

Prüfer: Prof. Dr. Volker Claus
Betreuer: Dipl. Inf. Friedhelm Buchholz
Dipl. Math. Nicole Weicker

begonnen am: 10. Januar 1996

beendet am: 9. Juli 1996

CR-Klassifikation: F.2.2, D.2.2, I.6.3

Diplomarbeit Nr. 1373

Verkehrsmodellierung

Markus Schaal

Institut für Informatik
Universität Stuttgart
Breitwiesenstraße 20–22
D–70565 Stuttgart

Zusammenfassung

Das Mobilitätsbedürfnis der Menschen führt zu einem ständig anwachsenden motorisierten Individualverkehr mit den bekannten negativen Auswirkungen auf unseren Lebensraum. Die Simulation von Verkehr kann dazu beitragen, die bestehenden Probleme besser analysieren zu können und sinnvolle Gegenmaßnahmen zu entwickeln.

Viele Simulationsmodelle sind jedoch zu grob, um die tatsächlichen Abläufe im Verkehr ausreichend genau widerzuspiegeln. Insbesondere werden viele Einflußfaktoren, wie z.B. Wetter, nicht berücksichtigt.

Eine feinkörnige Simulation scheitert normalerweise entweder an Datenmangel oder an fehlender Rechnerkapazität. Es muß also ein Simulationsmodell entwickelt werden, dessen Granularität den zur Verfügung stehenden Daten angepaßt werden kann und dessen Simulation ausreichend schnell ist, um reale Probleme damit bearbeiten zu können.

In dieser Arbeit wurde ein solches Simulationsmodell für den öffentlichen Personennahverkehr entwickelt und formal spezifiziert. Die Daten wurden vom Institut für Eisenbahn- und Verkehrswesen der Universität Stuttgart übernommen. Ein Prototyp des Modells wurde mit Hilfe von THOR-Netzen, einer Art höherer Petri-Netze, erstellt. Durch einige Simulationsläufe konnten erste Ergebnisse zur Simulationsgeschwindigkeit gesammelt werden.

Durch ein großes Beispiel wird demonstriert, daß die Simulation von öffentlichem Nahverkehr mit THOR-Netzen auch für reale Anwendungen geeignet ist. Die vielfältigen Erweiterungsmöglichkeiten werden im letzten Teil dieser Arbeit angesprochen.

Inhaltsverzeichnis

1	Einleitung	4
1.1	Motivation	4
1.2	Verkehr – Was ist das?	6
1.3	Aufgabenstellung	7
1.4	Übersicht	8
2	Prognose der Verkehrsnachfrage	10
2.1	Einführung ins Vierstufenmodell	12
2.2	Beispiel einer Planung	13
2.3	Zielgröße und Eingangsgrößen	14
2.3.1	Raum- und Siedlungsstruktur	15
2.3.2	Verkehrsangebot	17
2.3.3	Verkehrsnachfrage	26
2.4	Prognose mit dem Vierstufenmodell	27
2.5	Kritik	29
3	Simulationsmodell für den ÖPNV	31
3.1	Die Konzepte	31
3.1.1	Verkehrsobjekte und Verkehrsnetze	32
3.1.2	Routen	33
3.1.3	Das Zeitkonzept	35
3.1.4	Das Mitnahme-/Mitfahrkonzept	35
3.1.5	Fahrtverlauf	37
3.1.6	Kapazitäten	39
3.1.7	Erweiterungen	39
3.2	Formalisierung der Konzepte (Spezifikation)	40
3.2.1	Verkehrsnetze	40
3.2.2	Verkehrsobjekte	45
3.2.3	Zustände von Verkehrsobjekten	47
3.2.4	Übergänge auf Zuständen	47

3.2.5	Mitnahmesystem	51
3.2.6	Konfiguration von Mitnahmesystemen	53
3.2.7	Simulation	54
4	Modellierung am Beispiel Stuttgart	60
4.1	Das Fußwegenetz	62
4.2	Verkehrsnetze für U- und S-Bahn	62
4.2.1	Verkehrsnetz ohne Trennung der Fahrtrichtungen	63
4.2.2	Trennung der Fahrtrichtungen (Problem)	65
4.2.3	Verfeinerung von Orten	66
4.2.4	Abbiegehaltepunkte	68
4.2.5	Trennung der Fahrtrichtungen (Lösung)	71
4.2.6	Verschmelzung von Haltepunktbeziehungen	74
4.2.7	Verkehrsnetz mit Trennung der Fahrtrichtungen	79
4.3	Die Mitnahmebeziehung	80
4.4	Die Fahrplanmittel	81
4.5	Die Personen	82
5	Einführung in THOR-Netze	83
5.1	Vorstellung der Konzepte	83
5.1.1	Stellen mit Struktur	83
5.1.2	C++-Objekte als Marken	84
5.1.3	Kantentypen	86
5.1.4	Transitionen mit Programmierung	86
5.1.5	Die beiden Zeitkonzepte	87
5.1.6	Das Hierarchiekonzept	88
5.2	Simulation von THOR-Netzen	90
5.2.1	Ereignisse	90
5.2.2	Schaltbeginn und Schaltende	91
5.2.3	Die Schaltregel	93
5.3	Schwierigkeiten	93
5.3.1	Maximale Schaltregel	93
5.3.2	C++-Objekte und Verzögerungszeit	95
6	Prototyp mit THOR-Netzen	97
6.1	Der Klassenentwurf	98
6.2	Die Umsetzung in THOR-Netze	100
6.2.1	Modellierung der Zeit	100
6.2.2	Die Verkehrsnetze	101

6.2.3	Die Mitnahmebeziehung	102
6.2.4	Quellen	104
6.2.5	Senken	105
6.3	Abweichungen vom formalen Simulationsmodell	105
6.4	Beschleunigung der Simulation	106
7	Simulationsergebnisse	107
7.1	Kapazitätsprobleme	108
7.2	Simulationsgeschwindigkeit	109
7.3	Auswertung der Bewegung von Personen	109
7.4	Simulative Bestwagsuche	111
7.5	Zusammenfassung der Ergebnisse	114
8	Ausblick	115
A	Selbstdefinierte Marken – Deklaration	116
B	Verkehrsgraphen und THOR-Netze	121
C	Bilder zur simulativen Bestwagsuche	127
D	Abkürzungen	130
	Literaturverzeichnis	131

Kapitel 1

Einleitung

1.1 Motivation

Unsere heutige Gesellschaft wird sehr stark durch den motorisierten Individualverkehr (MIV) geprägt. Leider ist dieser Verkehr mit einer enormen Belastung für die Umwelt – und damit letztlich für den Menschen – verbunden.

