

Studiengang Informatik
Fachbereich 3: Mathematik und Informatik

Bachelorarbeit
**Reduktionen von Entscheidungsproblemen
für Graphen auf das
Erreichbarkeitsproblem für Petri-Netze**

Autor: Brice Arnaud Habenicht geb. Foamoukom Nono

Matrikelnummer: 2260395

Version vom: 18. April 2018

1. Gutachterin: Dr. Sabine Kuske
2. Gutachter: Prof. Dr. Martin Gogolla

Danksagung

Ich möchte mich ganz besonders herzlich bei Frau Dr. Sabine Kuske bedanken, die als erste Gutachterin mich bei der Erstellung dieser Bachelorarbeit fachlich unterstützt hat.

Mein besonderer Dank geht ebenfalls an Herr Prof. Dr. Martin Gogolla für seine Bereitwilligkeit, diese Arbeit zu begutachten.

Des Weiterem möchte ich meiner Frau dafür danken, dass sie mir in schwierigen Situationen immer beigestanden ist. Danke Jasmin Habenicht.

Zudem möchte ich mich bei meiner Mutter und meinem Vater für ihre moralische Unterstützung während meines Bachelorstudiums bedanken. Danke Jacqueline und Marc Nono.

Inhaltsverzeichnis

1	Einleitung	8
1.1	Motivation und Ziel der Arbeit	8
1.2	Aufbau der Arbeit	8
2	Graphen	10
2.1	Ungerichtete und gerichtete Graphen	10
2.2	Gewichtete Graphen	11
2.3	Wege et Kreise	12
2.3.1	Weg	12
2.3.2	Einfacher Weg	12
2.3.3	Kreis	12
2.3.4	Einfacher Kreis	12
2.3.5	Hamiltonscher Weg	12
2.3.6	Hamiltonscher Kreis	13
2.3.7	Hamiltonscher Graph	13
2.3.8	Eulerscher Weg	13
2.3.9	Eulerscher Kreis	13
2.3.10	Eulerscher Graph	13
2.3.11	Distanz	14
3	Entscheidungsprobleme	15
3.1	Das Hamiltonsche Wege-Problem für gerichtete Graphen	15
3.2	Das TSP für gerichtete Graphen	16
3.3	Die Klasse NP	17
3.4	Reduktion	18
3.5	NP-Vollständigkeit	18
4	Stellen/Transitions-Systeme	20
4.1	Netz	20
4.1.1	Vorbereich und Nachbereich	21
4.1.2	Markierung	21
4.2	Stellen/Transitions-Systeme	22
4.2.1	Aktivierte Transition	23
4.2.2	Schalten einer Transition	23
4.2.3	Schaltfolge	23
4.3	Erreichbarkeit	25
4.3.1	Erreichbare Markierung	25
4.3.2	Erreichbarkeitsgraph	26
4.4	Erreichbarkeitsproblem	27

4.5	Teilerreichbarkeitsproblem	28
4.6	Beschränkte Stellen/Transitions-Systeme	28
4.6.1	Beschränktheit	28
4.6.2	Beispiel	29
4.7	Deadlockfreies S/T-System	29
4.7.1	Deadlockfreiheit	29
4.7.2	Beispiel	29
5	Reduktion des Hamiltonschen-Wege-Problems auf das Erreichbarkeits-	
	problem	31
5.1	Konstruktion	31
5.2	Beispiel	32
5.3	Korrektheit	35
6	Reduktion des Rundreiseproblems auf das Erreichbarkeitsproblem	41
6.1	Konstruktion	41
6.2	Korrektheit	42
6.3	Beispiel	44
7	Zusammenfassung und Ausblick	47
	Literaturverzeichnis	48
	Eidesstattliche Erklärung	49

Abbildungsverzeichnis

1	Ungerichteter Graph(G')	11
2	Gerichteter Graph(G)	11
3	Ungerichteter Graph(G') mit Schlingen	11
4	Gewichteter Graph	12
5	Eulerscher Graph(G')	13
6	Graph mit Distanzen	14
7	Ein Netz	21
8	Eine Markierung M	22
9	Beispiel eines Stellen/Transitions-Systems	23
10	Markierung M_1 nach dem Schalten von t_0	24
11	Markierung M_2 nach dem Schalten von t_1	24
12	Markierung M_3 nach dem Schalten von t_2	25
13	Erreichbarkeitsgraph der Abbildung 9	26
14	Ein unbeschränktes S/T-System	27
15	Unendlicher Erreichbarkeitsgraph für das S/T-System in Abbildung 14	27
16	Ein beschränktes S/T-System	29
17	Ein deadlockfreies S/T-System	30
18	Von gerichteten Kanten zu Transitionen	31
19	Erweiterung um Stelle s_{count}	31
20	Graph G	32
21	S/T-System	33
22	Erreichbarkeitsgraph	33
23	S/T-System ($\mathcal{PN}(G)$) nach dem Schalten von $begin(v_0)$	34
24	S/T-System ($\mathcal{PN}(G)$) nach dem Schalten von $t((v_0, v_1))$	34
25	S/T-System ($\mathcal{PN}(G)$) nach dem Schalten von $t((v_1, v_2))$	35
26	Übersetzung einer gewichteten Kante	41
27	Graph G	45
28	S/T-System	45
29	S/T-System	46

Tabellenverzeichnis

1	Visualisierung der Komponenten eines Netzes	20
2	Markierungstabelle für das S/T-System in Abbildung 9	25

Abkürzungsverzeichnis

<i>STS, S/T-System</i>	Stellen/Transitions-System
<i>HAMPATH</i>	Hamiltonscher Weg
<i>TSP</i>	Traveling Salesman Problem
<i>TSPPATH</i>	Traveling Salesman Problem für Wege
<i>TEP</i>	Teilerreichbarkeitsproblem
<i>EP</i>	Erreichbarkeitsproblem

1 Einleitung

1.1 Motivation und Ziel der Arbeit

Graphen dienen in der Informatik als Darstellungsmittel zur Beschreibung und Veranschaulichung komplexer Sachverhalten ([Ill09]) und Zusammenhänge. Außerdem können sie eingesetzt werden, um kompliziert strukturierte Datenobjekte zu bearbeiten und zu untersuchen. Durch sie lassen sich Beziehungen oder Abhängigkeiten zwischen Einzelheiten ausdrücken (vgl. [Kus17]). Heutzutage finden Graphen ihre Anwendung zum Beispiel im Internet, in Straßennetzen und Rechnernetzen.

Petri-Netze sind Werkzeuge zur Modellierung, Analyse und Simulation von nichtdeterministischen, nebenläufigen, diskreten, verteilten und dynamischen Systemen. Sie wurden von dem deutschen Mathematiker und Informatiker Carl Adam Petri erfunden, der sie zum ersten Mal in seiner Dissertation „Kommunikation mit Automaten“ (1962) einsetzte. Sie finden ihre Anwendung zum Beispiel in der Modellierung von Geschäfts-, Produktions- und Betriebssystemprozessen.

Die am meisten behandelten und wichtigsten Probleme in der Graphentheorie sind Entscheidungsprobleme. Sie lassen sich allgemein durch den Beweis ihre Entscheidbarkeit lösen.

Bei Petri-Netzen beantwortet das Erreichbarkeitsproblem grundsätzlich die Frage, ob eine gegebene Markierung von der Startmarkierung aus erreichbar ist. Dieses Problem ist entscheidbar ([PW08]). Es stellt sich die Frage, ob Petri-Netz-Werkzeuge zum Beispiel zur Lösung von schwierigen Graphproblemen eingesetzt werden können.

In dieser Arbeit wird der Frage nachgegangen, ob und inwiefern sich Entscheidungsprobleme für Graphen auf das Erreichbarkeitsproblem für Petri-Netze reduzieren lassen.

1.2 Aufbau der Arbeit

Die vorliegende Arbeit gliedert sich in sieben Kapitel: Zunächst ist das vorliegende Kapitel der Einleitung gewidmet. Anschließend werden im zweiten Kapitel grundlegende Begriffe aus der Graphentheorie eingeführt. Darauf aufbauend wird im dritten Kapitel erläutert, was Entscheidungsprobleme sind. Zusätzlich werden einige Entscheidungsprobleme vorgestellt. Danach werden die Begriffe Komplexitätsklasse NP , Reduktion und NP -Vollständigkeit exponiert. Daraufhin steht im Fokus des vierten Kapitels die Einführung von Stellen/Transitionssystemen. Anschließend widmet sich das fünfte Kapitel der Reduktion des Hamiltonschen-Wege-Problems auf das Erreichbarkeitsproblem

für Stellen/Transitionssysteme. Das sechste Kapitel beschäftigt sich mit der Reduktion des Rundreiseproblems auf das Erreichbarkeitsproblem für Stellen/Transitionssysteme. Eine Zusammenfassung meiner Ergebnisse und ein kurzer Ausblick auf mögliche Erweiterungen schließen die Arbeit ab.

2 Graphen

Ein Graph ist ein Paar, bestehend aus einer endlichen nichtleeren Knotenmenge und einer Kantenmenge. Jedem Element der Kantenmenge ist eine Teilmenge der Knotenmenge zugeordnet, bestehend aus eins oder zwei Elemente. Die Elemente der Knotenmenge heißen Knoten oder Ecken ([Nit04]) und die der Kantenmenge Kanten. Die Knoten sind je nach Graphen-Art durch gerichtete oder ungerichtete Kanten miteinander verbunden.

Manche Graphen haben bestimmte Eigenschaften wie die von dem irischen Mathematiker und Physiker William Rowan Hamilton (4. August 1805 – 2. September 1865) erfundenen Hamiltonschen Graphen, die im Laufe dieser Arbeit oft zum Einsatz kommen werden.

In diesem Kapitel werden grundlegende Definitionen und Begriffe aus der Graphentheorie mit einigen Beispielen vorgestellt. Die Definitionen aus (2.1) orientieren sich an denen von gerichteten und ungerichteten Graphen aus dem Buch von Beierle und Kern-Isberner ([BKI14]). Weitere Definitionen wie Hamiltonscher Weg, Hamiltonscher Kreis und Hamiltonscher Graph orientieren sich an dem Buch von Krumke und Noltemeier ([KN09]). Darüber hinaus orientieren sich die Definitionen für den Eulerschen Weg, Eulerschen Kreis und Eulerschen Graph an dem Buch von Uwe Kastens und Hans Kleine Büning ([KB18]) und dem Buch von Krumke und Noltemeier ([KN09]).

2.1 Ungeriectete und gerichtete Graphen

Ein Graph ist ein Paar $G = (V, E)$. V ist die Knotenmenge von G und E die Kantenmenge von G mit $E \subseteq V \times V$. G heißt ungerichteter Graph, wenn $\forall v_0, v_1 \in V, (v_0, v_1) \in E \implies (v_1, v_0) \in E$ (symmetrische Eigenschaft von E), sonst heißt G gerichteter Graph. Bei einem ungerichteten Graphen gibt es für jede Kante von a nach b auch eine Kante von b nach a . Aus diesem Grund werden die beiden Kanten in graphischen Darstellungen zu einer Kante zusammengefasst und die Pfeilspitzen weggelassen. Abbildung 1 zeigt ein Beispiel eines ungerichteten Graphen; in Abbildung 2 ist ein gerichteter Graph zu sehen.

Eine Schlinge ist eine Kante der Form (v, v) , wobei ein Knoten mit sich selbst verbunden wird. Folglich gilt $E \subseteq \{(v, v') \in V \times V \mid v \neq v'\}$. Abbildung 3 auf der folgenden Seite stellt Schlingen in einem ungerichteten Graphen dar. Im weiteren Verlauf dieser Arbeit werden gerichtete Graphen ohne Schlingen betrachten.

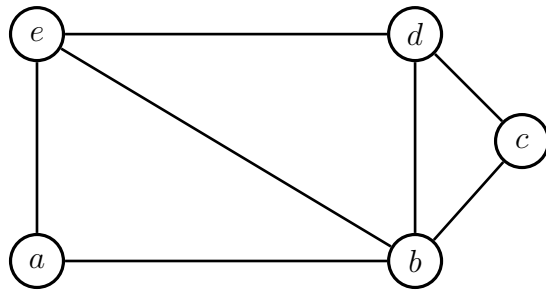


Abbildung 1: Ungerichteter Graph(G')

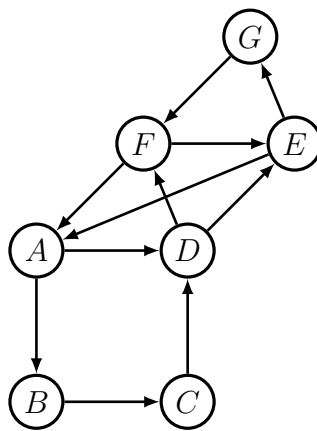


Abbildung 2: Gerichteter Graph(G)

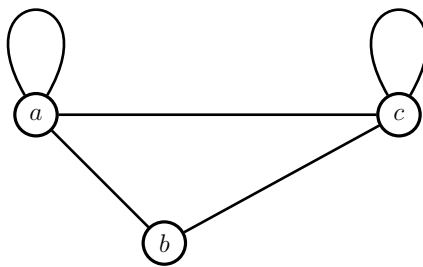


Abbildung 3: Ungerichteter Graph(G') mit Schlingen

2.2 Gewichtete Graphen

Ein Graph heißt gewichtet, falls seine Kanten mit natürlichen Zahlen als Gewicht versehen sind. Abbildung 4 auf der folgenden Seite stellt einen gewichteten Graphen dar.

