): M(s) \leq k$.

2.2.1.11: Folgerung

Ein Netz ist genau dann beschränkt, wenn sein Erreichbarkeitsgraph endlich ist.

Beweis: Sei S die Menge der Stellen des Netzes N.

"$\Rightarrow$" Wenn N beschränkt ist, so gibt es ein k, so dass alle Markierungen des Erreichbarkeitsgraphen nur Komponenten besitzen, die kleiner oder gleich k sind. Dann kann es aber höchstens $(k+1)^{|S|}$ Markierungen in ERR(N) geben, d.h., der Erreichbarkeitsgraph ist endlich.

"$\Leftarrow$" Wenn der Erreichbarkeitsgraph endlich ist, dann existiert das Maximum m über alle Komponenten von Markierungen in ERR(N). Offenbar ist jede Stelle dann m-beschränkt, d.h., das Netz ist beschränkt.

2.2.1.12: Größe eines Netzes. Wie groß können Erreichbarkeitsgraphen werden, bezogen auf die Größe des zugehörigen Netzes?

Hierzu müssen wir zunächst festlegen, was die Größe eines Netzes ist. Wie üblich ist dies die Länge einer Darstellung.

Meist wählt man eine normierte Darstellung. Eine relativ kurze Darstellung ist z.B. die Folgende, bei der die Stellen stets mit den Zahlen von 1 bis n = |S| und die Transitionen mit den Zahlen von 1 bis m = |T| bezeichnet werden; die i-te Stelle beschreibt man, indem man vor die Nummer i ein 's' setzt; analog sei 't'j die j-te Transition:

<Zahl der Stellen>; <Zahl der Transitionen>;
<Liste der Kanten in der Form (<Bezeichnung Knoten>, <Bezeichnung Knoten>, <Gewicht der Kante>) >;
<Kapazitäten als n-stelliger Vektor>;
<Anfangsmarkierung als n-stelliger Vektor>;;

Hierbei werden alle Zahlen binär aufgeschrieben.

Definition: Die Größe des Netzes ist Länge dieser Darstellung.

Ein Netz wird hier also als ein Wort über dem 7-elementigen Alphabet {s, t, 0, 1, ,, ;, ∞} aufgefasst.

Beispiel: In Beispiel 2.2.1.3 hatten wir folgendes Netz vorgestellt:
$S = \{s_1, s_2, s_3, s_4, s_5\}$, $T = \{t_1, t_2, t_3, t_4\}$, $M_0 = (1,0,0,0,1)$,
$F=\{(s_1,t_1),(s_2,t_2),(t_1,s_2),(t_2,s_1),(t_2,s_3),(s_3,t_3),(s_5,t_3),(t_3,s_4),(s_4,t_4),(t_4,s_5)\}$,
$W((x,y))=1$ für alle Kanten (x,y), $K(s_1)=K(s_2)=K(s_4)=K(s_5)=\infty$, $K(s_3)=k$.

Dieses Netz schreiben wir also für k=9 in folgender Form auf:

101;100;s1,t1,1,s10,t10,1,t1,s10,1,t10,s1,1,t10,s11,1,s11,t11,1,
s101,t11,1,t11,s100,1,s100,t100,1,t100,s101,1;∞,∞,1001,∞,∞;
1,0,0,0,1;;

Dieses Wort besteht aus 134 Zeichen. Unser Netz besitzt also die Größe 134.

Übungsaufgaben:

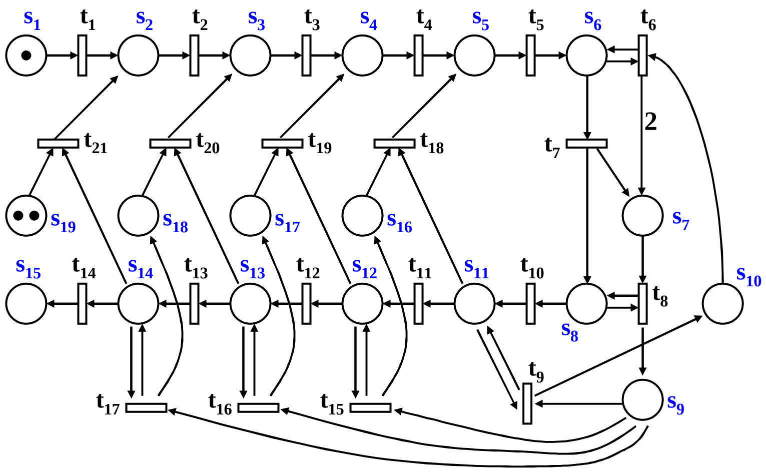
1. Welche Größen haben das "resultierende Netz" des Zug-Beispiels 2.2.1.1 (siehe Folie Nr. 9) und das letzte Netz aus Beispiel 2.2.1.4 (mit 5 Stellen und 4 Transitionen, vgl. Folie Nr. 20)?
2. Definieren Sie auf ähnliche Weise die Größe eines Erreichbarkeitsgraphen.
3. Welche Größen haben die Erreichbarkeitsgraphen der drei obigen Netze, für die die Größe bestimmt wurde? (Für zwei Beispiele wurden in 2.2.1.7 und 2.2.1.8 die Erreichbarkeits-graphen bereits angegeben; für das Zugbeispiel müssen Sie diesen Graphen selbst konstruieren.)