Private Kraftfahrzeuge werden mit wachsendem Aufwand an Energie und Ressourcen – größer, schwerer, leistungsfähiger – hergestellt, obwohl sie in der Regel weniger als eine Stunde täglich gefahren werden und nur zu etwa 25 % ausgelastet sind.

Verkehrswege und -räume, wie z.B. Parkieranlagen, werden für die privaten Kraftfahrzeuge bereitgestellt, die den Lebensraum der Menschen mehr und mehr einengen und zerschneiden.

Bei der Verbrennung von Kraftstoffen entstehen Abgase, die Krebs, Haut- und Lungenkrankheiten sowie Klimaveränderungen hervorrufen.

Weitere negative Auswirkungen bestehen in Unfallgefahren und Lärmbelästigungen mit z.T. schweren seelischen Folgen.

Es scheint so, als ob die Menschen, die über einen privaten PKW verfügen, ihre sogenannte „große Mobilität“ auf dem Rücken der Allgemeinheit austragen. Daher müssen die negativen Auswirkungen des MIV verringert und beseitigt werden. Die Mobilität¹ muß dazu nicht notwendigerweise verringert werden.

Die meisten dieser Aussagen sind den Büchern von Frederic Vester ([Ves90], [Ves95]) entnommen. Einige Aussagen stammen aus der Vortragsreihe „Stadt und Mobilität“, die an der Universität Stuttgart im Wintersemester 1995/96 von den Fakultäten für Bauingenieurwesen und Architektur gemeinsam organisiert wurde.

Zur Reduktion des MIV's scheinen folgende Strategien geeignet zu sein:

1. Verbesserung des öffentlichen Verkehrs (ÖV) mit dem Ziel, Verkehrsmengen vom MIV abzuziehen.

¹Der Autor dieser Arbeit versteht Mobilität als die subjektiv empfundene Beweglichkeit. Gleichbleibende oder wachsende Mobilität muß daher nicht an private Kraftfahrzeuge oder wachsende Kilometerleistung gekoppelt sein.

2. Vergrößerung der Auslastung aller Verkehrsmittel durch Bildung von Mitfahrgemeinschaften (MFGs) und Carsharing-Organisationen bei privaten Kraftfahrzeugen sowie durch eine bedarfsgerechte Planung des öffentlichen Verkehrs.
3. Gezielte Reduzierung der Verkehrsräume des MIV, um die Attraktivität des ÖV und des nichtmotorisierten Individualverkehrs gegenüber dem MIV zu erhöhen.
Beispiel: Die Schaffung von autofreien Wohnsiedlungen erhöht die Qualität des Lebensraumes dort und reduziert außerdem die Benutzung vom MIV für Kurzstrecken, da der Laden um die Ecke weniger entfernt ist als das private Kraftfahrzeug.
4. Ausrichtung der räumlichen Entwicklung und der Raumordnungspolitik auf die Verkürzung mittlerer Weglängen und -zeiten.
5. Umlegung der tatsächlichen Folgekosten des MIV nur auf die Benutzer des MIV. Dabei sollte auch die Beeinträchtigung der Lebensqualität durch den MIV berücksichtigt werden.
6. Optimierung des Güterverkehrs, insbesondere auch beim privaten Einkauf.
Beispiel: Statt den Großeinkauf mit dem Auto zu erledigen, kann der Einkauf auch durch persönliche oder schriftliche Bestellung in Auftrag gegeben werden. Ein Lieferservice übernimmt die Zustellung.

Das Erreichen dieser Teilziele erfordert in erster Linie gesellschaftliche Akzeptanz und Förderung. Viele planerische Entscheidungen sind allerdings durch die Verkehrsplanung von gestern und deren Modelle schon vorzementiert.

Obwohl der Begriff „Verkehrsplanung“ in der Literatur zumeist den Anschein erweckt, eine von objektiven Kriterien geleitete Aufgabenstellung zu sein, wird Verkehrsplanung oft in erster Linie von wirtschaftlichen Erwägungen gesteuert. Der Wohn- und Freizeitwert des Raumes, der von Verkehrswegen durchschnitten wird, steht meist nicht zur Debatte. Diese Problematik kann genauer bei Rudolf Menke [Men75] nachgelesen werden, der sich schon in den 70er Jahren mit einer alternativen Stadtverkehrsplanung auseinandergesetzt hat.

Im Jahre 1933 wurde in einer internationalen Konferenz von Architekten und Städtebauern die räumliche Trennung von Wohnen, Arbeit/Schule, Versorgung und Freizeit als erstrebenswertes Ziel formuliert. Das Abschlußpapier dieser Konferenz ist als „Charta von Athen“ ([Hil84]) bekannt geworden und hat nachhaltig auf die Trennung der Lebensbereiche in der Raumordnungspolitik gewirkt. Die Trennung der Lebensbereiche und die starke Zunahme des MIV haben sich in den folgenden Jahrzehnten wechselseitig bedingt.

Eine moderne Verkehrsplanung, die sich von althergebrachten Zwängen lösen will, muß auch ihre Modellvorstellungen vom Verkehr den neuen Zielsetzungen anpassen. Im Zeitalter der Informationsgesellschaft werden Modelle von Rechnern verarbeitet und müssen dazu zunächst formalisiert werden. Diese Formalisierung hat neben den positiven klärenden Auswirkungen auch den ungewollten

Nebeneffekt, einmal geschaffene Modelle langfristig festzulegen. Nur im Bewußtsein dieses Zusammenhangs können formale Modelle spezifiziert und verwendet werden. Bei ihrer Spezifikation ist größte Umsicht geboten, um möglichst viele relevante Einfluß- und Wirkfaktoren des zu modellierenden Systems zu berücksichtigen. Es liegt im Charakter eines Modells, daß niemals alle Faktoren berücksichtigt werden können.