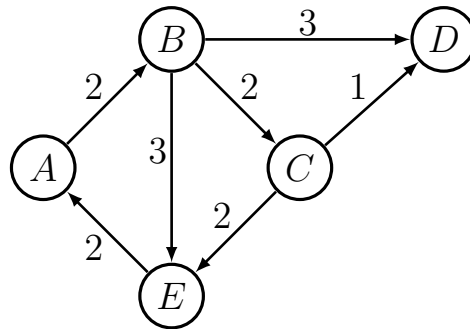


Abbildung 4: Gewichteter Graph

2.3 Wege et Kreise

2.3.1 Weg

Ein Weg in G von v nach v' der Länge n ist eine Folge $v_0 \dots v_n$ mit $v_0 = v, v_n = v'$ und $(v_{i-1}, v_i) \in E$ für $i = 1, \dots, n$. In Abbildung 2 stellt $p = ABCDEAB$ einen Weg dar, der die Kante (A, B) zwei Mal besucht.

2.3.2 Einfacher Weg

Ein Weg in G von v nach v' heißt einfach, wenn jeder seiner Knoten nur einmal besucht wird. In Abbildung 2 stellt $p = ABCDEG$ einen einfachen Weg im Graphen G dar.

2.3.3 Kreis

Ein Kreis ist ein Weg im Graphen G von v nach v der Länge $n \geq 2$ mit gleichem Anfangs- und Endknoten. In Abbildung 2 stellt $p = GFADFEG$ einen Kreis im Graphen G dar.

2.3.4 Einfacher Kreis

Ein Kreis v_0, \dots, v_n ist ein einfacher Kreis, falls v_0, \dots, v_{n-1} ein einfacher Weg ist. In Abbildung 2 stellt $p = ABCDEA$ einen einfachen Kreis im Graphen G dar.

2.3.5 Hamiltonscher Weg

Ein Hamiltonscher Weg im Graphen $G = (V, E)$ ist ein einfacher Weg in G der Länge

$|V| - 1$, das heißt, dass jeder Knoten aus V genau einmal besucht wird. In Abbildung 2 stellt $p = ABCDFEG$ einen Hamiltonschen Weg im Graphen G dar.

2.3.6 Hamiltonscher Kreis

Ein Hamiltonscher Kreis im Graphen $G = (V, E)$ ist ein einfacher Kreis der Länge $|V|$. In Abbildung 2 stellt $p = ABCDEGFA$ einen Hamiltonschen Kreis im Graphen G dar.

2.3.7 Hamiltonscher Graph

Ein Graph $G = (V, E)$ wird Hamiltonsch genannt, falls er einen Hamiltonschen Kreis enthält. Der Graph G in Abbildung 2 ist ein Hamiltonscher Graph, da er einen Hamiltonschen Kreis (wie zum Beispiel $p = ABCDEGFA$) enthält.

2.3.8 Eulerscher Weg

Ein Eulerscher Weg im Graphen $G = (V, E)$ ist ein Weg, der jede Kante aus E genau einmal durchläuft. In Abbildung 1 stellt $p = eabcdbed$ einen Eulerschen Weg in dem ungerichteten Graphen G' dar.

2.3.9 Eulerscher Kreis

Ein Eulerscher Kreis im Graphen $G = (V, E)$ ist ein Kreis, der jede Kante aus E genau einmal durchläuft.

2.3.10 Eulerscher Graph

Ein Graph $G = (V, E)$ heißt Eulersch, falls er einen Eulerschen Kreis enthält.

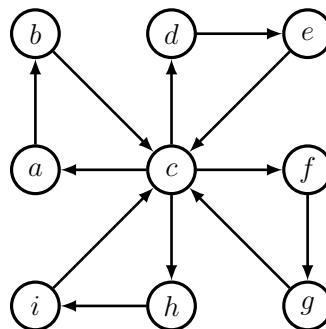


Abbildung 5: Eulerscher Graph(G')

Abbildung 5 auf der vorherigen Seite stellt einen Eulerschen Graphen dar, denn sie enthält den Eulerschen Kreis $p = abcdecfgchica$.

2.3.11 Distanz

Sei $p = v_0 \dots v_n$ ein Weg, dann ist $dist(p) = \sum_{i=1}^n dist((v_{i-1}, v_i))$ die Distanz dieses Weges, das heißt die Summe der Distanzen aller seiner besuchten Kanten.

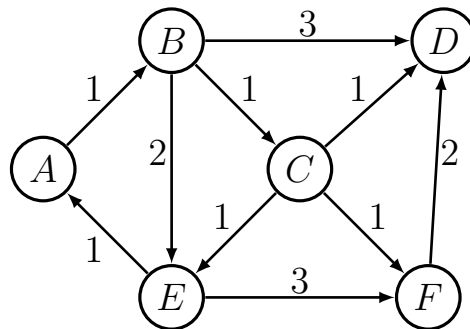


Abbildung 6: Graph mit Distanzen

In Abbildung 6 ist der Weg $p = ABCEFD$ erkennbar. Die Distanz dieses Weges beträgt:

$$\begin{aligned}
 dist(p) &= dist(A, B) + dist(B, C) + dist(C, E) + dist(E, F) + dist(F, D) \\
 &= 1 + 1 + 1 + 3 + 2 \\
 &= 8
 \end{aligned}$$

3 Entscheidungsprobleme

Die wichtigsten und bedeutungsvollen Probleme der mathematischen Logik sind Entscheidungsprobleme, die 1928 von dem deutschen Mathematiker David Hilbert eingeführt wurden (vgl.[BGG10]). Entscheidungsprobleme lassen sich in Komplexitätsklassen aufteilen. In diesem Kapitel werden sowohl die Definition für das Entscheidungsproblem als auch die in dieser Arbeit betrachteten Beispiele vorgestellt. Außerdem werden die Definitionen für Reduktion und Polynomialzeitreduktion eingeführt, die dazu dienen, die Entscheidbarkeit bzw. Unentscheidbarkeit von Problemen zu zeigen ([Wim08]). Da sich Entscheidungsprobleme in der Informatik meist in Komplexitätsklassen aufteilen lassen, wird eine davon vorgestellt, und zwar die Klasse NP .

Die Aussagen aus dem folgenden Abschnitt über Entscheidungsprobleme orientieren sich an dem Buch von Alexander Asteroth und Christel Baier ([AB02]). Die Definitionen für die Klasse NP (siehe Abschnitt 3.3) und Reduktion (siehe Abschnitt 3.4) orientieren sich an dem Vorlesungsskript über Algorithmen auf Graphen von Dr. Sabine Kuske ([Kus17]). Die Definition für die NP -Vollständigkeit eines Problems orientiert sich an dem Buch von Weicker ([WW13]). Außerdem orientiert sich die Definition des Traveling-Salesman-Problems für gerichtete Graphen an dem Buch von Volker Turau ([Tur09]).

Entscheidungsprobleme sind Probleme der Form $D : IN \longrightarrow \{Ja, Nein\}$, die jeweils einen Eingabebereich IN besitzen und für jede Eingabe entweder Ja oder $Nein$ liefern. Für solche Probleme muss es einen Algorithmus geben, der diese löst, sonst heißen sie unentscheidbar. Das Halteproblem ist eines der bekanntesten unentscheidbaren Probleme, die es gibt, wobei es darum geht, mit Hilfe eines Algorithmus herauszufinden, ob ein gegebenes Programm nach endlich vielen Schritten terminiert oder nicht. Entscheidbare Probleme sind zum Beispiel das Hamiltonsche-Kreis-Problem und das Traveling-Salesman-Problem. Im Laufe dieser Arbeit werden diese beiden Probleme für Wege (siehe Abschnitt 3.1 und 3.2) statt Kreise in gerichteten Graphen behandelt. Sowohl die Entscheidbarkeit als auch die Unentscheidbarkeit von Problemen kann mit Hilfe von Reduktion bewiesen werden. Erfolgt eine Reduktion in polynomieller Zeit, heißt sie Polynomialzeitreduktion. Mit Polynomialzeitreduktion kann zum Beispiel die NP -Härte eines Entscheidungsproblems bewiesen werden.

3.1 Das Hamiltonsche Wege-Problem für gerichtete Graphen

Das Hamiltonsche Wege-Problem stellt die Frage, ob es in einem gerichteten Graphen $G = (V, E)$ einen einfachen Weg der Länge $|V| - 1$ gibt. Es kann wie folgt formuliert

werden:

HAMPATH

Eingabe: $G = (V, E)$

Ausgabe: $\begin{cases} Ja & \text{falls } G \text{ einen Hamiltonschen Weg enthält} \\ Nein & \text{sonst} \end{cases}$

Beobachtung 1:

HAMPATH ist entscheidbar.

Beweis.

Um zu zeigen, dass *HAMPATH* entscheidbar ist, reicht es einen (nichtdeterministischen) Algorithmus anzugeben, der das Problem löst.

Algorithmus:

1. Rate eine Permutation π der Knotenmenge V .
2. Teste deterministisch in Polynomialzeit, ob π ein Hamiltonscher Weg in G ist.
 - Falls ja, gibt *Ja* aus (und stoppe).
 - Falls nein, gibt *Nein* aus (und stoppe).

Der oben vorgestellte Algorithmus löst *HAMPATH*.

HAMPATH ist also entscheidbar.

■

3.2 Das TSP für gerichtete Graphen

Das Traveling-Salesman-Problem stellt die Frage, ob es in einem gerichteten Graphen G einen Hamiltonschen Kreis gibt, dessen Distanz höchstens k (vorgegebene natürliche Zahl $k \in \mathbb{N}$) ist. Es kann wie folgt formuliert werden:

TSP

Eingabe: $G = (V, E)$, $dist : E \rightarrow \mathbb{N}$, $k \in \mathbb{N}$.

Ausgabe: $\begin{cases} Ja & \text{falls } \exists \text{ ein Hamiltonscher Kreis } p \text{ in } G \text{ mit } dist(p) \leq k \\ Nein & \text{sonst} . \end{cases}$

Beobachtung 2:

TSP ist entscheidbar.

Der Beweis geht analog zu dem Beweis der Entscheidbarkeit von *HAMPATH* aus dem Abschnitt 3.1 und wird deswegen hier nicht ausgeführt.

Bei dem Entscheidungsproblem *TSPPATH* stellt sich die Frage, ob es in einem gerichteten Graphen G einen Hamiltonschen Weg gibt, dessen Distanz höchstens k (vorgegebene natürliche Zahl $k \in \mathbb{N}$) ist. Es kann wie folgt formuliert werden:

TSPPATH

Eingabe: $G = (V, E)$, $dist : E \rightarrow \mathbb{N}$, $k \in \mathbb{N}$.

Ausgabe: $\begin{cases} Ja & \text{falls } \exists \text{ ein Hamiltonscher Weg } p \text{ in } G \text{ mit } dist(p) \leq k \\ Nein & \text{sonst.} \end{cases}$

Beobachtung 3:

TSPPATH ist entscheidbar.

Der Beweis geht analog zu dem Beweis der Entscheidbarkeit von *HAMPATH* aus dem Abschnitt 3.1 und wird deswegen hier nicht ausgeführt.

3.3 Die Klasse NP

Ein Entscheidungsproblem liegt genau dann in der Klasse *NP*, wenn es sich durch einen nichtdeterministischen Algorithmus mit polynomiellm Zeitaufwand lösen lässt. Folgender Satz besagt, dass folgende Entscheidungsprobleme in *NP* liegen: *HAMPATH*, *TSP* und *TSPPATH*.

Satz 1:

$HAMPATH, TSP, TSPPATH \in NP$

Beweis.

Dass *HAMPATH* in *NP* liegt, wurde bereits im Beweis von Beobachtung 1 gezeigt. Für *TSP* und *TSPPATH* verläuft die Argumentation analog. ■

3.4 Reduktion

Seien D und D' zwei Entscheidungsprobleme mit den Eingabemengen IN bzw. IN' . Dann heißt D auf D' reduzierbar (Notation: $D \leq D'$), falls es eine berechenbare Übersetzungsfunktion $f : IN \rightarrow IN'$ gibt, so dass gilt:

$$D(x) = Ja \iff D'(f(x)) = Ja$$

Lässt sich f in Polynomialzeit berechnen, so handelt es sich um eine Polynomialzeitreduktion (Notation $D \leq_P D'$). Lässt sich ein Entscheidungsproblem D auf ein Entscheidungsproblem D' reduzieren und gibt es einen Algorithmus zum Lösen von D' , so lässt sich dieser Algorithmus zum Lösen von D benutzen.