4. Definieren Sie, was es bedeuten soll, dass eine Klasse von Netzen polynomiell konstruierbare Erreichbarkeitsgraphen besitzt. Wieviel Platz benötigt man für diese Konstruktionen höchstens?
5. Geben Sie unendlich große Klassen von Netzen an, deren Erreichbarkeitsgraphen sich genau mit linearem, bzw. mit quadratischem Zeitaufwand konstruieren lassen.
6. Es gibt Netze, deren Erreichbarkeitsgraphen exponentiell größer sind als die Netze selbst. Versuchen Sie, solche Netze selbst zu entdecken, oder durchsuchen Sie die Literatur hiernach. (Hinweis: Wir haben eine Bibliothek im Erdgeschoss.)
7. Gibt es Klassen von Netzen, deren Erreichbarkeitsgraphen endlich sind, aber viel stärker als exponentiell wachsen?

Warnung zu Punkt 7: Die Erreichbarkeitsgraphen von S/T-Netzen können gewaltig wachsen.

Das S/T-Netz auf der folgenden Folie mit 19 Stellen und 21 Transitionen vollzieht die Berechnung der sog. Ackermann-Funktion A (in der fünften Stufe) nach. Dies ist eine totale berechenbare Funktion A: $\mathbb{N}_0 \times \mathbb{N}_0 \rightarrow \mathbb{N}_0$, die schneller als jede Funktion wächst, die man nur mit elementaren Anweisungen, der Sequenz, der Alternative und der for-Schleife darstellen kann. Der Erreichbarkeitsgraph dieses S/T-Netzes ist endlich, besitzt aber mehr als

$2^{\underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{65535 \text{ mal}}}$ Knoten. Vollziehen Sie etwa 100 Schritte nach, um die Wirkungsweise des Netzes zu erahnen.



2.2.1.13: Lebendigkeit eines Netzes

Ein Netz soll lebendig heißen, wenn jede Transition irgendwann noch einmal schalten könnte. Genauer: Für jede Transition t muss gelten: Wenn man sich, ausgehend von M_0, in irgendeiner Markierung M befindet, dann muss von M aus eine Markierung M' erreichbar sein, unter der t aktiviert ist (also schalten kann).

Definition: Sei N = (S, T, F, K, W, M_0) ein S/T-Netz.

(1) Eine Transition t des Netzes N heißt lebendig, wenn es zu jeder von M_0 aus erreichbaren Markierung M eine von M aus erreichbare Markierung M' mit M' gibt, d.h.,
$\forall M \in ERR(N)\ \exists M' \in ERR(M): M'[t>$.

(2) N heißt (stark) lebendig, wenn jede Transition von N lebendig ist.

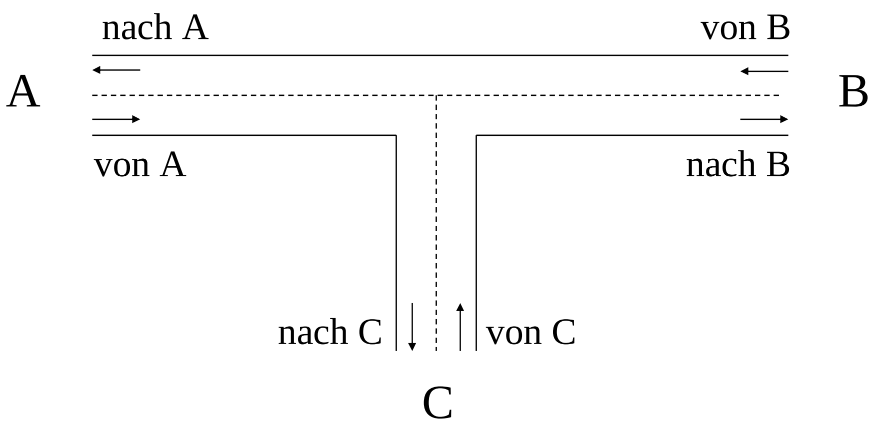
Man könnte ein Netz auch lebendig nennen, wenn es immer weiterschalten kann. Dies ist gleichbedeutend damit, dass es keine von M_0 aus erreichbare Markierung gibt, zu der es keine Folge-Markierung gibt. Solche Markierungen bezeichnet man als Verklemmung.