Mit dieser Arbeit möchte ich einen kleinen Beitrag dazu leisten, die Probleme im Bereich Verkehr durch sinnvolle Planung beherrschbar zu machen. Die Entwicklung eines realitätsnahen Simulationsmodells ist ein wichtiges Hilfsmittel bei der Analyse von Verkehrsproblemen und bei der Bewertung verkehrsplanerischer Maßnahmen.

1.2 Verkehr – Was ist das?

In diesem Abschnitt sollen die Begriffe Verkehr, Verkehrsmodellierung und Verkehrssimulation geklärt werden, um die Aufgabenstellung im nächsten Abschnitt klarer fassen zu können.

In [Hen76] wird **Verkehr** wie folgt definiert:

„**Verkehr** ist die Gesamtheit der Vorgänge, die der Raumüberwindung dienen: Ortsveränderungen von Personen, Gütern, Nachrichten. In Analysen wird Verkehr i.a. definiert als Ortsveränderung bezüglich einer Verkehrszelleneinteilung (...). Die erkennbaren Verkehrsvorgänge (...) sind daher eine Teilmenge des Gesamtverkehrs.“

Der Begriff „Verkehr“ umfaßt nicht nur den Straßenverkehr, sondern auch den Transport von Gas, Öl, Wasser, Energie und Informationen mit Hilfe von Rohren, Kanälen, (licht-) elektrischen Leitungen und elektromagnetischen Wellen. Es gibt Dinge, die sich von selbst bewegen, z.B. Personen, Tiere, Fahrer-KFZ-Einheiten, usw., und andere, die bewegt werden müssen, wie z.B. Güter, Flüssigkeiten in Rohren, Informationen in elektrischer Form, usw..

Auch in Fällen, bei denen dies nicht offensichtlich ist, greift eine Vielzahl verschiedener Arten von Verkehr ineinander. Wenn z.B. ein Autofahrer über das Radio über einen bevorstehenden Stau informiert wird, gelangt die Information in Form von Radiowellen zum Verkehrsteilnehmer und beeinflusst dessen Verkehrsverhalten.

Ein **Verkehrsmodell** ist demnach ein Modell² für die Vorgänge, die der Raumüberwindung dienen. Von einem **Simulationsmodell** spricht man, wenn die zeitliche Reihenfolge der realen Vorgänge eine Entsprechung im Modell findet. Eine **Simulation** berechnet diese Vorgänge oder die dazwischenliegenden Zustände in der richtigen zeitlichen Reihenfolge, meist aufgrund von Ursache-Wirkungs-Zusammenhängen. Verschiedene Arten der Simulation sind in Fishwick [Fis95] dargestellt.

²s.a. Stachowiak [Sta73] *Allgemeine Modelltheorie*.

Eine **Verkehrssimulation** erfordert ein Verkehrsmodell, das sowohl die bewegten Objekte wie Personen, Güter oder Informationen als auch das dynamische Verhalten dieser Objekte abbildet. Die bewegten Objekte heißen im weiteren Verlauf dieser Arbeit **Verkehrsobjekte**. Die Analyse der Simulationsergebnisse dient zunächst hauptsächlich der Modellvalidierung, bei der geprüft wird, ob die am Modell beobachtbaren Ausgaben mit den realen Daten übereinstimmen. Erst wenn das Modell validiert ist, können die Analysedaten für Aussagen über den abgebildeten Realitätsausschnitt verwendet werden.

Der erste und zugleich schwierigste Schritt der Modellbildung ist die Wahl der Modellgrenze. Es muß entschieden werden, welcher Realitätsausschnitt mit welcher Genauigkeit abgebildet werden soll und welche Aussagen vom Modell erwartet werden. Dabei ist bereits die Beschränkung auf „Verkehr“ eine scharfe Grenze, die andere wichtige Zusammenhänge ausblendet. Man sollte sich bei jedem Abgrenzungsschritt darüber im klaren sein, daß man viele bekannte und noch mehr unbekanntere Einfluß- und Wirkfaktoren ohne nähere Untersuchung ihrer Wichtigkeit beiseite läßt. In seinen Büchern *Ausfahrt Zukunft* [Ves90] und *Crashtest Mobilität, Fakten – Strategien – Lösungen* [Ves95] weist Frederic Vester nachhaltig auf die Wichtigkeit vernetzten Systemdenkens gerade im Bereich Verkehr hin.

1.3 Aufgabenstellung

In dieser Arbeit soll das bestehende Verkehrsmodell am Institut für Eisenbahn- und Verkehrswesen (IEUV) formalisiert werden. Ausgehend von diesem Modell soll dann ein Simulationsmodell für den Verkehr entwickelt werden, das

1. die Bewegung einzelner Verkehrsobjekte simuliert,
2. festgelegte reale Abläufe ausreichend genau widerspiegelt und
3. eine variable Beschreibungsgenauigkeit besitzt.

Die zweite Anforderung bezieht sich dabei auf die Feinkörnigkeit der Raumstruktur. Es soll möglich sein, vorhandene Beschreibungen von Verkehrsinfrastruktur (z.B. Straßennetz) und Bewegungsabläufen (z.B. Fahrplan eines Busses) einfach in das Modell zu übertragen, ohne zuerst die Beschreibungsgenauigkeit aufwendig anpassen zu müssen.