Auf den Beweis des folgenden Satzes wird hier verzichtet. Der ist zum Beispiel in [HMU02] zu finden.

Satz 2:

Es seien D_1 und D_2 zwei Entscheidungsprobleme mit $D_1 \leq D_2$

- *ist D_2 entscheidbar, so ist auch D_1 entscheidbar.*
- *ist D_1 unentscheidbar, so ist auch D_2 unentscheidbar.*

In späteren Kapiteln wird gezeigt, dass sich $\mathcal{HAMPATH}$ und $\mathcal{TSPPATH}$ auf ein Entscheidungsproblem für Petri-Netze reduzieren lässt. So könnte zum Beispiel untersucht werden, inwieweit sich Petri-Netz-Werkzeuge zur Lösung dieser Probleme einsetzen lassen.

3.5 NP-Vollständigkeit

NP -vollständige Probleme sind die schwersten Problemen in der Klasse NP und wurden 1971 von dem Mathematiker und Informatiker Stephen Cook in seiner wissenschaftlichen Publikation mit dem Titel „The Complexity of Theorem Proving Procedures“ eingeführt [GJ79].

Ein Entscheidungsproblem D ist NP -vollständig, falls:

- $D \in NP$
- D ist NP -hart, das heißt, jedes Problem D' in NP lässt sich in Polynomialzeit auf D reduzieren ($\forall D' \in NP : D' \leq_P D$).

Folgender Satz besagt, dass, wenn sich das Entscheidungsproblem D_1 in Polynomialzeit auf das Entscheidungsproblem D_2 reduzieren lässt und D_1 NP -hart ist, dann ist D_2 auch NP -hart. Auf den Beweis wird hier verzichtet, weil ein analoger Beweis mit der NP -Vollständigkeit in [GJ79, S.38] zu finden ist.

Satz 3:

Es seien D_1 und D_2 zwei Entscheidungsprobleme.

$D_1 \leq_p D_2$ und D_1 ist NP -hart $\implies D_2$ ist NP -hart.

Beobachtung 4:

$HAMPATH$, TSP , $TSPPATH$ sind NP -vollständig.

Die Beweise der NP -Vollständigkeit von $HAMPATH$, TSP und $TSPPATH$ sind zum Beispiel in [GJ79] zu finden, daher werden sie hier nicht mehr ausgeführt.

4 Stellen/Transitions-Systeme

In diesem Kapitel wird eine Petri-Netz-Art vorgestellt, und zwar Stellen /Transitions-Systeme. Außerdem werden die Begriffe Netz, Schaltfolge, Erreichbarkeit, Erreichbarkeitsproblem, Teilerreichbarkeitsproblem und beschränkte Stellen/Transitions-Systeme eingeführt. Die in diesem Kapitel vorgestellten Definitionen orientieren sich an dem Vorlesungsskript über Petri-Netze von Dr. Sabine Kuske ([Kus18]), das bei dem Teil über Netze (4.1) auf der Arbeit von Rozenberg und Engelfriet [RE98] basiert und bei dem Teil über Stellen/Transitions-Systeme (4.2) auf den Büchern von Priese und Wimmel [PW08] und Starke [Sta90] basiert. Die Teile über die Komplexität des Erreichbarkeitsproblems und das Teilerreichbarkeitsproblem orientieren sich an dem Buch von Priese und Wimmel [PW08].

4.1 Netz

Ein Netz besteht aus einer Menge von Stellen, einer Menge von Transitionen. Transitionen und Stellen sind durch eine Flussrelation miteinander verbunden. Keine Stelle darf gleichzeitig eine Transition sein. Formal ist ein Netz wie folgt definiert.

Ein Netz ist ein Triple $N = (S, T, F)$ mit:

- S ist eine endliche Menge von Stellen
- T ist eine endliche Menge von Transitionen mit $S \cap T = \emptyset$
- F ist eine Flussrelation mit $F \subseteq (S \times T) \cup (T \times S)$.
Die Elemente in F werden Kanten genannt.

Es ist außerdem Folgendes anzumerken:

- Stellen werden graphisch durch Kreise dargestellt.
- Transitionen werden graphisch durch Rechtecke dargestellt.
- Gerichtete Kanten verbinden Stellen und Transitionen miteinander.

Stelle	○
Transition	□
Kante	→

Tabelle 1: Visualisierung der Komponenten eines Netzes

Abbildung 7 zeigt das Netz $N = (S, T, F)$ mit :
 $S = \{s_0, s_1, s_2, s_3\}$,

$T = \{t_0, t_1, t_2\}$ und

$F = \{(s_0, t_0), (t_0, s_1), (s_1, t_1), (t_1, s_1), (t_1, s_3), (t_0, s_2), (s_2, t_2), (t_2, s_3)\}$.

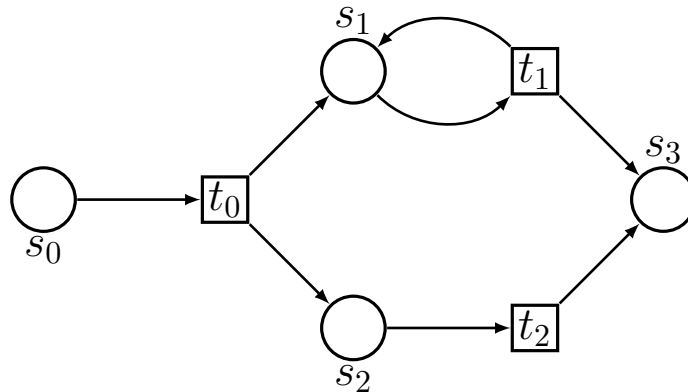


Abbildung 7: Ein Netz

4.1.1 Vorbereich und Nachbereich

Eine Stelle s heißt Vorstelle einer Transition t , falls es eine Kante von s nach t im Netz gibt. Falls es eine Kante von t nach s gibt, wird s als Nachstelle von t bezeichnet. Die Menge aller Vorstellen einer Transition ist ihr Vorbereich und die Menge aller Nachstellen ihr Nachbereich. In analoger Weise haben auch Stellen Vor- und Nachbereiche.

Sei $x \in S \cup T$.

$\bullet x = \{y \in S \cup T \mid (y, x) \in F\}$ wird Vorbereich von x genannt.

$x^\bullet = \{y \in S \cup T \mid (x, y) \in F\}$ wird Nachbereich von x genannt.

In Abbildung 7 sind $\bullet t_0 = \{s_0\}$ und $t_0^\bullet = \{s_1, s_2\}$ die Vor- und Nachstellen der Transition t_0 . $\bullet s_1 = \{t_0, t_1\}$ und $s_1^\bullet = \{t_1\}$ sind die Vor- und Nachtransitionen der Stelle s_1 .

4.1.2 Markierung

Sei $STS = (S, T, F, W, M_0)$ ein S/T-System. Eine Markierung von STS ist eine Abbildung $M : S \rightarrow \mathbb{N}$, die jeder Stelle eine natürliche Zahl zuordnet.

Die Tokens werden graphisch durch kleine gefüllte Kreise (\bullet) innerhalb einer Stelle dargestellt. Abbildung 8 zeigt das Netz aus Abbildung 7 mit der Markierung M , die den Stellen s_0, s_1, s_2 und s_3 die natürlichen Zahlen 2, 1, 0 und 0 zuordnet, das heißt $M(s_0) = 2, M(s_1) = 1, M(s_2) = 0$ und $M(s_3) = 0$.

Werden die Stellen in eine Reihenfolge gebracht, können Markierungen als Tupel notiert

werden. Zum Beispiel kann so die genannte Beispielmartierung M als $(M(s_0), M(s_1), M(s_2), M(s_3))$ geschrieben werden, das heißt $M = (2, 1, 0, 0)$.

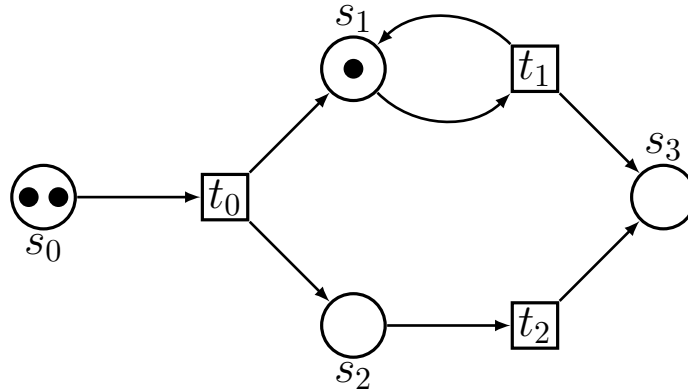


Abbildung 8: Eine Markierung M

4.2 Stellen/Transitions-Systeme

Stellen/Transitions-Systeme (kurz S/T-Systeme) bestehen aus einem Netz mit einer Anfangsmarkierung. Jede Kante ist zusätzlich mit einem Gewicht versehen, das neben die Kante geschrieben wird, wobei die Zahl 1 in graphischen Darstellungen weggelassen werden kann.

Formal ist ein S/T-System wie folgt definiert:

Ein 5-Tupel $STS = (S, T, F, W, M_0)$ heißt Stellen/Transitions-System, wenn gilt:

- (S, T, F) ist ein Netz.
- $W : F \rightarrow \mathbb{N}_{>0}$ ist eine Abbildung, die jeder Kante eine positive natürliche Zahl als Gewicht zuordnet.
- $M_0 : S \rightarrow \mathbb{N}$ ist eine Abbildung, die jeder Stelle eine natürliche Zahl zuordnet. M_0 wird Anfangsmarkierung genannt.

Abbildung 9 stellt das Stellen/Transitions-System (S, T, F, W, M_0) dar mit:

- $S = \{s_0, s_1, s_2, s_3\}$,
 $T = \{t_0, t_1, t_2\}$ und
 $F = \{(s_0, t_0), (t_0, s_1), (s_1, t_1), (t_1, s_1), (t_1, s_3), (t_0, s_2), (s_2, t_2), (t_2, s_3)\}$.
- $W(s_0, t_0) = 1$, $W(t_0, s_1) = 1$, $W(s_1, t_1) = 2$, $W(t_1, s_1) = 1$, $W(t_1, s_3) = 1$,
 $W(t_0, s_2) = 1$, $W(s_2, t_2) = 2$, $W(t_2, s_3) = 1$.
- $M_0(s_0) = 2$, $M_0(s_1) = 1$, $M_0(s_2) = 0$ und $M_0(s_3) = 0$.

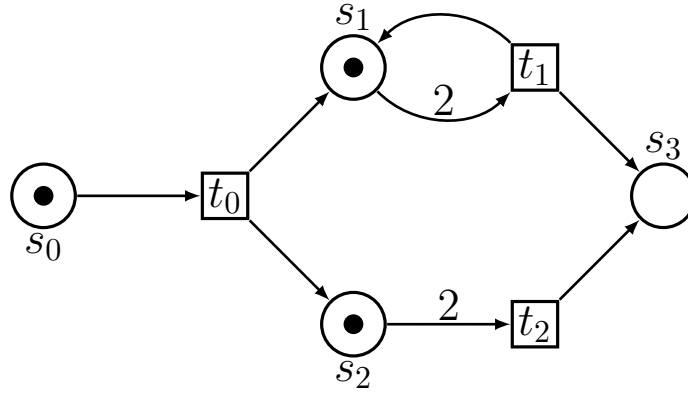


Abbildung 9: Beispiel eines Stellen/Transitions-Systems

4.2.1 Aktivierte Transition

Sei $STS = (S, T, F, W, M_0)$ ein S/T-System und sei $M : S \rightarrow \mathbb{N}$ eine Markierung von STS . Eine Transition $t \in T$ heißt M -aktiviert, wenn $\forall s \in \bullet t$ gilt: $M(s) \geq W(s, t)$.

Die Tokenzahl jeder Vorstelle s von t muss also größer gleich dem Gewicht der Kante sein, welche die Stelle s und die Transition t miteinander verbindet.

Die Transition t_0 in Abbildung 9 ist unter der Anfangsmarkierung M_0 aktiviert, da $M(s_0) = 1 \geq W(s_0, t_0) = 1$.

4.2.2 Schalten einer Transition

Sei $STS = (S, T, F, W, M_0)$ ein S/T-System und seien $M, M' : S \rightarrow \mathbb{N}$ Markierungen von STS . Eine M -aktivierte Transition $t \in T$ schaltet von M nach M' (Notation $M[t]M'$), wenn gilt:

$$M'(s) = \begin{cases} M(s) - W(s, t) & \text{für } s \in \bullet t - t \bullet \\ M(s) + W(t, s) & \text{für } s \in t \bullet - \bullet t \\ M(s) - W(s, t) + W(t, s) & \text{für } s \in \bullet t \cap t \bullet \\ M(s) & \text{sonst.} \end{cases}$$

M' heißt Folgemarkierung von M .