Fortsetzung der Definition:

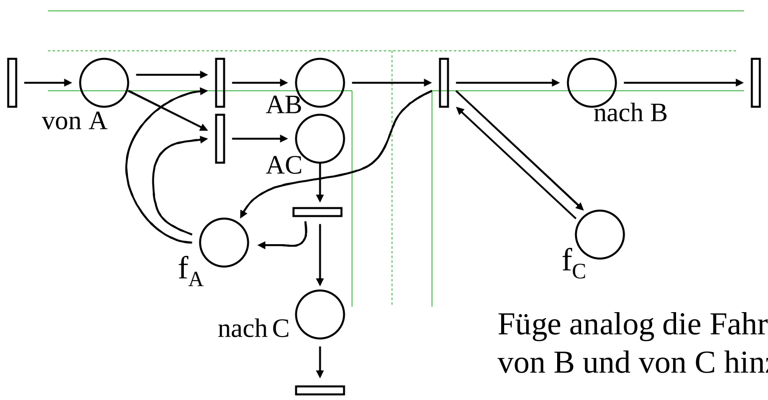
(3) Eine Markierung M heißt Verklemmung ("Deadlock"), wenn unter M keine Transition aktiviert ist, d.h.
$\neg \exists t \in T: M[t>$.

(4) Das Netz N heißt schwach lebendig, wenn es keine von M_0 aus erreichbare Verklemmung besitzt, d.h.
$\forall M \in ERR(N)\ \exists t \in T: M[t>$.

2.2.1.14: Beispiel:
Straßenkreuzung mit Vorfahrtsregel "rechts vor links".

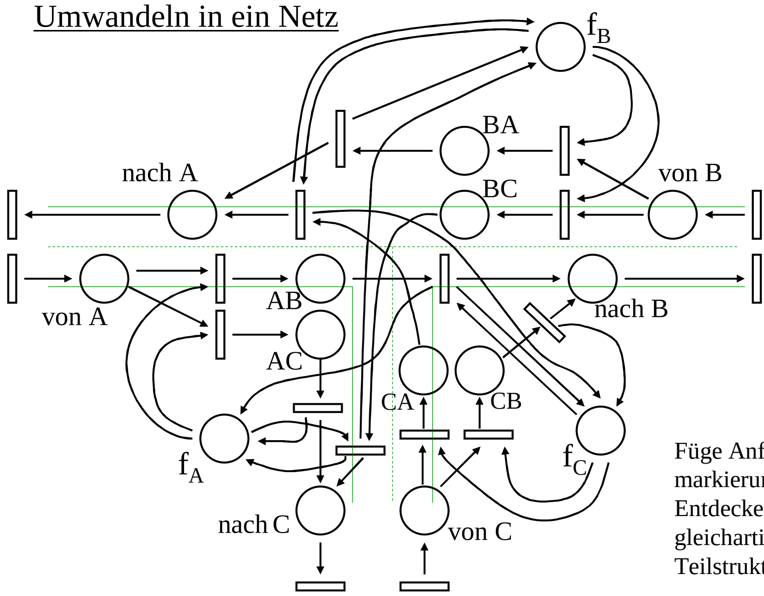


Umwandeln in ein Netz, wobei explizit "von rechts kommt niemand" durch die Stellen f modelliert wird. Wir betrachten zunächst nur die von A kommenden Fahrzeuge, für die nur wichtig ist, ob die Strecke von C zur Kreuzung frei ist (f_C).



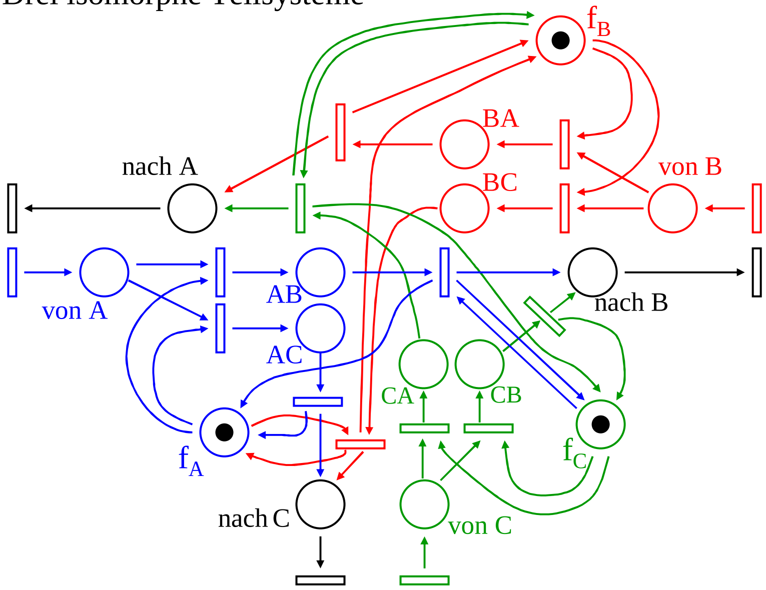
Füge analog die Fahrten von B und von C hinzu.