In einem weiteren Teil soll das Simulationsmodell prototypisch mit Hilfe von THOR-Netzen – einer Art höherer Petri-Netze mit C++-Objekten als Marken – implementiert werden. Dabei ist die prinzipielle Verwendbarkeit von THOR-Netzen im Hinblick auf größere Anwendungen im Bereich Verkehrssimulation zu untersuchen.

ergeben haben. Dabei wird auch gezeigt, wie leicht ein vorhandenes Simulationsmodell für den ÖPNV dazu genutzt werden kann, die Bestwegsuche, also die Berechnung bester Fahrtrouten und minimaler Kosten (meist Fahrtdauer) zwischen je zwei Punkten eines gegebenen Verkehrsnetzes, simulativ durchzuführen.

Die Kapitel 3, 4, 6 und 7 stellen also die Entwicklung, Umsetzung und Implementierung sowie die ersten Erfahrungen mit einem neuen Modell zur Simulation von Verkehr dar. Kapitel 2 beschreibt die Datengrundlage für die konkrete Modellierung des U- und S-Bahn-Netzes von Stuttgart. In Kapitel 5 wird die prototypische Realisierung des Simulationsmodells mit THOR-Netzen erklärt. Sowohl die Datengrundlage als auch die verwendete Implementierung sind austauschbar.

Abbildung 1.1 soll den Leser bei der Auswahl der Kapitel unterstützen. Die dargestellten Abhängigkeiten stellen dabei keine vorgegebene Lesereihenfolge dar, können jedoch den Einstieg erleichtern. Ich wünsche viel Vergnügen beim Lesen und habe die Hoffnung, daß diese Arbeit zu weiterer Forschung anregt.

Kapitel 2

Prognose der Verkehrsnachfrage

Eine der wichtigsten Fragestellungen der Verkehrsplanung ist die Prognose zukünftiger Entwicklungen der Verkehrsnachfrage. Die Verkehrsnachfrage beschreibt dabei die Menge aller Fahrten von Personen und kann klassifiziert werden durch

- Start und Ziel,
- Fahrtzweck,
- Abfahrtszeit,
- Ankunftszeit,
- benutzte Verkehrsmittel,
- Fahrtroute,
- zeitl. und räuml. Verlauf der Fahrt

sowie durch weitere Parameter. Bei einer Verkehrsprognose wird üblicherweise ein Untersuchungszeitraum (z.B. die Hauptverkehrszeit) festgelegt, für den dann die Verkehrsnachfrage als Zahl aller Fahrten je Start-Ziel-Kombination, Verkehrsmittel und Fahrtroute ausgedrückt wird.

Solche Untersuchungen werden z.B. unter folgenden Gesichtspunkten durchgeführt:

- Wirtschaftlichkeitsuntersuchungen von neuen ÖPNV-Linien oder geänderter Bedienungshäufigkeit.
- Bewertung von Maßnahmen zur Verbesserung des ÖPNV auf der Grundlage der „Standardisierten Bewertung von Verkehrsweginvestitionen des Öffentlichen Personennahverkehrs“. ([HMD⁺])
- Beurteilung von verkehrsberuhigenden Maßnahmen.

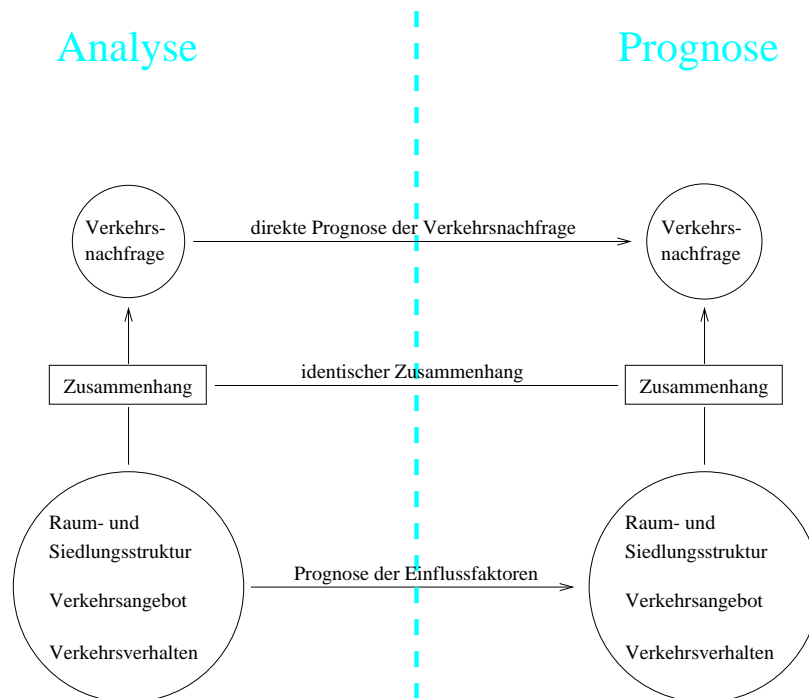


Abbildung 2.1: direkte und indirekte Prognose

Die Verkehrsnachfrage hängt nach [Wer94] ab von

- der **Raum- und Siedlungsstruktur**, also der Verteilung der Wohnbevölkerung und der Arbeits-, Ausbildungs- und Freizeiteinrichtungen,
- dem **Verkehrsangebot**, also der Verkehrsinfrastruktur und den Verkehrsmitteln mit ihren betrieblichen Eigenschaften und
- den **verkehrsrelevanten Verhaltensweisen** der Bevölkerung.

Statt die Verkehrsnachfrage direkt aus einer früheren Verkehrsnachfrage vorherzusagen, also eine Trendprognose durchzuführen, kann man auch einen Zusammenhang zwischen den genannten Einflußfaktoren und der Verkehrsnachfrage modellieren, die Veränderung der Einflußfaktoren prognostizieren und die Verkehrsnachfrage nach der Veränderung mit Hilfe des modellierten Zusammenhangs berechnen (Abb. 2.1).

Die Trendprognose wird in der Abbildung durch den direkten Pfeil von der Verkehrsnachfrage aus der Analyse zur prognostizierten Verkehrsnachfrage dargestellt. Die indirekte Prognose verwendet eine Prognose der Einflußfaktoren und kann daher auch die Ursachen der Verkehrsentstehung mitberücksichtigen.