4.2.3 Schaltfolge

Sei $STS = (S, T, F, W, M_0)$ ein S/T-System. Es gilt:

- (1) Das leere Wort λ ist eine Schaltfolge von M nach M (Notation $M[\lambda]M$) für jede Markierung $M : S \rightarrow \mathbb{N}$.

- (2) Seien $M, M', M'' : S \rightarrow \mathbb{N}$ Markierungen von STS , $w \in T^*$ eine Schaltfolge von M nach M' , das heißt $M[w]M'$, und sei $t \in T$ mit $M'[t]M''$. Dann ist wt eine Schaltfolge von M nach M'' , das heißt $M[wt]M''$.

Eine mögliche Schaltfolge des Systems in Abbildung 9 ist $t_0t_1t_2$. Nach dem Schalten der Transition t_0 entsteht die in Abbildung 10 dargestellte Markierung M_1 .

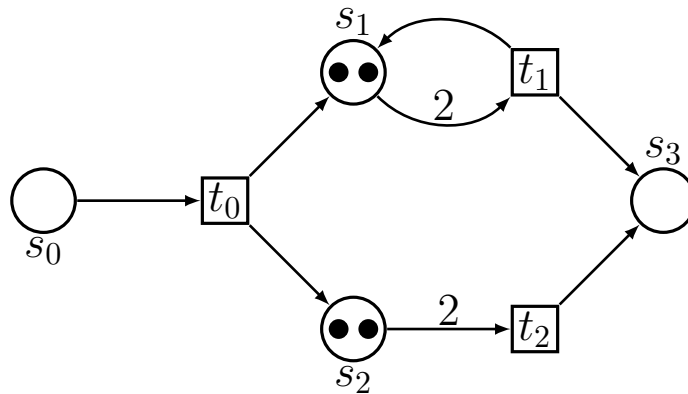


Abbildung 10: Markierung M_1 nach dem Schalten von t_0

Nachdem t_0 geschaltet hat, kann sie nicht mehr schalten, da $M_1(s_0) < W(s_0, t_0)$. Nur t_1 oder t_2 kann geschaltet werden.

Nach dem Schalten der Transition t_1 entsteht die Markierung M_2 aus Abbildung 11.

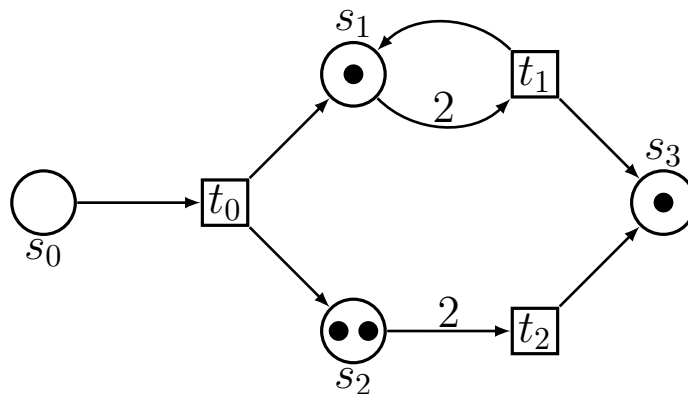


Abbildung 11: Markierung M_2 nach dem Schalten von t_1

Nachdem t_1 geschaltet hat, kann sie nicht mehr schalten, da $M_2(s_1) < W(s_1, t_1)$. Nur t_2 kann zum Schluss geschaltet werden. Beim Schalten der Transition t_2 werden zwei Tokens aus s_2 herausgenommen und eins in s_3 reingesteckt. s_3 besitzt jetzt zwei Tokens. Es entsteht die Markierung M_3 aus Abbildung 12.

Bei M_3 kann keine Transition mehr geschaltet werden. Das S/T-System befindet sich somit in einem Deadlock. Weiter unten (Abschnitt 4.7) werden die Begriffe Deadlock und Deadlockfreiheit erläutern.

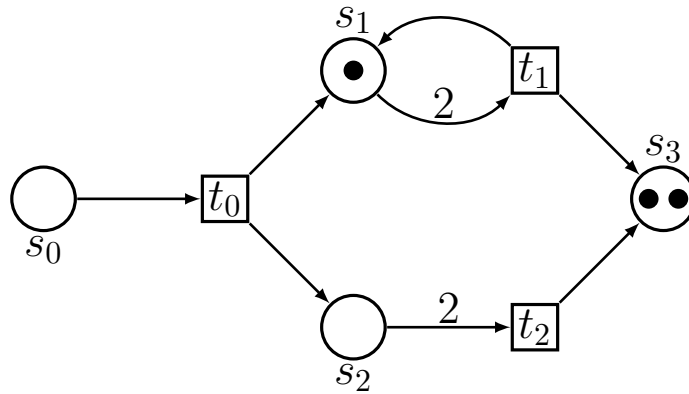


Abbildung 12: Markierung M_3 nach dem Schalten von t_2

4.3 Erreichbarkeit

Bestimmte Markierungen eines S/T-Systems können von anderen aus mit einer Schaltfolge erreichbar sein. Die erreichbaren Markierungen eines S/T-Systems sind alle Markierungen die von der Startmarkierung M_0 erreichbar sind.

4.3.1 Erreichbare Markierung

Sei $STS = (S, T, F, W, M_0)$ ein S/T-System und seien $M, M' : S \rightarrow \mathbb{N}$ Markierungen von STS . Dann heißt M' erreichbar von M (Notation $M[*]M'$), falls es eine Schaltfolge $w \in T^*$ gibt, so dass $M[w]M'$. Die Menge der von M erreichbaren Markierungen ist definiert als $Reach(M) = \{M' : S \rightarrow \mathbb{N} \mid M[*]M'\}$. Die Menge der in STS erreichbaren Markierungen ist $Reach(STS) = Reach(M_0)$.

Laut Tabelle 2 ist die Markierung M_3 von der Anfangsmarkierung M_0 durch die Schaltfolge $t_0t_1t_2$ erreichbar, das heißt $M_0[t_0t_1t_2]M_3$ ($M_0[t_0]M_1[t_1]M_2[t_2]M_3$ mit $M_0 = (1, 1, 1, 0)$, $M_1 = (0, 2, 2, 0)$, $M_2 = (0, 1, 2, 1)$ und $M_3 = (0, 1, 0, 2)$).

Markierungen	Stellen				Mögliche Schaltungen
	s_0	s_1	s_2	s_3	
M_0	1	1	1	0	$t_0 \rightarrow M_1$
M_1	0	2	2	0	$t_1 \rightarrow M_2$ $t_2 \rightarrow M_4$
M_2	0	1	2	1	$t_2 \rightarrow M_3$
M_4	0	2	0	1	$t_1 \rightarrow M_3$
M_3	0	1	0	2	

Tabelle 2: Markierungstabelle für das S/T-System in Abbildung 9

Der Erreichbarkeitsgraph stellt alle in einem S/T-System von M_0 erreichbaren Markierungen graphisch dar.

4.3.2 Erreichbarkeitsgraph

Sei $STS = (S, T, F, W, M_0)$ ein S/T-System. Der Erreichbarkeitsgraph von STS ist der Graph $G(STS) = (V, E)$ mit der Knotenmenge $V = Reach(STS)$ und der Kantenmenge $E = \{(M, t, M') \in V \times T \times V \mid M[t]M'\}$. Hier handelt es sich um einen so genannten kantenmarkierten Graphen, da jede Kante als Triple und nicht als Paar wie in Kapitel 2 definiert ist.

Abbildung 13 veranschaulicht den Erreichbarkeitsgraph des S/T-Systems aus Abbildung 9. Es ist leicht zu sehen, dass ein enger Zusammenhang zwischen dem Erreichbarkeitsgraph und Tabelle 2 besteht.

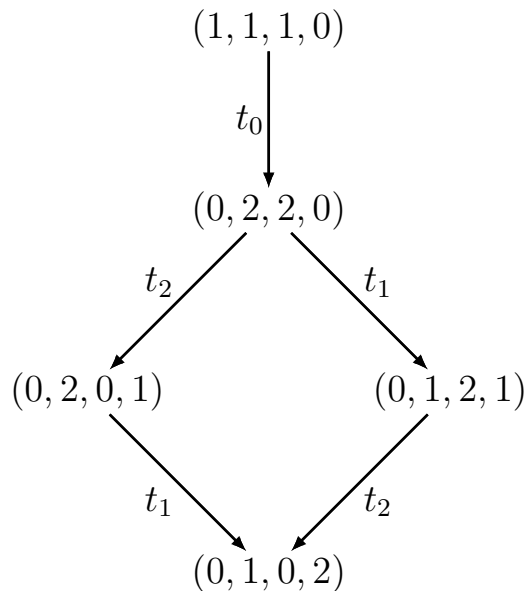


Abbildung 13: Erreichbarkeitsgraph der Abbildung 9

Eigenschaften von Erreichbarkeitsgraphen

- (1) $G(STS)$ ist zusammenhängend.
- (2) Zu jedem Knoten $v \in V$ führt ein einfacher Weg, der bei M_0 beginnt.
- (3) Jeder Knoten von $G(STS)$ besitzt höchstens $|T|$ hineingehende Kanten und höchstens $|T|$ herausgehende Kanten.
- (4) $G(STS)$ kann unendlich sein.

Abbildung 14 zeigt ein S/T-System, dessen Erreichbarkeitsgraph (Abbildung 15) unendlich ist. Die Transition t wird unendlich oft immer wieder schalten können. Nach jedem Schalten wird die Stelle s_0 immer einen Token beinhalten, wobei die Anzahl der Tokens in den Stellen s_1 und s_2 sich immer um eins erhöhen wird.

Ein S/T-System mit unendlichem Erreichbarkeitsgraph heißt unbeschränkt.

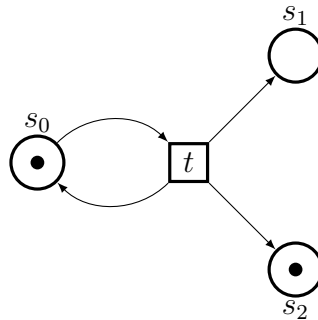


Abbildung 14: Ein unbeschränktes S/T-System

$$(1, 0, 1) \xrightarrow{t} (1, 1, 2) \xrightarrow{t} (1, 2, 3) \xrightarrow{t} (1, 3, 4) \xrightarrow{t} \dots$$

Abbildung 15: Unendlicher Erreichbarkeitsgraph für das S/T-System in Abbildung 14

4.4 Erreichbarkeitsproblem

Das Erreichbarkeitsproblem beantwortet bei Petri-Netzen grundsätzlich die Frage, ob eine gegebene Markierung von der Startmarkierung aus erreichbar ist.

Das Erreichbarkeitsproblem (kurz \mathcal{EP}) ist ein Problem, das wegen seiner EXPSPACE-harten Platzkomplexität schwierig zu lösen ist, weil seine Lösung im schlechtesten Fall exponentiellen Speicherplatz benötigt. Im Jahr 1980 behandelte der Mathematiker und Informatiker Ernst Wilhelm Mayr dieses Problem in seiner Doktorarbeit mit dem Titel „Ein Algorithmus für das allgemeine Erreichbarkeitsproblem bei Petrinetzen und damit zusammenhängende Fragen“. Ihm ist die Lösung dieses Problems zu verdanken, die zu einem der fundamentalen Probleme in der theoretischen Informatik geworden ist (Vgl. [dW10], [May80]).

Das Erreichbarkeitsproblem kann wie folgt formuliert werden:

\mathcal{EP}

Eingabe: Ein S/T-System $STS = (S, T, F, W, M_0)$ und eine Markierung $M : S \rightarrow \mathbb{N}$

Ausgabe: $\begin{cases} Ja & \text{falls } M \in Reach(M_0) \\ Nein & \text{sonst.} \end{cases}$

4.5 Teilerreichbarkeitsproblem

Das Teilerreichbarkeitsproblem bekommt als Eingabe ein S/T-System $STS = (S, T, F, W, M_0)$ und eine Markierung einer Teilmenge von der Menge aller Stellen. Es wird *Ja* ausgegeben, falls es eine erreichbare Markierung gibt, die für jede Stelle der Teilmenge die gewünschte Tokenzahl aufweist. Ansonsten wird *Nein* als Antwort ausgegeben. Es ist anzumerken, dass das Erreichbarkeitsproblem ein Spezialfall des Teilerreichbarkeitsproblems ist. Formal ist das Teilerreichbarkeitsproblem wie folgt definiert:

\mathcal{TEP}

Eingabe: Ein S/T-System $STS = (S, T, F, W, M_0)$, eine Markierung $\hat{M} : \hat{S} \rightarrow \mathbb{N}$ mit $\hat{S} \subseteq S$.