Umwandeln in ein Netz

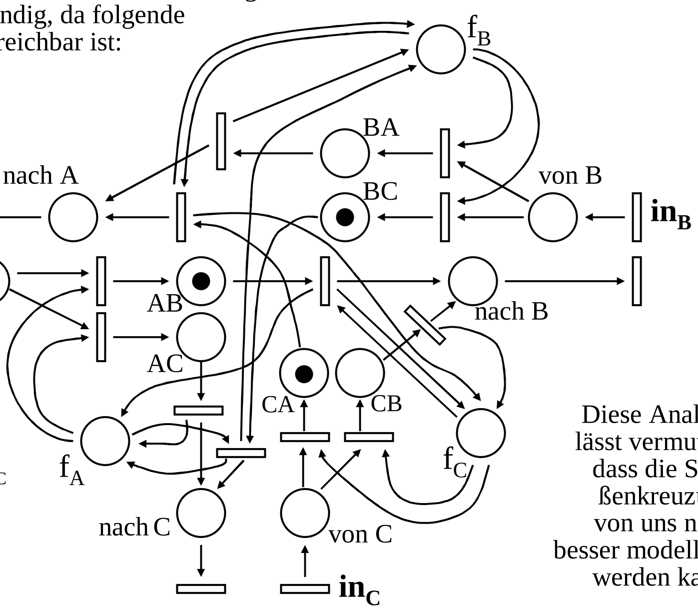


Füge Anfangs-markierung hinzu. Entdecke gleichartige Teilstrukturen:

Drei isomorphe Teilsysteme



Dieses Netz ist zwar schwach lebendig, aber nicht lebendig, da folgende Markierung erreichbar ist:

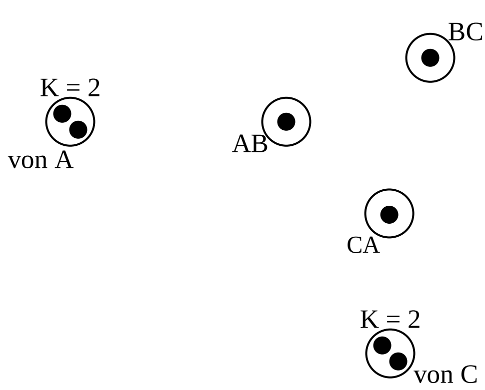


Nun können zwar noch die Transitionen in_A, in_B und in_C schalten, aber alle anderen Transitionen nicht mehr.

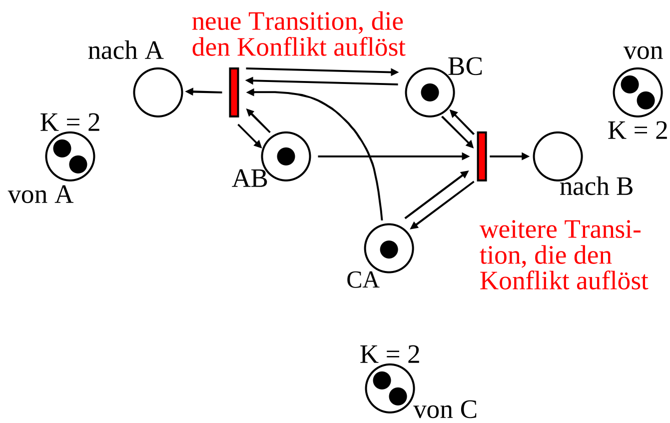
Diese Analyse lässt vermuten, dass die Straßenkreuzung von uns noch besser modelliert werden kann.

Wir sollten die Eingangs-Stellen "von A", "von B" und "von C" mit einer Kapazität, z.B. mit 2 belegen.

In diesem Fall gibt es eine Verklemmung, nämlich (wir zeigen nur den relevanten Ausschnitt aus dem Netz, also die Stellen der Verklemmung, die noch Marken tragen):



Diese Verklemmung können wir beseitigen, indem wir eine Transition einfügen, die in diesem Fall z.B. das von C nach A fahrende Fahrzeug als erstes über die Kreuzung lässt:



Wir können zusätzlich eine Transition hinzufügen, die das von A nach B fahrende Fahrzeug bevorzugt.

Als drittes können wir eine Transition hinzufügen, die das von B nach C fahrende Fahrzeug als erstes über die Kreuzung lässt.