Das Vierstufenmodell ist dabei ein Grobraster für den Zusammenhang zwischen Einflußfaktoren und Verkehrsnachfrage, wobei die Verkehrsnachfrage nach Start, Ziel, Verkehrsmittel und Fahrtroute klassifiziert wird.

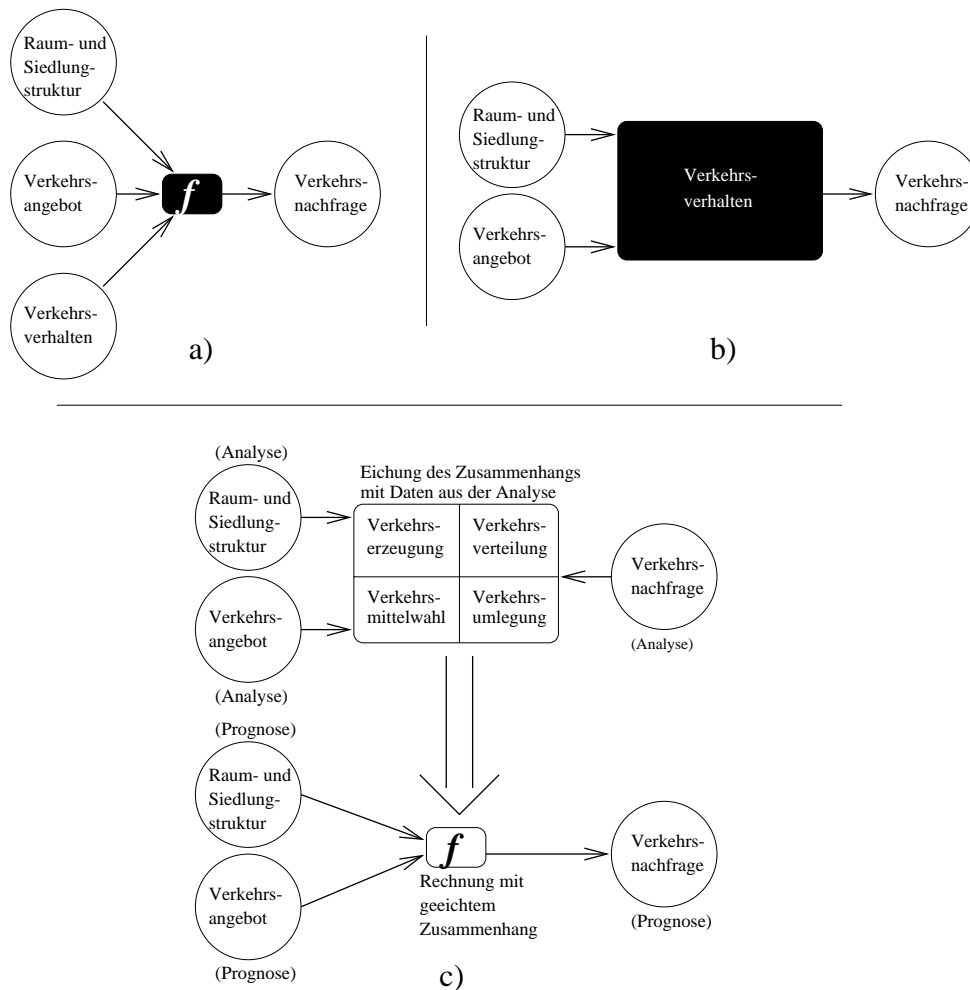


Abbildung 2.2: a) Ursachen der Verkehrsnachfrage (Wermuth), b) Verkehrsverhalten als „Black Box“, c) Modellierung des Verkehrsverhaltens aus analysierter Verkehrsnachfrage

2.1 Einführung ins Vierstufenmodell

Die Vorstellung, daß sich aus Raum- und Siedlungsstruktur, Verkehrsangebot und Verkehrsverhalten die Verkehrsnachfrage direkt ergibt (Abbildung 2.2 a)), scheidet daran, daß sich das verkehrsrelevante Verhalten der Menschen weder beschreiben noch exakt ermitteln läßt.

Deshalb geht das Verkehrsverhalten in den Zusammenhang mit ein. Wie in Abbildung 2.2 b) veranschaulicht, entsteht dabei eine „Black Box“ unbekannter Zusammenhänge zwischen Raum- und Siedlungsstruktur sowie Verkehrsangebot als Eingangsgrößen und der daraus resultierenden Verkehrsnachfrage als Zielgröße.

Mit bekannter Verkehrsnachfrage (sowie bekannter Siedlungsstruktur und bekanntem Verkehrsangebot) aus der Analyse stellt man ein Modell für das

Verkehrsverhalten her. Diesen Modellzusammenhang überträgt man für eine geänderte Raum- und Siedlungsstruktur oder ein geändertes Verkehrsangebot auf den Prognosefall (Abbildung 2.2 c)). Dabei wird natürlich unterstellt, daß der Zusammenhang korrekt modelliert wurde und daß sich das Verkehrsverhalten nicht ändert.

Die Bezeichnung „Vierstufenmodell“ kommt daher, daß dieser Zusammenhang nun in vier Teilmodelle gegliedert wird:

Verkehrserzeugung: Die Erzeugung bestimmt die Verkehrsmenge als Anzahl angetretener bzw. beendeter Fahrten und modelliert somit die Entscheidung zur Fahrt.

Verkehrsverteilung: Die Verteilung ordnet jeder angetretenen eine beendete Fahrt zu und modelliert somit die Entscheidung zum Ziel.

Verkehrsmittelwahl: Die Mittelwahl ordnet jeder Fahrt ein Verkehrsmittel oder eine Kombination von Verkehrsmitteln zu und modelliert somit die Wahl des Verkehrsmittels.

Verkehrsumlegung: Die Umlegung ordnet jeder Fahrt einen genauen Weg zu und modelliert somit die Wegwahl.

Durch diese Gliederung lassen sich Einflußgrößen finden, die nur für einige Teilmodelle relevant sind. Die Relevanz wird dabei vom Modellierer aus seiner Erfahrung mit der Prognose von Verkehrsnachfragen beurteilt. So sind Einwohner- und Beschäftigtenzahl eines Bezirkes relevant für die Verkehrserzeugung, aber nicht für die Verkehrsmittelwahl oder die Verkehrsumlegung.