Ausgabe:
$$\begin{cases} Ja & \text{falls } \exists M \in Reach(STS) : M(\hat{s}) = \hat{M}(\hat{s}) \ \forall \hat{s} \in \hat{S} \\ Nein & \text{sonst .} \end{cases}$$

Satz 4:

\mathcal{TEP} ist entscheidbar.

Der Beweis ist in [PW08] zu finden.

4.6 Beschränkte Stellen/Transitions-Systeme

Hat eine Stelle in allen erreichbaren Markierungen höchstens k Token, ist sie k -beschränkt. Andernfalls ist die Stelle unbeschränkt. Ist jede Stelle eines S/T-Systems k -beschränkt, so ist das System ebenfalls k -beschränkt.

4.6.1 Beschränktheit

Sei $STS = (S, T, F, W, M_0)$ ein S/T-System.

- Für ein $k \in \mathbb{N}$ heißt eine Stelle $s \in S$ k -beschränkt, falls $M(s) \leq k$ für alle $M \in Reach(M_0)$.
- Eine Stelle s heißt beschränkt, falls es ein $k \in \mathbb{N}$ gibt, so dass s k -beschränkt ist.
- STS heißt k -beschränkt, falls jede Stelle in STS k -beschränkt ist.

- *STS* heißt beschränkt, falls jede Stelle in *STS* beschränkt ist.

4.6.2 Beispiel

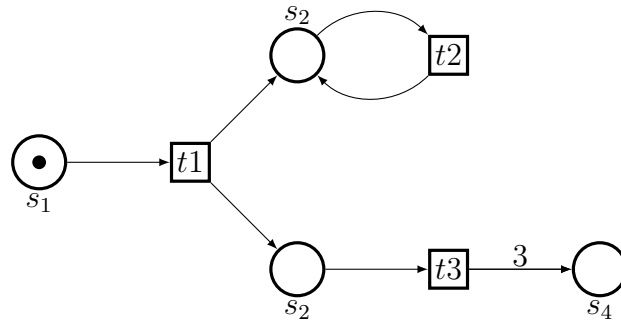


Abbildung 16: Ein beschränktes S/T-System

Das in Abbildung 16 dargestellte S/T-System ist 3-beschränkt, denn es gilt $\forall s \in S$ Folgendes: $M(s) \leq 3 \quad \forall M \in Reach(M_0)$. Die Stellen s_1, s_2 und s_3 sind 1-beschränkt. Die Stelle s_4 ist 3-beschränkt.

4.7 Deadlockfreies S/T-System

In einem deadlockfreien System kann immer eine Transition schalten. Ein System befindet sich jedoch in einem Deadlock, wenn keine Transition aktiviert ist.

4.7.1 Deadlockfreiheit

Ein S/T-System heißt deadlockfrei, falls bei jeder erreichbaren Markierung mindestens eine Transition aktiviert ist.

4.7.2 Beispiel

Das in Abbildung 17 dargestellte S/T-System ist deadlockfrei, denn die Transition t_1 kann bei jeder erreichbaren Markierung immer schalten.

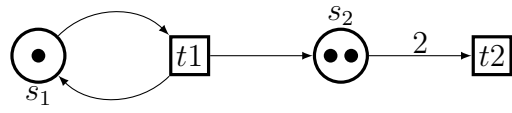


Abbildung 17: Ein deadlockfreies S/T-System

5 Reduktion des Hamiltonschen-Wege-Problems auf das Erreichbarkeitsproblem

In diesem Kapitel wird eine Übersetzung von Graphen in S/T-Systeme vorgestellt, so dass das Durchlaufen eines Weges im Graphen einer Schaltfolge im übersetzten S/T-System entspricht.

5.1 Konstruktion

Abbildung 18 zeigt, wie eine Kante von v nach v' in eine Transition mit Vorstelle $s(v)$ und Nachstelle $s(v')$ übersetzt wird.

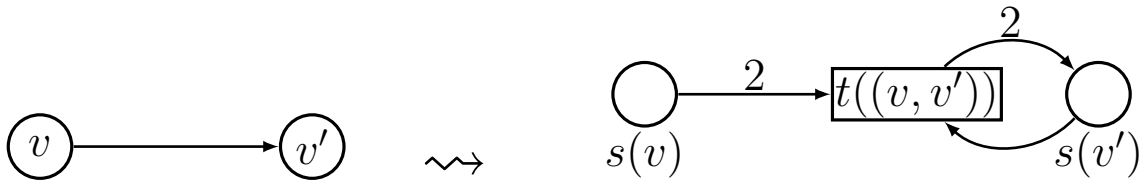


Abbildung 18: Von gerichteten Kanten zu Transitionen

Befindet man sich beim Durchlaufen eines einfachen Weges gerade in Knoten v , so soll die Stelle $s(v)$ genau zwei Token enthalten. Für jeden noch nicht besuchten Knoten v' soll die Stelle $s(v')$ genau einen Token enthalten. Nachdem $t((v, v'))$ geschaltet hat, besitzt $s(v)$ keinen Token und $s(v')$ zwei Tokens. Gäbe es im Graphen eine weitere Kante (v', v'') mit $v'' \neq v$, könnte nach dem Schalten von $t((v, v'))$ die Transition $t((v', v''))$ geschaltet werden, falls die Stelle $s(v'')$ einen Token enthält. Auf diese Weise kann das Durchlaufen einfacher Wege mit Schaltfolgen modelliert werden.

Abbildung 19 stellt dar, wie die durchlaufenen Kanten gezählt werden können:

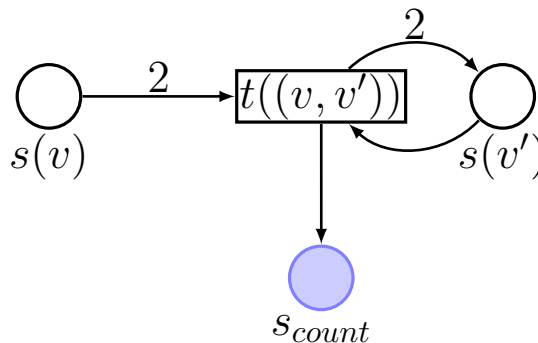


Abbildung 19: Erweiterung um Stelle s_{count}

Die Zahl der durchlaufenen Kanten im Graphen entspricht der Zahl der Tokens in Stelle s_{count} .

Formal ist die Konstruktion wie folgt definiert:

Sei $G = (V, E)$ ein gerichteter Graph und $\mathcal{PN}(G) = (S, T, F, W, M_0)$ das S/T-System mit:

$$S = \{s(v) \mid v \in V\} \uplus \{s_0, s_{count}\}$$

$$T = \{t(e) \mid e \in E\} \uplus \{begin(v) \mid v \in V\}$$

$$F = \{(s_0, begin(v)) \mid v \in V\} \cup \{(begin(v), s(v)) \mid v \in V\} \cup \\ \{(s(v), t(e)) \mid pr_1(e) = v\} \cup \{(t(e), s(v)) \mid pr_2(e) = v\} \cup \\ \{(s(v), t(e)) \mid pr_2(e) = v\} \cup \{(t(e), s_{count}) \mid e \in E\}$$

$$W(f) = \begin{cases} 2 & \text{falls } f \in \{(s(v), t(e)) \mid pr_1(e) = v\} \cup \{(t(e), s(v)) \mid pr_2(e) = v\} \\ 1 & \text{sonst} \end{cases}$$

$$M_0(s) = \begin{cases} 0 & \text{falls } s = s_{count} \\ 1 & \text{sonst} \end{cases}$$

Dabei bezeichnet für $e = (v, v')$ $pr_1(e) = v$ und $pr_2(e) = v'$, das heißt, pr_i ist die Projektion auf die i -te Komponente für $i = 1, 2$.

Die disjunkte Vereinigung von zwei Mengen A und B ist $A \uplus B = A \cup B$ unter der Bedingung, dass $A \cap B = \emptyset$ (Vgl. [PW08]).

Darüber hinaus sorgt die Stelle s_0 dafür, dass das Durchlaufen aller möglichen einfachen Wege im Graphen durch entsprechende Schaltfolgen in dem übersetzten S/T-System modelliert werden können. um außerdem darzustellen, ab welchem Knoten im Graphen jeder einfache Weg durchlaufen wurde, bekommt jede Stelle in dem übersetzten S/T-System eine *begin*-Transition als Vortransition, die ebenfalls die Stelle s_0 als Vorstelle besitzt.

5.2 Beispiel

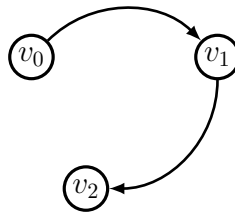


Abbildung 20: Graph G

Der in Abbildung 20 dargestellte gerichtete Graph dient als Beispiel für die Konstruktion. Da laut Konstruktion nur einfache Wege im Graphen durchlaufen werden, lässt sich jede durchlaufene Kante durch eine Transition im Stellen/Transitions-System modellieren.

Abbildung 21 stellt das konstruierte S/T-System dar. Das Durchlaufen des Weges $v_0v_1v_2$ in G wird durch die Schaltfolge $begin(v_0)t(v_0, v_1)t(v_1, v_2)$ im S/T-System $\mathcal{PN}(G)$ modelliert.

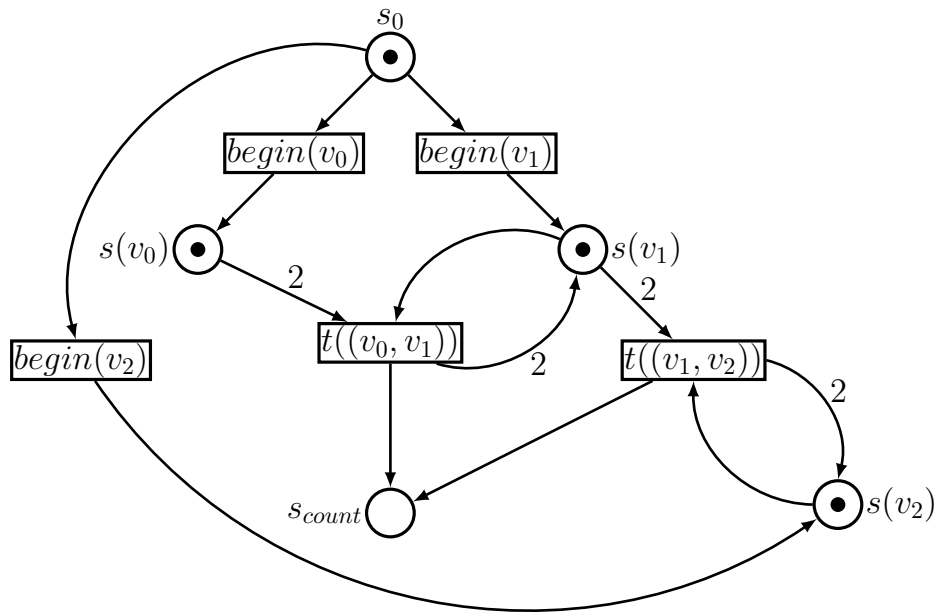


Abbildung 21: S/T-System ($\mathcal{PN}(G)$)

Der Erreichbarkeitsgraph von $\mathcal{PN}(G)$ aus Abbildung 21 ist in Abbildung 22 dargestellt, wobei die Reihenfolge der Stellen durch $s_0, s(v_0), s(v_1), s(v_2)$, und s_{count} gegeben ist.

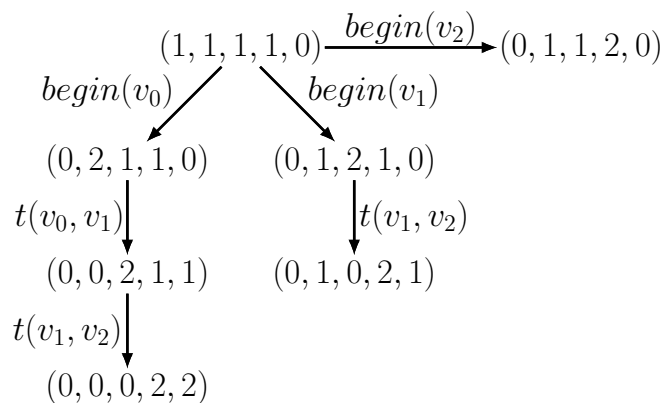


Abbildung 22: Erreichbarkeitsgraph

Aus dem Erreichbarkeitsgraph von $\mathcal{PN}(G)$ wird sichtlich, dass sich ausgehend von

der Startmarkierung die Schaltfolgen $begin(v_1)t(v_1, v_2)$, $begin(v_0)t(v_0, v_1)t(v_1, v_2)$ und $begin(v_2)$ schalten lassen. Von diesen simuliert aber nur die zweite einen Durchlauf durch einen Hamiltonschen Weg in G . Im Detail induziert der Weg $v_0v_1v_2$ im Beispielgraphen G aus Abbildung 20 die folgende Schaltfolge: $begin(v_0)t(v_0, v_1)t(v_1, v_2)$. Abbildung 23 zeigt das S/T-System nach dem Schalten von $begin(v_0)$.