Dies führt zu einem Netz, das relativ realistisch die Situationen an einer Kreuzung widerspiegelt. Es enthält alle Möglichkeiten, eine Verklemmung aufzulösen und unsinnige Überlastungen in gewissen Teilen des Netzes zu vermeiden.

Die Leser(innen) mögen diese Einzelheiten in das bereits vorhandene Netz eintragen und das neue Netz untersuchen.

Ende Beispiel 2.2.1.14

2.2.1.15: Wirkung des Schaltens einer Transition t_j:

$$\Delta t_j = \begin{pmatrix} W'((t_j,s_1)) - W'((s_1,t_j)) \\ W'((t_j,s_2)) - W'((s_2,t_j)) \\ W'((t_j,s_3)) - W'((s_3,t_j)) \\ \dots \\ W'((t_j,s_n)) - W'((s_n,t_j)) \end{pmatrix} \in \mathbb{N}_0^n \quad \text{mit } n = |S|$$

Wenn t_j schaltet, wird die Markierung genau um diesen Vektor verändert, d.h.: Aus $M\,[\,t_j > M'$ folgt $M' = M + \Delta t_j$.

Wir übertragen diese Aussage nun auf eine Folge von Transitionen. Hierzu sei $T = \{t_1, t_2, ..., t_m\}$.

Hierzu definieren wir für $w \in T^*$:

$\#_j w$ = Anzahl der t_j, die in w vorkommen.

Formal: $\#_j\varepsilon = 0$ und $\#_j vt = \underline{\text{if}}\ t_j = t\ \underline{\text{then}}\ \#_j v + 1\ \underline{\text{else}}\ \#_j v\ \underline{\text{fi}}$ $(\forall v \in T^*, \forall t \in T)$.

Dann gilt für jede Folge w von Transitionen mit M[w>M':

$M' = M + \#_1 w \cdot \Delta t_1 + \#_2 w \cdot \Delta t_2 + \#_3 w \cdot \Delta t_3 \ldots + \#_m w \cdot \Delta t_m$.

Dies formulieren wir nun in Matrizenschreibweise. Hierzu sei #w der (Spalten-) Vektor, dessen Komponenten $\#_1 w$, $\#_2 w$, ..., $\#_m w$ sind:

$$\#w = \begin{pmatrix} \#_1 w \\ \#_2 w \\ \#_3 w \\ \ldots \\ \#_m w \end{pmatrix}$$

<u>2.2.1.16 Definition</u>:

Gegeben sei ein Netz N = (S, T, F, K, W, M_0). Es seien $S = \{s_1, s_2, \ldots, s_n\}$ und $T = \{t_1, t_2, \ldots, t_m\}$. Die (n,m)-Matrix

$$C = (\Delta t_1, \Delta t_2, \Delta t_3, \ldots, \Delta t_m)$$

heißt Inzidenzmatrix des Netzes N.

<u>2.2.1.17 Folgerung</u>:

Für alle Markierungen M und M' und für alle Schaltfolgen $w \in T^*$ gilt:

$\boxed{\text{Aus M[w>M' folgt } M' = M + C \cdot \#w.}$

2.2.1.18 Definition:

(1) Jeder (Zeilen-) Vektor $y \in \mathbf{Z}^n$ mit $y \cdot C = 0$ heißt S-Invariante.

(2) y heißt echte S-Invariante $\Leftrightarrow$ y ist eine S-Invariante mit nichtnegativen Komponenten und $y \neq 0$.

(3) Jeder (Spalten-) Vektor $x \in \mathbf{Z}^m$ mit $C \cdot x = 0$ heißt T-Invariante.

Beachte 2.2.1.17, so sieht man: Eine T-Invariante gibt an, wie oft die einzelnen Transitionen schalten müssen, damit die ursprüngliche Markierung wieder hergestellt wird. Eine S-Invariante y besagt, dass der Wert $y \cdot M = y \cdot M'$ für alle von M aus erreichbaren Markierungen M' gleich ist.

2.2.1.19 Satz:

Sei N = (S, T, F, K, W, M_0) ein Netz.

(1) Sei y eine S-Invariante, dann gilt für alle $M' \in ERR(M)$: $y \cdot M = y \cdot M'$.

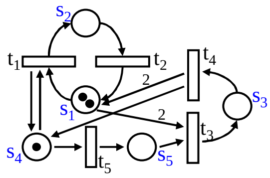
(2) Sei y eine echte S-Invariante. dann gilt für jede Stelle s von N, deren zugehörige Komponente in y positiv ist: s ist k-beschränkt für $k = y \cdot M_0$.

(3) Wenn es eine Markierung M und eine Schaltfolge $w \in T^*$ mit M[w>M gibt, dann ist #w eine T-Invariante von N.