Im nächsten Abschnitt wird nun ein Fallbeispiel für eine Prognose eingeführt, an dem dann Schritt für Schritt die Eingangs- und Zielgrößen sowie die Durchführung des Vierstufenalgorithmus und der Verkehrsnachfrageprognose erklärt werden.

Die formale Beschreibung beschränkt sich dabei auf das Modell für das Verkehrsangebot, da die anderen Teile der Theorie zu Vierstufenalgorithmus und Verkehrsnachfrageprognose im weiteren Verlauf dieser Arbeit nicht mehr benötigt werden. Weitere Informationen zum Vierstufenalgorithmus finden sich in Steierwald [Ste94], Mäcke [Mäc64] und dem Vorlesungsskript zur „Verkehrsprognose“ des IEUV [DSH95].

2.2 Beispiel einer Planung

Als fiktives Beispiel wähle ich die Umgebung eines kleinen Orts namens „Netzheim“ (Abbildung 2.3). Neben einer bestehenden Buslinie, die „Netzstadt“ mit „Fernheim“ verbindet, soll eine zweite Buslinie eingeführt werden, die von „Nahheim“ über „Netzheim“ nach „Beiheim“ führt.

Dadurch wird „Nahheim“ an den öffentlichen Verkehr angebunden. Es entsteht eine Umsteigemöglichkeit nach „Netzstadt“ sowie eine direkte Busverbindung zum Bahnhof in „Beiheim“.

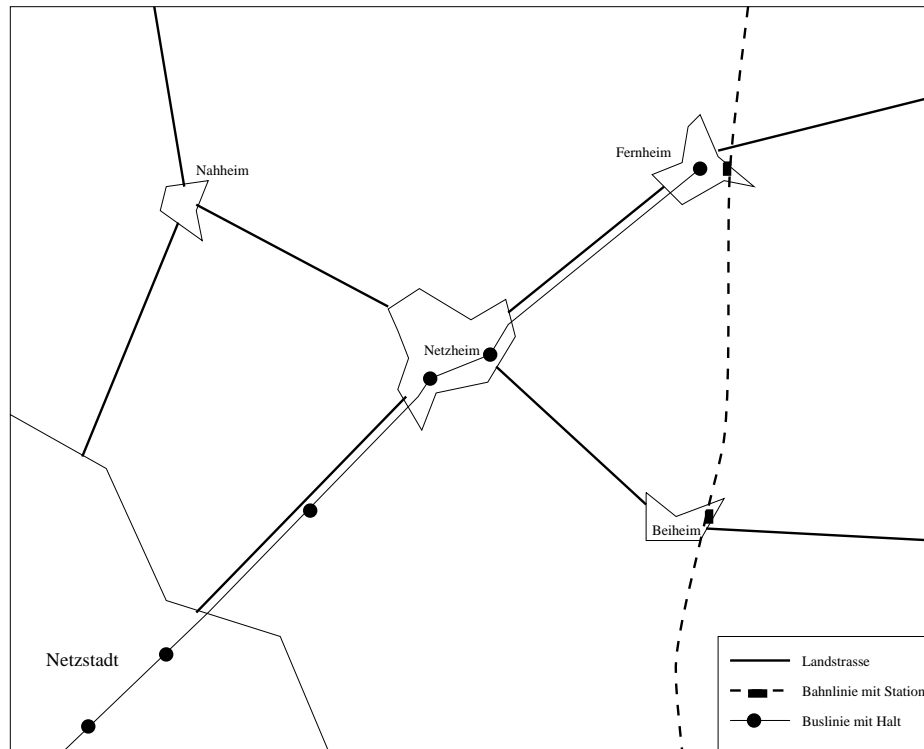


Abbildung 2.3: Die Umgebung von Netzheim

2.3 Zielgröße und Eingangsgrößen

Beim Vierstufenalgorithmus soll die Verkehrsmenge differenziert nach Start, Ziel, Verkehrsmittel und Route berechnet werden. Daher soll

- die Raum- und Siedlungsstruktur (zur Beschreibung von Start und Ziel) sowie
- das Verkehrsangebot (zur Beschreibung von Verkehrsmittel und Fahrtroute)

als **Eingangsgrößen** und die Verkehrsnachfrage als **Zielgröße** in den folgenden Abschnitten formal gefaßt werden.

Verwendete Notationen:

1. Namen werden immer als Element der universalen Namenmenge $N = \{n \mid n \text{ ist Name}\}$ dargestellt.
2. Tupel werden rund $(())$, Listen eckig $([])$ geklammert.
3. Auf Elemente eines Tupels wird zuweilen mit der Punktschreibweise zugegriffen, z.B. besteht $l = (R, time, dist)$ aus $l.R$, $l.time$ und $l.dist$.

2.3.1 Raum- und Siedlungsstruktur

Um Start und Ziel für die Verkehrsnachfrage benennen zu können, braucht man eine Unterteilung des Raums, in dem die Untersuchung durchgeführt wird. Die Einteilung erfolgt in sogenannte Verkehrszellen und ist für das Beispiel „Netzheim“ in Abbildung 2.4 dargestellt.

Definition 2.1

Eine **Verkehrszelle** ist eine nicht-leere Teilmenge von $\mathbb{R} \times \mathbb{R}$.

Der Auftraggeber der Verkehrsprognose bestimmt die Ausdehnung von Planungs- und Untersuchungsraum. Der Planungsraum ist dabei der Raum, in dem eine nennenswerte Änderung der Verkehrsnachfrage aufgrund der Verkehrsmaßnahme erwartet wird. Der Untersuchungsraum ist so groß, daß für jeden nennenswerten Verkehr im Planungsraum sowohl Start als auch Ziel im Untersuchungsraum liegen. Der Untersuchungsraum umfaßt den Planungsraum.