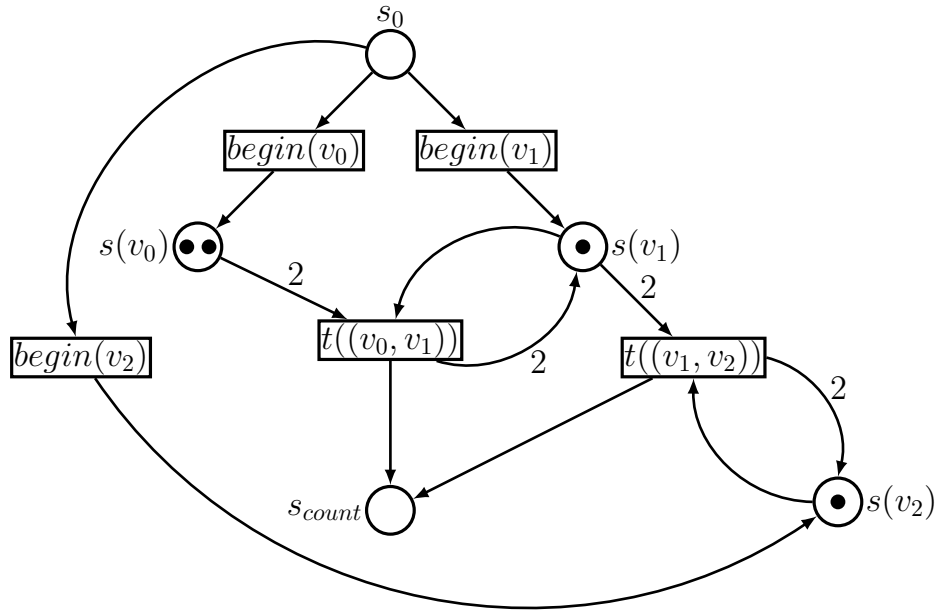


Abbildung 23: S/T-System ($\mathcal{PN}(G)$) nach dem Schalten von $begin(v_0)$

Abbildung 24 stellt das S/T-System nach dem Schalten von $t(v_0, v_1)$ dar.

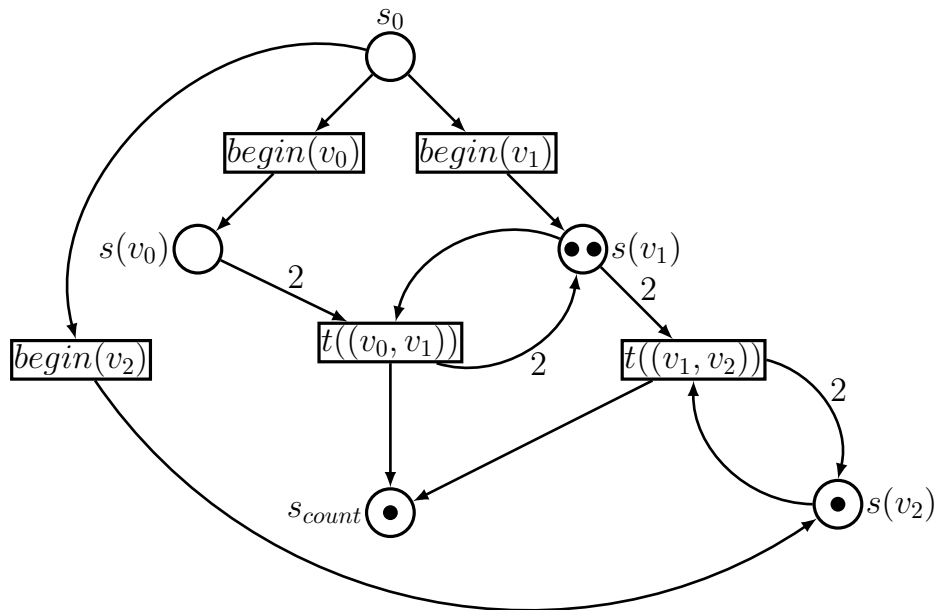


Abbildung 24: S/T-System ($\mathcal{PN}(G)$) nach dem Schalten von $t((v_0, v_1))$

Abbildung 25 stellt das S/T-System nach dem Schalten von $t(v_1, v_2)$ dar.

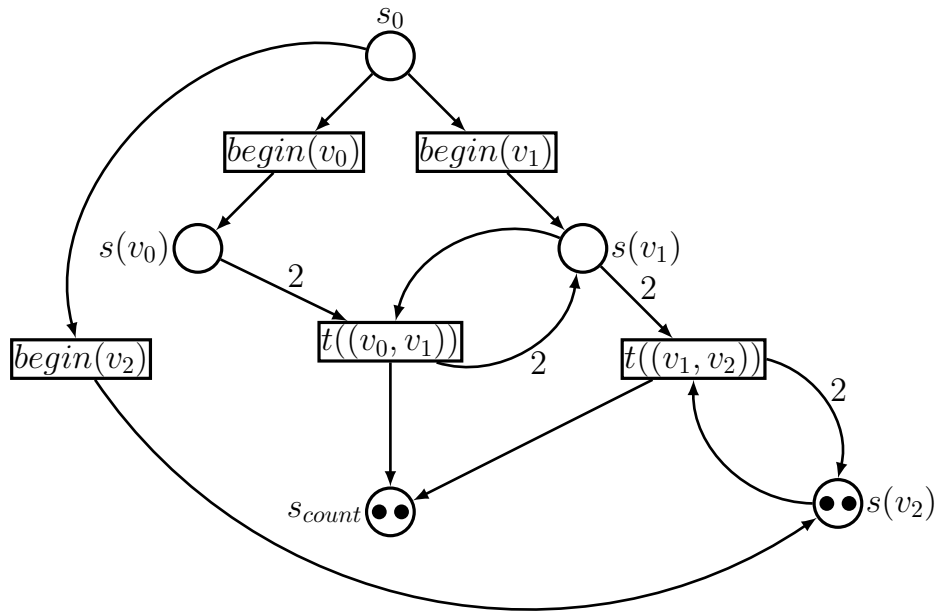


Abbildung 25: S/T-System ($\mathcal{PN}(G)$) nach dem Schalten von $t((v_1, v_2))$

5.3 Korrektheit

In diesem Abschnitt wird gezeigt, dass in G genau dann ein Hamiltonscher Weg existiert, wenn in $\mathcal{PN}(G)$ eine Markierung M mit $M(s_{count}) = |V| - 1$ erreichbar ist.

Das folgende Lemma besagt, dass, wenn ein Graph $G = (V, E)$ einen einfachen Weg $p = v_0 \dots v_n$ enthält, dann gibt es in $\mathcal{PN}(G)$ eine erreichbare Markierung M mit $M(s_{count}) = n$, die durch Schalten einer Schaltfolge von der Startmarkierung M_0 erreichbar ist.

Lemma 1:

Sei $G = (V, E)$ ein Graph. Dann gilt: G hat einen einfachen Weg $p = v_0 \dots v_n \implies M_0[\text{begin}(v_0) t((v_0, v_1)) \dots t((v_{n-1}, v_n))]M$ in $\mathcal{PN}(G)$ mit

$$M(s) = \begin{cases} 0 & \text{falls } s \in \{s(v_0), \dots, s(v_{n-1})\} \cup \{s_0\} \\ 2 & \text{falls } s = s(v_n) \\ n & \text{falls } s = s_{count} \\ 1 & \text{sonst.} \end{cases}$$

Beweis. (durch vollständige Induktion über die Länge der Schaltfolgen n)

Induktionsanfang (I.A): Für $n = 0$ gilt $p = v_0$ und $M_0[begin(v_0)]M$ mit $M(s_0) = 0$, $M(s(v_0)) = 2$, $M(s_{count}) = 0$ und $M(s) = 1$ für alle $s \in S \setminus \{s_0, s(v_0), s_{count}\}$

Induktionsschritt (I.S): Betrachte $p = v_0 \dots v_{n+1}$.

Nach Induktionsvoraussetzung(I.V): $M_0[begin(v_0)t((v_0, v_1)) \dots t((v_{n-1}, v_n))]M$ mit

$$M(s) = \begin{cases} 0 & \text{falls } s \in \{s(v_0), \dots, s(v_{n-1})\} \cup \{s_0\} \\ 2 & \text{falls } s = s(v_n) \\ n & \text{falls } s = s_{count} \\ 1 & \text{sonst} \end{cases}$$

Da $M(s(v_n)) = 2$ und $M(s(v_{n+1})) = 1$, ist die Transition $t(v_n, v_{n+1})$ aktiviert. Nach dem Schalten von $t(v_n, v_{n+1})$ bei M , erhalten wir die Markierung M' mit

$$M'(s) = \begin{cases} 0 & \text{falls } s \in \{s(v_0), \dots, s(v_n)\} \cup \{s_0\} \\ 2 & \text{falls } s = s(v_{n+1}) \\ n + 1 & \text{falls } s = s_{count} \\ 1 & \text{sonst.} \end{cases}$$

■

Das folgende Lemma besagt, dass, wenn es in $\mathcal{PN}(G)$ eine erreichbare Markierung M gibt, die durch Schalten einer nichtleeren Schaltfolge von der Startmarkierung M_0 erreichbar ist, dann hat diese Schaltfolge die Form $begin(v_0)t((v_0, v_1)) \dots t((v_{n-1}, v_n))$.

Lemma 2:

Sei $M_0[w]M$ in $\mathcal{PN}(G)$ eine Schaltfolge mit $|w| \geq 1$.

Dann hat w die Form $begin(v_0)t((v_0, v_1)) \dots t((v_{n-1}, v_n))$ und es gilt:

$$M(s) = \begin{cases} 0 & \text{falls } s \in \{s(v_0), \dots, s(v_{n-1})\} \cup \{s_0\} \\ 2 & \text{falls } s = s(v_n) \\ n & \text{falls } s = s_{count} \\ 1 & \text{sonst.} \end{cases}$$

Beweis. (durch vollständige Induktion über die Länge der Schaltfolgen n)

I.A: Für $|w| = 1$

(Konstr.) $\implies w = \text{begin}(v)$ für $v \in V$.

I.S: $M_0[w > \widetilde{M}[t > M$

(I.V) $\implies w = \text{begin}(v_0)t((v_0, v_1)) \dots t((v_{n-1}, v_n))$ und

$$\widetilde{M}(s) = \begin{cases} 0 & \text{falls } s \in \{s(v_0), \dots, s(v_{n-1})\} \cup \{s_0\} \\ 2 & \text{falls } s = s(v_n) \\ n & \text{falls } s = s_{count} \\ 1 & \text{sonst} \end{cases}$$

Da $\widetilde{M}(s(v_n)) = 2$ und $\widetilde{M}(s(v_{n+1})) = 1$, ist die Transition $t(v_n, v_{n+1}) = t$ mit $v_{n+1} \neq v_i$ für $i = 1, \dots, n$ aktiviert. Nach dem Schalten von $t(v_n, v_{n+1})$ bei \widetilde{M} , erhalten wir die Markierung M mit

$$M(s) = \begin{cases} 0 & \text{falls } s \in \{s(v_0), \dots, s(v_n)\} \cup \{s_0\} \\ 2 & \text{falls } s = s(v_{n+1}) \\ n + 1 & \text{falls } s = s_{count} \\ 1 & \text{sonst.} \end{cases}$$

■

Das folgende Lemma besagt, dass, wenn es einen Graphen gibt, so dass es in einem Stellen/Transitions-System eine erreichbare Markierung M existiert, die durch Schalten einer Schaltfolge von der Startmarkierung M_0 erreichbar ist, dann enthält dieser Graph einen einfachen Weg.

Lemma 3:

$M_0[\text{begin}(v_0)t((v_0, v_1)) \dots t((v_{n-1}, v_n))]M \implies v_0 \dots v_n$ ist ein einfacher Weg in G .

Beweis. (durch vollständige Induktion über die Länge der Schaltfolge n)

I.A: Für $n = 0$, v_0 ist offensichtlich ein einfacher Weg in G .

I.S: $M_0[begin(v_0)t((v_0, v_1)) \dots t((v_{n-1}, v_n))\widetilde{M}[t((v_n, v_{n+1}))M$.

Nach I.V ist $v_0 \dots v_n$ ein einfacher Weg in G . Nach Konstruktion von $\mathcal{PN}(G)$, \exists eine Kante (v_n, v_{n+1}) in G . Da nach Lemma 2 $\widetilde{M}(v_i) = 0$ für $i = 0, \dots, n - 1$ und nach Definition von Schalten die Nachstelle $s(v_{n+1})$ von $t((v_n, v_{n+1}))$ mindestens einen Token enthalten muss, gilt $s(v_i) \neq s(v_{n+1})$ für $i = 0, \dots, n - 1$. Außerdem gilt nach Konstruktion $s(v_{n+1}) \neq s(v_n)$, da es keine Schlingen in G gibt.