Dieser Satz folgt unmittelbar aus den bisherigen Ausführungen. Bei (2) beachte man für die Komponente M(s):
$M(s) \leq y(s) \cdot M(s) \leq y_1 \cdot M_1 + y_2 \cdot M_2 + \ldots + y_n \cdot M_n = y \cdot M = y \cdot M_0 = k$
Aus der linearen Algebra wissen wir, dass die Menge der S- bzw. T-Invarianten einen Untervektorraum bildet.

Anwenden auf das Beispiel aus 2.2.1.4:

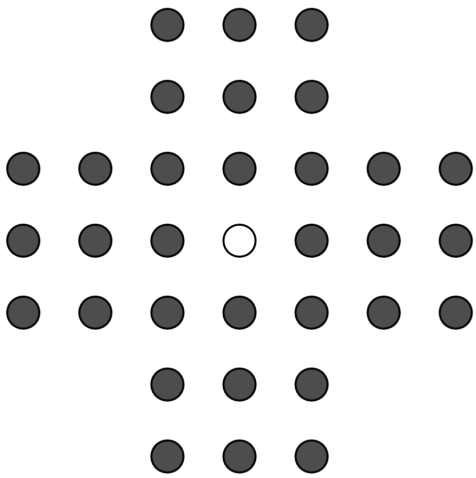
$$C = \begin{pmatrix} -1 & 1 & -2 & 2 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 0 & 1 \end{pmatrix} \quad M_0 = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$



Eine echte S-Invariante lautet: y = (1, 1, 3, 1, 1). Nach Satz 2.2.1.19 (2) sind daher alle Stellen 3-beschränkt und somit ist auch das Netz beschränkt, siehe auch Beispiel 2.2.1.7.

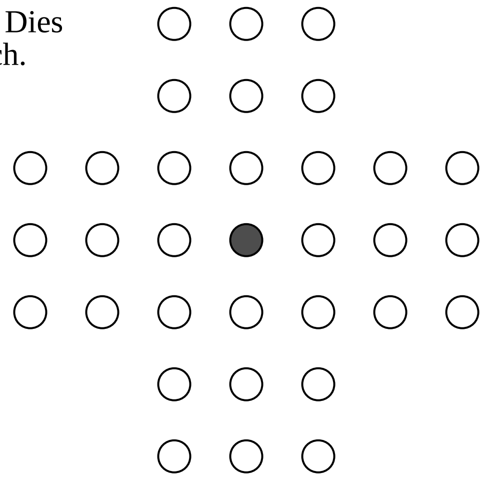
(1,1,0,0,0) und (0,0,1,1,1) sind T-Invarianten, daher verändern die Schaltfolgen t_1t_2 oder t_2t_1 bzw. $t_3t_4t_5$ oder $t_3t_5t_4$ oder $t_4t_3t_5$ oder $t_4t_5t_3$ oder $t_5t_3t_4$ oder $t_5t_4t_3$ (sofern sie schalten können) die Markierung nicht.

<u>Beispiel 2.2.1.20</u>: Solitaire-Spiel

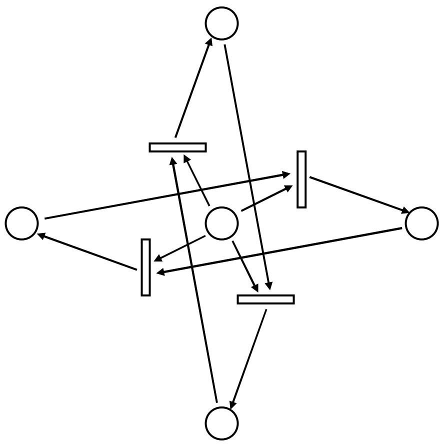


33 Plätze, 32 Steine, d.h., ein Platz bleibt frei. Ein Zug = Springe in gerader Richtung (nicht diagonal) über einen Nachbarstein auf einen freien Platz und entferne den übersprungenen Stein.
Ziel: Am Ende soll nur noch ein Stein (möglichst auf einem vorgegebenen Platz) übrig sein.