Die Einteilung von Planungs- und Untersuchungsraum in Verkehrszellen erfolgt ebenfalls aufgrund der Planung. Die Ausdehnung der Zellen orientiert sich normalerweise an der Bezirkseinteilung des statistischen Landesamtes, da dann auch Einwohner- und Beschäftigtenzahlen direkt übernommen werden können. In Räumen, die von der zu untersuchenden Verkehrsmaßnahme (im Beispiel die Buslinie) ausreichend weit entfernt sind, werden mehrere statistische Bezirke zu einer Verkehrszelle zusammengefaßt. In Räumen, die unmittelbar von der Maßnahme betroffen sind, werden die statistischen Bezirke weiter unterteilt.

Definition 2.2

Eine endliche Menge von Verkehrszellen $P = \{cell_1, \dots, cell_{p_max}\}$ heißt **Planungsraum**, falls $\forall_{1 \leq i, j \leq p_max, i \neq j} : cell_i \cap cell_j = \emptyset$.

Eine endliche Menge von Verkehrszellen $U = \{cell_1, \dots, cell_{u_max}\}$ heißt **Untersuchungsraum**, falls $\forall_{1 \leq i, j \leq u_max, i \neq j} : cell_i \cap cell_j = \emptyset$.

Eine Siedlungsstruktur ist nun ein Untersuchungsraum, der einen Planungsraum umfaßt und dem statistische Größen zugeordnet sind. Die Siedlungsstruktur ist Eingangsgröße der Verkehrserzeugung.

Definition 2.3

Eine **Siedlungsstruktur** $Struc = (P, U, name, pop, empl)$ besteht aus

- einem Planungsraum P .
- einem Untersuchungsraum U mit $P \subseteq U$.
- einer Abbildung $name : U \rightarrow N$, die jeder Verkehrszelle des Untersuchungsraums einen Namen zuordnet.
- einer Abbildung $pop : P \rightarrow \mathbb{N}$, die jeder Verkehrszelle des Planungsraums die Einwohnerzahl zuordnet.

- einer Abbildung $empl : P \rightarrow \mathbb{N}$, die jeder Verkehrszelle des Planungsraums die Beschäftigtenzahl zuordnet.

Bemerkung: Statt Einwohnerzahl und Beschäftigtenzahl können selbstverständlich auch andere statistische Größen zur Beschreibung der Verkehrszellen herangezogen werden.

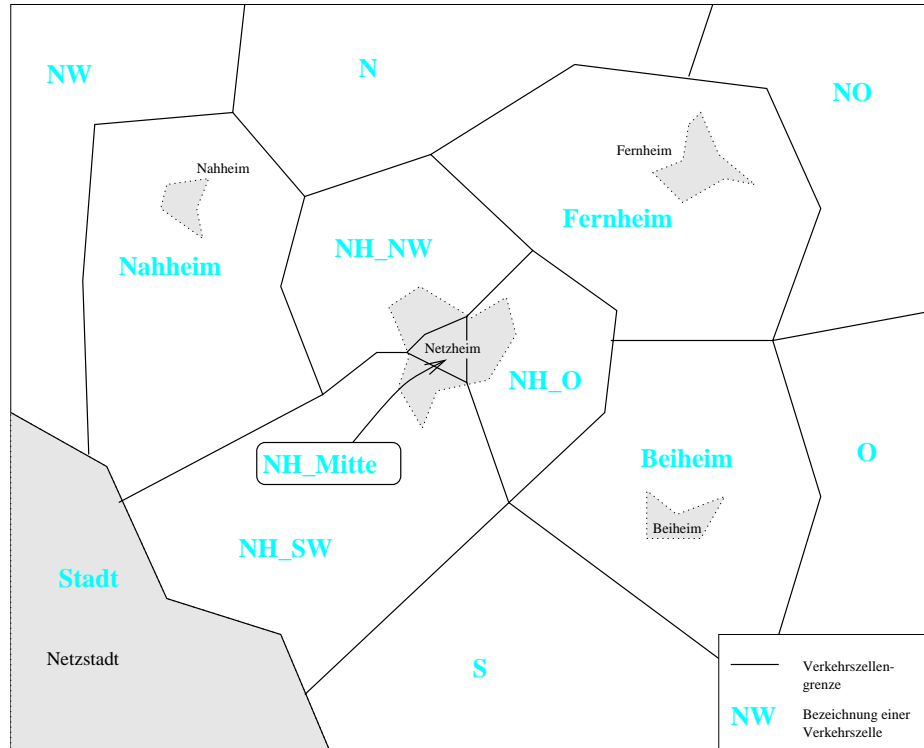


Abbildung 2.4: Auswahl der Verkehrszellen

Beispiel: In Abbildung 2.4 wurde der Raum um Netzheim herum mehr oder weniger willkürlich eingeteilt. Die Verkehrszellen von Netzheim (**NH_xxx**) stellen zusammen mit **Nahheim**, **Fernheim** und **Beiheim** den Planungsraum dar. Der Untersuchungsraum enthält auch noch die anderen Verkehrszellen. Netzheim selbst wurde dabei in 4 Verkehrszellen unterteilt. Die Siedlungsstruktur des Beispiels sieht also so aus:

- $Struc = (P, U, name, pop, empl)$.
- $P = \{NH_Mitte, NH_NW, NH_SW, NH_O, Nahheim, Fernheim, Beiheim\}$.
- $U = P \cup \{Stadt, NW, N, NO, O, S\}$.
- Die Verkehrszellen wurden aus Darstellungsgründen direkt mit ihrem Namen identifiziert. Daher sind nur pop und $empl$ in Tabelle 2.1 dargestellt.

$cell \in P$	$pop(cell)$	$empl(cell)$
NH_Mitte	1000	200
NH_NW	1000	300
NH_O	1000	200
NH_SW	1000	100
Nahheim	700	600
Fernheim	1000	200
Beiheim	800	200

Tabelle 2.1: Die statistischen Größen für das Beispiel „Netzheim“

Die Verkehrszelle „Nahheim“ hat also z.B. 700 Einwohner und 600 Beschäftigte. Die hohe Beschäftigtenzahl liegt daran, daß Nahheim einen großen Industriekomplex besitzt. Natürlich pendeln die meisten Beschäftigten von „Nahheim“ jeden Tag von „Netzheim“ oder „Netzstadt“.