Folglich gilt $s(v_{n+1}) \neq s(v_i)$ für $i = 0, \dots, n$. Somit ist $v_0 \dots v_{n+1}$ ein einfacher Weg in G . ■

Der folgende Satz besagt, dass es genau dann einen Hamiltonschen Weg in $G = (V, E)$ (G soll aus mindestens zwei Knoten bestehen) gibt, wenn eine von M_0 erreichbare Markierung M existiert, so dass $M(s_{count}) = |V| - 1$. Dieser Satz folgt aus Lemma1, Lemma2 und Lemma 3.

Satz 5:

Sei $G = (V, E)$ ein Graph mit $|V| \geq 2$. Dann gilt: $\mathcal{HAMPATH}(G) = Ja \iff \exists M \in Reach(\mathcal{PN}(G))$ mit $M(s_{count}) = |V| - 1$.

Beweis.

(\implies)

Sei $\mathcal{HAMPATH}(G) = Ja$. Dann gilt nach Definition von $\mathcal{HAMPATH}$, dass es einen einfachen Weg $v_0 \dots v_n$ in G mit $n = |V| - 1$ Kanten existiert.

Nach Lemma 1 gilt nun $M_0[begin(v_0)t((v_0, v_1)) \dots t((v_n, v_{n-1}))M$ mit $M(s_{count}) = n = |v| - 1$.

Dann gilt nach Definition von $Reach$, dass $M \in Reach(\mathcal{PN}(G))$ mit $M(s_{count}) = n = |V| - 1$.

(\impliedby)

$M \in Reach(\mathcal{PN}(G))$ mit $M(s_{count}) = |V| - 1$. Dann gibt es eine Schaltfolge w , sodass $M_0[w > M$ mit $M(s_{count}) = |V| - 1$.

Da $|V| - 1 \geq 1$ und die Stelle s_{count} in M_0 keinen Token ($M_0(s_{count}) = 0$) enthält, muss w mindestens die Länge 1 ($|w| \geq 1$) haben.

Nach Lemma 2 gilt Folgendes: $w = begin(v_0)t((v_0, v_1)) \dots t((v_{|V|-2}, v_{|V|-1}))$ mit $M(s_{count}) = |V| - 1$.

Nach Lemma 3 gilt Folgendes: Es existiert einen einfachen Weg $v_0 \dots v_n$ in G mit $n = |V| - 1$ Kanten.

$$\implies \mathcal{HAMPATH}(G) = Ja.$$

■

Lemma 4:

$\mathcal{PN}(G)$ ist in Polynomialzeit konstruierbar.

Beweis.

Für die Konstruktion von $\mathcal{PN}(G)$ werden $|V| + 2$ Stellen, $|E| + |V|$ Transitionen und $2(|V|) + 3(|V| - 1) + |E|$ Kanten benötigt. Daraus folgt, dass $\mathcal{PN}(G)$ in Polynomialzeit konstruierbar ist. ■

Corollar:

1. $\mathcal{HAMPATH} \leq_P \mathcal{TEP}$.

Das Lösen von $\mathcal{HAMPATH}$ lässt sich in Polynomialzeit auf das Lösen von \mathcal{TEP} reduzieren. Außerdem ist \mathcal{TEP} laut Beobachtung 4 entscheidbar. Daraus folgt, dass $\mathcal{HAMPATH}$ auch entscheidbar ist.

2. \mathcal{TEP} ist NP-hart.

Dem Satz 3 zufolge gilt folgendes:

$$\mathcal{HAMPATH} \leq_P \mathcal{TEP} \text{ und } \mathcal{HAMPATH} \text{ ist NP-hart} \implies \mathcal{TEP} \text{ ist NP-hart.}$$

Beobachtung 5:

Sei $G = (V, E)$ ein Graph. Dann ist $\mathcal{PN}(G)$ beschränkt.

Beweis.

Sei $M \in \text{Reach}(M_0)$ mit $M \neq M_0$. \implies es existiert eine Schaltfolge w mit $|w| \geq 1$ und $M_0[w]M$.

Aus Lemma 2 folgt:

$$w = \text{begin}(v_0)t((v_0, v_1)) \dots t((v_{n-1}, v_n)) \text{ und}$$
$$M(s) = \begin{cases} 0 & \text{falls } s \in \{s(v_0), \dots, s(v_{n-1})\} \cup \{s_0\} \\ 2 & \text{falls } s = s(v_n) \\ n & \text{falls } s = s_{\text{count}} \\ 1 & \text{sonst.} \end{cases}$$

Aus Lemma 3 folgt, dass $v_0 \dots v_n$ ein einfacher Weg in G ist.

Nach Definition von einfachem Weg folgt $n \leq |v| - 1$.

Es folgt nun, dass $\forall v_i \in V$ $s(v)$ 2-beschränkt ist; s_{count} ist $|V| - 1$ beschränkt und s_0 ist 1-beschränkt. ■

6 Reduktion des Rundreiseproblems auf das Erreichbarkeitsproblem

Dieses Kapitel beschäftigt sich mit der Übersetzung von $\mathcal{TSPPATH}$ in ein S/T-System. Diese Übersetzung wird anhand von Abbildungen und am Ende des Kapitels mit einem Beispiel veranschaulicht. Außerdem wird seine Korrektheit mit Hilfe von Lemmata bewiesen.

Der folgende Abschnitt beschreibt, wie sich jede Eingabe für $\mathcal{TSPPATH}$ mit S/T-System konstruieren lässt.

6.1 Konstruktion

Wie in Kapitel 5 beschrieben, kann das Durchlaufen von einfachen Wegen mit Schaltfolgen modelliert werden. Diese Idee wird hier aufgegriffen. Zusätzlich muss man sich beim Schalten einer Transition $t(v, v')$ die Distanz der Kante (v, v') merken. Dies geschieht in der Stelle s_{count} . Abbildung 26 zeigt ein Beispiel für die Übersetzung einer Kante (v, v') in die Transition $t(v, v')$ mit Vor- und Nachstellen sowie den entsprechenden Flusskanten.

Die Anzahl der Tokens in der Stelle s_{count} erhöht sich beim Schalten von $t(v, v')$ um $dist(v, v') = 4$ und entspricht der Distanz der durchlaufenen Kante (v, v') im Graphen.

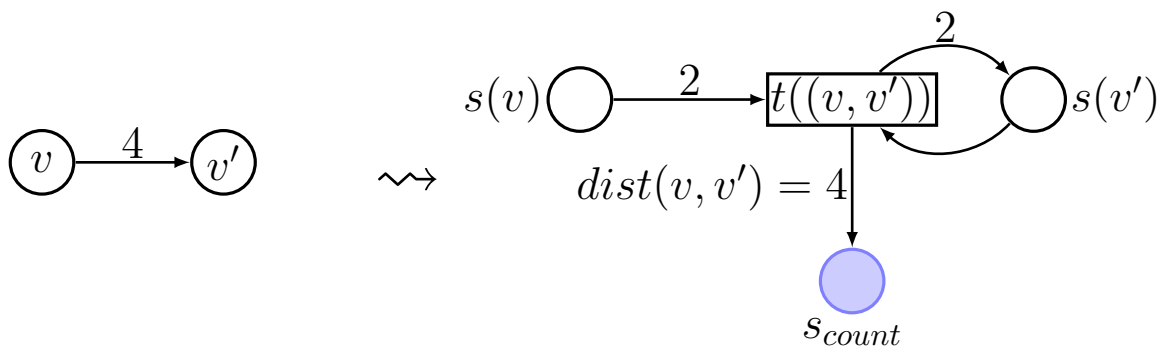


Abbildung 26: Übersetzung einer gewichteten Kante

Würde zum Beispiel eine andere Transition $t(v', v'')$ anschließend schalten, würde die Stelle s_{count} zusätzliche $dist(v', v'')$ Tokens dazu bekommen.

Auf diese Weise kann das Durchlaufen einfacher Wege modelliert werden, wobei die Distanz der einzelnen Kanten als Tokenzahl in s_{count} gespeichert wird. Somit entspricht die Distanz eines durchlaufenen Hamiltonschen Weges im Graphen der Anzahl der Tokens in Stelle s_{count} .

Die formale Übersetzung gewichteter Graphen ist wie folgt definiert:

Sei $G = (V, E)$ ein gerichteter Graph mit Kantengewichtungen, $dist : E \rightarrow \mathbb{N}$. Dann ist $\mathcal{PN}(G, dist) = (S, T, F, W, M_0)$ das S/T-System, in dem S, T, F und M_0 wie in Konstruktion 5.1 definiert sind, aber der Vollständigkeit halber hier noch einmal aufgeführt sind.

$$S = \{s(v) \mid v \in V\} \uplus \{s_0, s_{count}\}$$

$$T = \{t(e) \mid e \in E\} \uplus \{begin(v) \mid v \in V\}$$

$$F = \{(s_0, begin(v)) \mid v \in V\} \cup \{(begin(v), s(v)) \mid v \in V\} \cup \\ \{(s(v), t(e)) \mid pr_1(e) = v\} \cup \{(t(e), s(v)) \mid pr_2(e) = v\} \cup \\ \{(s(v), t(e)) \mid pr_2(e) = v\} \cup \{(t(e), s_{count}) \mid e \in E\}$$

$$W(f) = \begin{cases} 2 & \text{falls } f \in \{(s(v), t(e)) \mid pr_1(e) = v\} \cup \{(t(e), s(v)) \mid pr_2(e) = v\} \\ dist(e) & \text{falls } f \in \{(t(e), s_{count}) \mid e \in E\} \\ 1 & \text{sonst} \end{cases}$$

$$M_0(s) = \begin{cases} 0 & \text{falls } s = s_{count} \\ 1 & \text{sonst} \end{cases}$$

6.2 Korrektheit

In diesem Abschnitt wird gezeigt, dass in G genau dann ein Hamiltonscher Weg existiert, dessen Distanz höchstens k ($k \in \mathbb{N}$) beträgt, wenn in $\mathcal{PN}(G, dist)$ eine erreichbare Markierung M mit $M(s_{count}) \leq k$ gibt.

Das folgende Lemma besagt, dass zu jedem Hamiltonschen Weg p mit der Distanz höchstens k eine erreichbare Markierung M in $\mathcal{PN}(G, dist)$ mit $M(s_{count}) = dist(p)$ gibt, die durch Schalten einer Schaltfolge von der Startmarkierung M_0 erreichbar ist. Die Beweise von Lemma 5, Lemma 6 und Lemma 7 gehen analog zu den Beweisen von Lemma 1, Lemma 2 und Lemma 3 in Kapitel 5 und werden daher weggelassen.

Lemma 5:

Sei $G = (V, E)$ ein gerichteter Graph, $dist : E \rightarrow \mathbb{N}$ und sei $k \in \mathbb{N}$. Dann gilt: G hat einen einfachen Weg $p = v_0 \dots v_n \implies M_0[begin(v_0) t((v_0, v_1)) \dots t((v_{n-1}, v_n))]M$ in $\mathcal{PN}(G, dist)$ mit

$$M(s) = \begin{cases} 0 & \text{falls } s \in \{s(v_0), \dots, s(v_{n-1})\} \cup \{s_0\} \\ 2 & \text{falls } s = s(v_n) \\ \text{dist}(p) & \text{falls } s = s_{\text{count}} \\ 1 & \text{sonst} \end{cases}$$

Das folgende Lemma besagt, dass jede nicht leere Schaltfolge in $\mathcal{PN}(G, \text{dist})$, die von der Startmarkierung M_0 zu einer erreichbaren Markierung M schaltet, die Form $\text{begin}(v_0)t((v_0, v_1)) \dots t((v_{n-1}, v_n))$ hat, mit $M(s_{\text{count}}) = \sum_{i=1}^n \text{dist}((v_{i-1}, v_i))$

Lemma 6:

Sei $M_0[w]M$ in $\mathcal{PN}(G, \text{dist})$ eine Schaltfolge mit $|w| \geq 1$.

Dann hat w die Form $\text{begin}(v_0)t((v_0, v_1)) \dots t((v_{n-1}, v_n))$ und es gilt:

$$M(s) = \begin{cases} 0 & \text{falls } s \in \{s(v_0), \dots, s(v_{n-1})\} \cup \{s_0\} \\ 2 & \text{falls } s = s(v_n) \\ \sum_{i=1}^n \text{dist}((v_{i-1}, v_i)) & \text{falls } s = s_{\text{count}} \\ 1 & \text{sonst} \end{cases}$$

Das folgende Lemma besagt, dass, wenn eine Schaltfolge $\text{begin}(v_0)t((v_0, v_1)) \dots t((v_{n-1}, v_n))$ von der Startmarkierung M_0 zu einer erreichbaren Markierung M schaltet, dann stellt $v_0 \dots v_n$ einen einfachen Weg in G dar, dessen Distanz gleich $M(s_{\text{count}})$ ist.