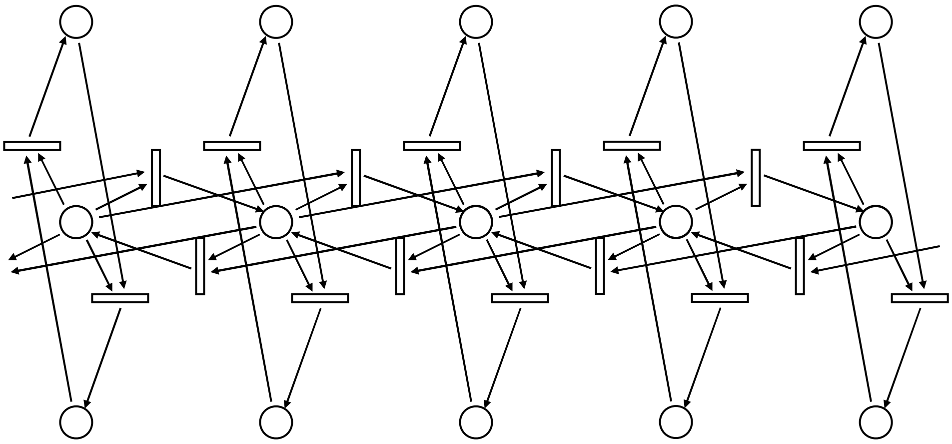
Meist versucht man, diese Endsituation zu erreichen. Dies geht tatächlich.

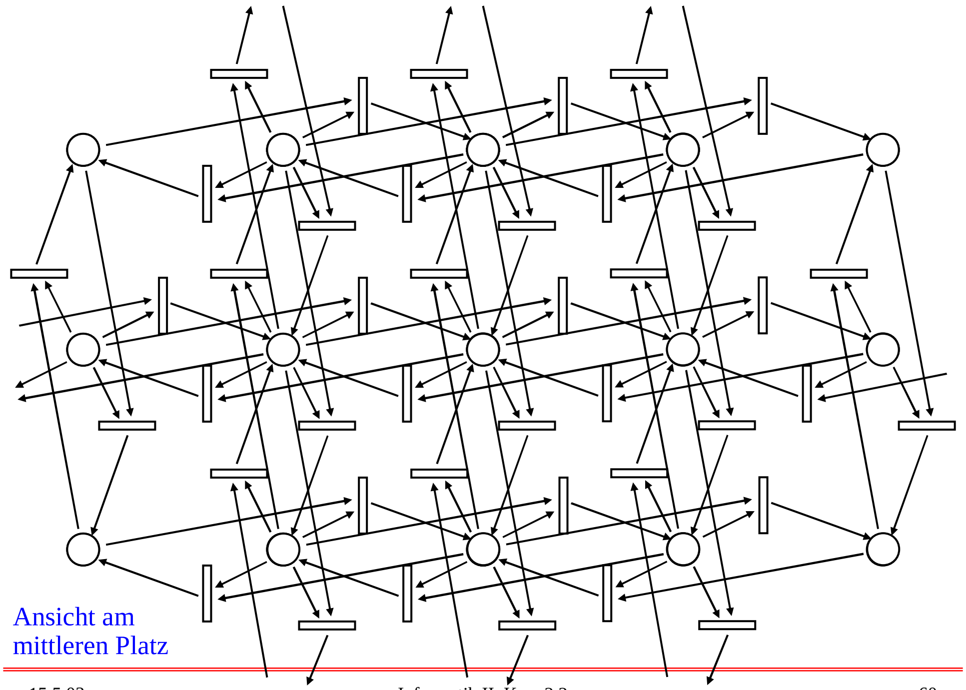


Solitaire als S/T-Netz formulieren. Betrachte einen Platz:

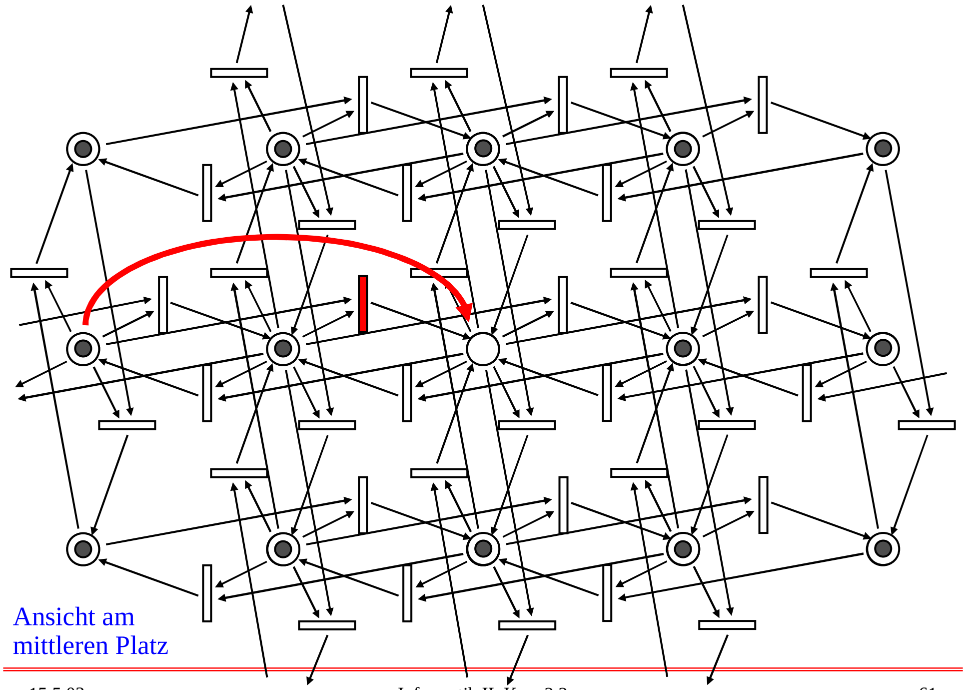


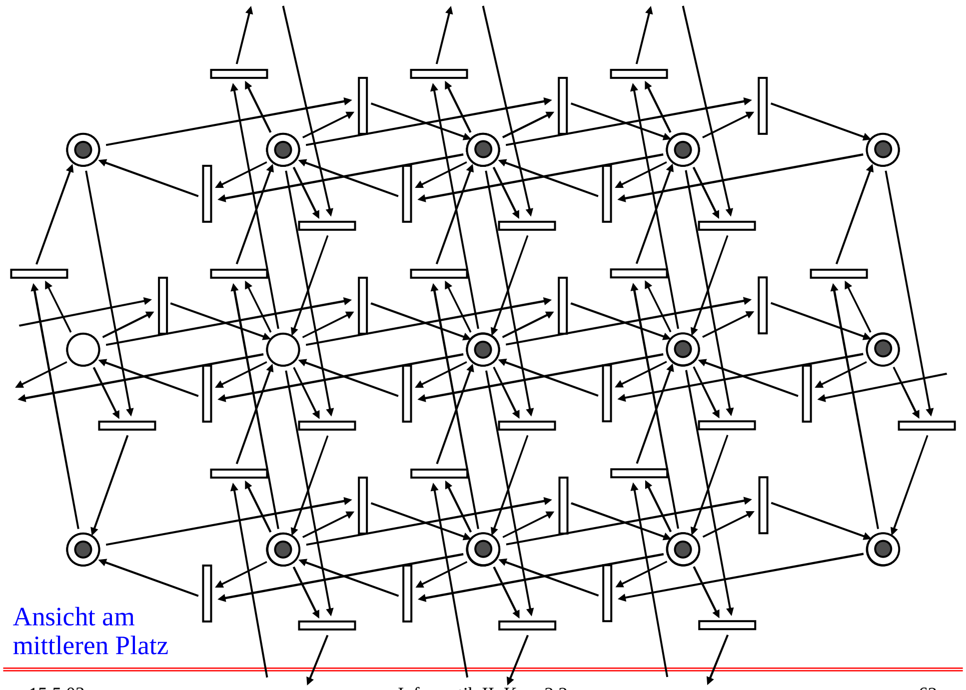
Plätze werden also Stellen, Transitionen sind das Überspringen.
Dieses Muster muss nun an jeder Stelle eingefügt werden.





Ansicht am
mittleren Platz


Ansicht am
mittleren Platz


Ansicht am
mittleren Platz

Das Ziel des Spiels besteht nun darin nachzuweisen, dass von der Anfangsmarkierung (1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1) eine vorgegebene Markierung, zum Beispiel (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), erreicht werden kann.

Vergleiche hierzu die Fragen 2.2.1.5a. Hinweise dazu:
Die Erreichbarkeit ist entscheidbar, allerdings i. A. nur mit einem gewaltigen Zeitaufwand.
Das gleiche gilt für die Beschränktheit. (Beachten Sie, dass Satz 2.2.1.19 nur ein hinreichendes Kriterium enthält. Der Nachweis der Beschränktheit erfolgt mit sog. Überdeckungsgraphen. Diese kann man auch für die Untersuchung der Lebendigkeit einsetzen.)
Fairness haben wir hier nicht definiert. Das Problem, ob ein beliebiges S/T-Netz nur faire Schaltfolgen besitzt, ist nicht entscheidbar.

Umfangreiche Aufgabe zum Üben:

(1) Schreiben ein Paket "ST_Netz" in Ada, das die Stellen-Transitionsnetze darstellt.

(2) Formulieren Sie hierzu eine Prozedur, die zu einem gegebenen S/T-Netz den Erreichbarkeitsgraphen (bis zu einer vorzugebenden Größe) aufbaut.

(3) Implementieren Sie die Aufstellung der Inzidenzmatrix und die Berechnung von S- und T-Invarianten (sofern sie existieren).

(4) Schlagen Sie in der Literatur nach und implementieren Sie weitere Analyseverfahren (z.B. den Aufbau eines Überdeckungsgraphen, den Farkas-Algorithmus, den Nachweis der Erreichbarkeit mittels Backtracking usw.).