Lemma 7:

$M_0[\text{begin}(v_0)t((v_0, v_1)) \dots t((v_{n-1}, v_n))]M \implies p = v_0 \dots v_n$ ist ein einfacher Weg in G mit $\text{dist}(p) = M(s_{\text{count}})$.

Der folgende Satz besagt, dass es genau dann einen Hamiltonschen Weg in einem Graphen mit mindestens zwei Knoten gibt, dessen Distanz höchstens k ($k \in \mathbb{N}$) beträgt, wenn es eine von M_0 erreichbare Markierung M mit $M(s_{\text{count}}) \leq k$ existiert.

Dieser Satz folgt aus Lemma 5, Lemma 6 und Lemma 7.

Satz 6:

Sei $G = (V, E)$ ein gerichteter Graph, $\text{dist} : E \rightarrow \mathbb{N}$, $k \in \mathbb{N}$ und $|V| \geq 2$. Dann gilt:

$\mathcal{TSPPATH}(G, dist, k) = Ja \iff \exists M \in Reach(\mathcal{PN}(G, dist))$ mit $M(s_{count}) \leq k$,
 $M(s(\hat{v})) = 2$ für ein $\hat{v} \in V$ und $M(s(v)) = 0$ für alle $v \in V \setminus \{\hat{v}\}$.

Beweis.

(\implies)

Sei $\mathcal{TSPPATH}(G, dist, k) = Ja$. Dann gilt nach Definition von $\mathcal{TSPPATH}$, dass es einen einfachen Weg $p = v_0 \dots v_n$ in G mit $dist(p) \leq k$ existiert.

Nach Lemma 5 gilt nun $M_0[begin(v_0)t((v_0, v_1)) \dots t((v_n, v_{n-1}))]M$ mit $M(s_{count}) = dist(p) \leq k$, $M(s(\hat{v})) = 2$ für ein $\hat{v} \in V$ und $M(s(v)) = 0$ für alle $v \in V \setminus \{\hat{v}\}$.

Nach Definition von $Reach$ gilt $M \in Reach(\mathcal{PN}(G))$ mit $M(s_{count}) = dist(p) \leq k$, $M(s(\hat{v})) = 2$ für ein $\hat{v} \in V$ und $M(s(v)) = 0$ für alle $v \in V \setminus \{\hat{v}\}$.

(\impliedby)

$M \in Reach(\mathcal{PN}(G, dist))$ mit $M(s_{count}) \leq k$, $M(s(\hat{v})) = 2$ für ein $\hat{v} \in V$ und $M(s(v)) = 0$ für alle $v \in V \setminus \{\hat{v}\}$. Dann gibt es eine Schaltfolge w , sodass $M_0[w > M$ mit $M(s_{count}) \leq k$.

Da $M_0(s_0) = 1$, muss w mindestens die Länge 1 ($|w| \geq 1$) haben.

Nach Lemma 6 gilt Folgendes: $w = begin(v_0)t((v_0, v_1)) \dots t((v_{|V|-2}, v_{|V|-1}))$ mit $M(s_{count}) \leq k$.

Nach Lemma 7 gilt Folgendes: Es existiert einen einfachen Weg $p = v_0 \dots v_n$ in G mit $n = |V| - 1$ Kanten und $dist(p) = M(s_{count}) \leq k$.

$\implies \mathcal{TSPPATH}(G, dist, k) = Ja$. ■

6.3 Beispiel

Betrachte den gerichteten Graph $G = (V, E)$ mit Kantengewichtungen $dist$ in Abbildung 27.

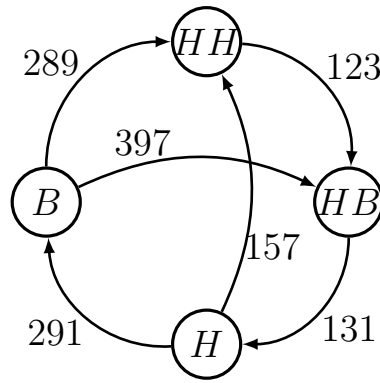


Abbildung 27: Graph G

Legende: HH = Hamburg, HB = Bremen, H = Hannover, B = Berlin.
 In Abbildung 28 ist das übersetzte S/T-System dargestellt.

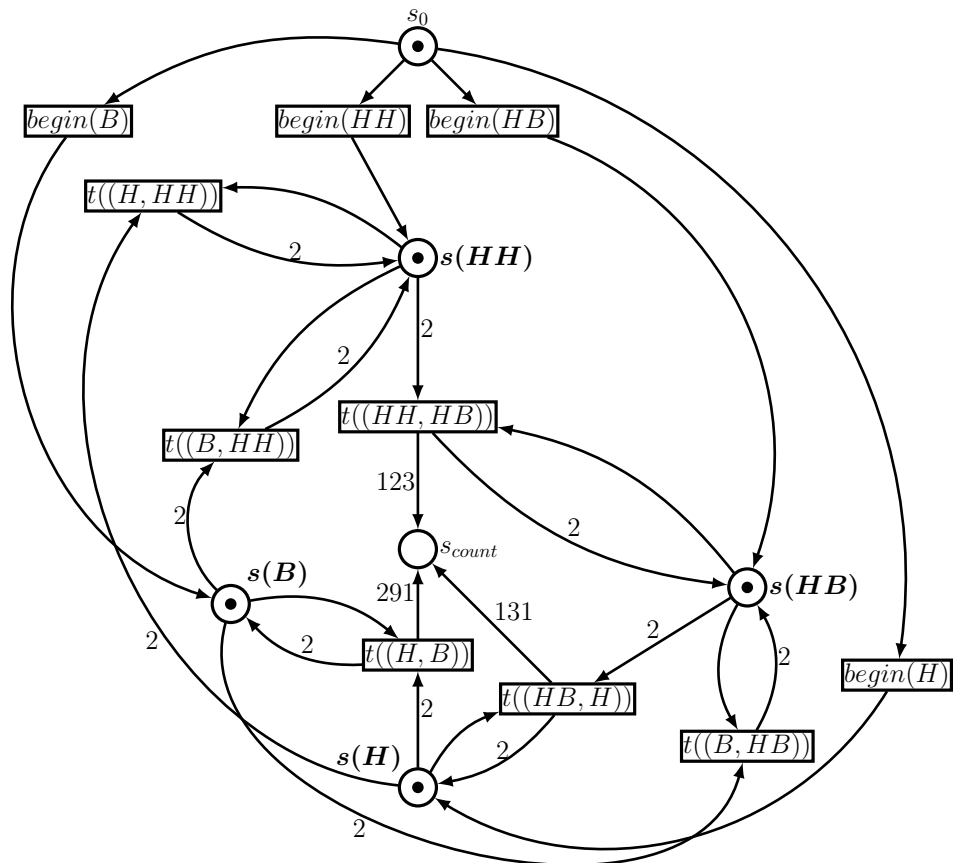


Abbildung 28: S/T-System($\mathcal{PN}(G, dist, 550)$)

Frage: Gibt es in G einen Hamiltonschen Weg, dessen Distanz höchsten 550 ist?
 Wird nun die Schaltfolge $begin(HH)t(HH, HB)t(HB, H)t(H, B)$ geschaltet, entsteht die Markierung M mit folgenden Eigenschaften:
 $M(s_{count}) = dist(HH, HB) + dist(HB, H) + dist(H, B) = 123 + 131 + 291 = 545 \leq 550$

7 Zusammenfassung und Ausblick

Das Hauptziel dieser Arbeit war es, auf die Frage zu beantworten, ob und inwiefern sich Entscheidungsprobleme für Graphen auf das Erreichbarkeitsproblem für Petri-Netze reduzieren lassen. Dafür wurden im Laufe der Arbeit einige Entscheidungsprobleme für gerichtete Graphen behandelt: Das Hamiltonsche-Wege-Problem und das Traveling-Salesman-Problem für Wege. Die allgemeine Idee, die verfolgt wurde, war es, herauszufinden, wie das Durchlaufen eines Weges im Graphen durch eine Schaltfolge im S/T-System übersetzt werden könnte. Diese ließ sich mit Hilfe von Konstruktionen und Beispielen umsetzen. Anschließend wurde die Übersetzung auf Korrektheit durch Beweise von Lemmata und Sätzen überprüft. So ließ sich zeigen, dass sich Entscheidungsprobleme für Graphen auf das Erreichbarkeitsproblem für Petri-Netze reduzieren lassen. Die mögliche Anwendung von Petri-Netz-Werkzeugen bei der Lösung von Graphproblemen wurde somit in dieser Arbeit bestätigt. Es sollte jedoch nicht außer Acht gelassen werden, dass die generierten Petri-Netze schon bei Graphen mit mehr als vier Knoten unübersichtlich sind, was dazu führen kann, dass sich dadurch die Fehleranfälligkeit bei der Modellierung und die Suche nach möglich aufgetretenen Fehlern nach der Modellierung erhöhen.

Angesichts des resultierenden vorgestellten, positiven Ergebnisses aus dieser Arbeit wäre die Durchführung eines Eulersch-Test mit einem Petri-Netz-Werkzeug durchaus möglich. Eine weitere mögliche Fortsetzung dieser Arbeit wäre zum Beispiel die Übersetzung von Graphen-Traversierungsalgorithmen in Petri-Netze. Darüber hinaus wäre es gut vorstellbar, dass sich auch andere Entscheidungsprobleme wie Vertex Cover, das Färbungsproblem (COLOR), das Cliquesproblem (CLIQUE) und Independent Set auf das Erreichbarkeitsproblem für Petri-Netze reduzieren lassen.

Literaturverzeichnis

- [AB02] Alexander Asteroth and Christel Baier. *Eine Einführung in Berechenbarkeit, Komplexität und formale Sprachen mit 101 Beispielen*. Pearson Studium, München, p.85ff, 2002.
- [BGG10] Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Springer Verlag, Berlin, p.1, 2010.
- [BKI14] Christoph Beierle and Gabriele Kern-Isberner. *Methoden wissenbasierter Systeme: Grundlagen, Algorithmen, Anwendungen*. 5.Auflage. Springer-Verlag, p.507, September 2014.
- [dW10] Bayerische Akademie der Wissenschaften. *Jahrbuch 2009*. .H.Beck Verlag, München, p. 254, 2010.
- [GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [HMU02] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie*. 2.Auflage. Pearson Studium, 2002.
- [Ill09] Johann Anton Illik. *von der Automatentheorie zu Algorithmen und Datenstrukturen*. expert Verlag, 2009.
- [KB18] Uwe Kastens and Hans Kleine Büning. *Modellierung: Grundlagen und formale Methoden*. 4. erweiterte Auflage. Hanser Verlag, 2018.
- [KN09] Sven Oliver Krumke and Hartmut Noltemeier. *Graphentheoretische Konzepte und Algorithmen*. 2.Auflage. Vieweg+Teubner Verlag, p.50, Juli 2009.
- [Kus17] Sabine Kuske. *Algorithmen auf Graphen*. Vorlesungsskript , Sommersemester 2017.
- [Kus18] Sabine Kuske. *Petri-Netze*. Vorlesungsskript , Wintersemester 2017/2018.
- [May80] Ernst Wilhelm Mayr. *Ein Algorithmus für das allgemeine Erreichbarkeitsproblem bei Petrinetzen und damit zusammenhängende Probleme*. Institut für Informatik, Technische Universität München. 1980.
- [Nit04] Manfred Nitzsche. *Graphen für Einsteiger*. 1.Auflage. Vieweg Verlag, 2004.
- [PW08] Lutz Priese and Harro Winkelmann. *Petri-Netze*. 2.Auflage. Springer Verlag, 2008.
- [RE98] Grzegorz Rozenberg and Joost Engelfriet. *Elementary Net Systems*. In Reisig and Rozenberg [RR98], p.12-121, 1998.
- [RR98] Wolfgang Reisig and Grzegorz Rozenberg, editors. *Lectures on Petri Nets 1: Basic Models, Advances in Petri Nets*, volume 1491 of Lecture Notes in Computer Science. Springer Verlag, 1998.
- [Sta90] Peter H. Starke. *Analyse von Petri-Netz-Modellen*. B.G. Teubner Stuttgart, 1990.

- [Tur09] Volker Turau. *Algorithmische Graphentheorie*. 3.Auflage. Oldenbourg Verlag München, p.305, 2009.
- [Wim08] Harro Wimmel. *Entscheidbarkeit bei Petri Netzen*. Springer-Verlag, p.20, 2008.
- [WW13] Karsten Weicker and Nicole Weicker. *Algorithmen und Datenstrukturen*. Springer Vieweg verlag, p.35, 2013.

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als solche kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Unterschrift :

Ort, Datum :