2.3.2 Verkehrsangebot

Das Verkehrsangebot wird dargestellt durch ein Staßennetz für den Individualverkehr (IV) und ein Liniennetz für den öffentlichen Verkehr (ÖV). Diese Aufteilung erfolgt in Anlehnung an die Vorgehensweise des IEUV.

Diese Netze dienen dazu, die Kosten für eine bestimmte Verbindung zwischen zwei Verkehrszellen zu berechnen. Dazu gibt es Anschlüsse von jeder Verkehrszelle an das Straßen- und das Liniennetz. So kann auch der sogenannte Bestweg, also der Weg mit den minimalen Kosten zwischen zwei Verkehrszellen, ermittelt werden. Die Aufteilung in Staßennetz für den IV und Liniennetz für den ÖV ergibt sich aus der Notwendigkeit, getrennte Kostenrechnungen für beide Verkehrsmittel durchzuführen.

Im Vierstufenmodell werden die Ergebnisse dann wie folgt verwendet:

- die Kosten für den besten Weg im IV und ÖV zwischen je zwei Verkehrszellen dienen als Eingangsgröße für die Verkehrsverteilung.
- die Kosten für den besten Weg, getrennt nach IV und ÖV, dienen als Eingangsgröße für die Verkehrsmittelwahl.
- der Bestweg (auch die besten Wege) ist Grundlage der Verkehrsumlegung.

In den Abschnitten 2.3.2.1 und 2.3.2.3 werden die Modelle für die Kosten- und Bestwegberechnung für den IV und den ÖV formal dargestellt. Natürlich sind auch andere Aufteilungen denkbar. Am IEUV wurde das Modell z.B. so erweitert, daß die Bestwegsuche auch für den P+R-Verkehr durchgeführt werden kann.

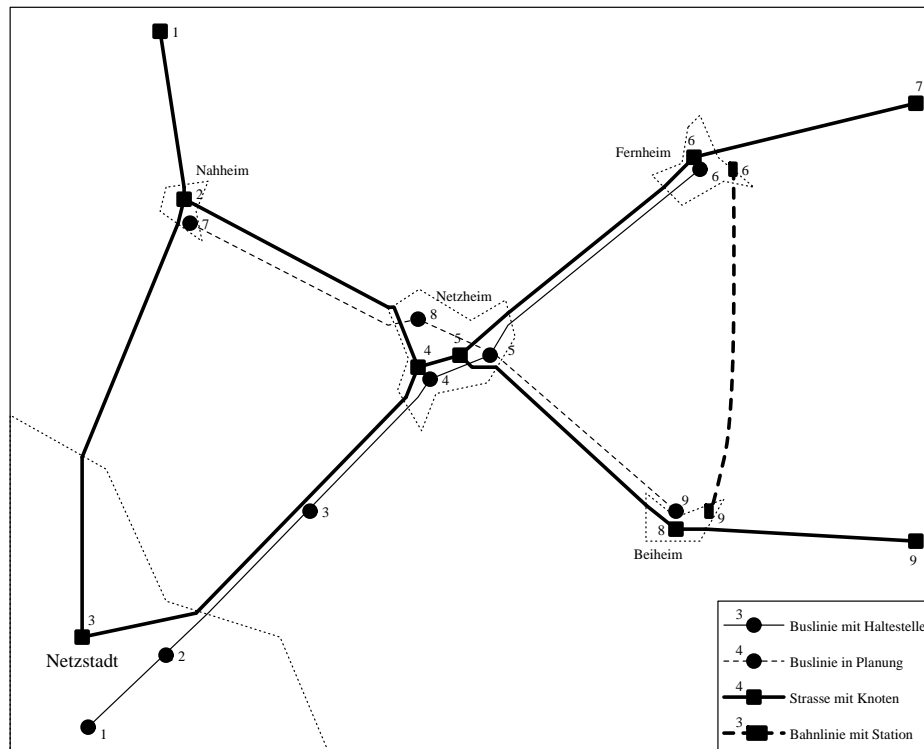


Abbildung 2.5: Das Verkehrsangebot in der Umgebung von Netzheim

2.3.2.1 Modell für das Straßenverkehrsnetz

Ein Verkehrsnetz für den IV besteht aus einem gerichteten Verbindungsgraph für das Straßennetz und den Anschlüssen an die Verkehrszellen eines Untersuchungsraums. Außerdem sind die Verbindungen und die Anschlüsse noch mit der jeweiligen Fahrtdauer beschriftet.

Definition 2.4

Ein **Straßen-Netz** $IVNet = (V, E, IVCon, time)$ zu einem Untersuchungsraum U besteht aus

- einer endlichen Menge von **Straßenverkehrsknoten** V ,
- einer Menge von gerichteten **Verbindungen** zwischen den Straßenverkehrsknoten $E \subseteq V \times V$,
- einer Menge von ungerichteten **Anbindungen** der Verkehrszellen an Straßenverkehrsknoten $IVCon \subset U \times V$. Dabei wird jede Verkehrszelle angebunden:

$$\forall_{cell \in U} \exists_{cross \in V} : (cell, cross) \in IVCon,$$
- einer Abbildung $time : (E \cup IVCon) \rightarrow \mathbf{N}$, die den Verbindungen und den Anbindungen eine **Fahrtdauer** in Sekunden zuordnet.

Bemerkung: Die ausschließliche Beschriftung der Verbindungen und Anbindungen mit der Fahrdauer ist eine starke Vereinfachung. Am Institut für Eisenbahn- und Verkehrswesen werden die Verbindungen mit Länge, Geschwindigkeit und einem Stauzuschlag für die Hauptverkehrszeit beschriftet. Außerdem wird angegeben, ob es sich um eine innerörtliche Strecke handelt. So ist es möglich, die Fahrdauer anhand des Planungszeitraums (je nach Verkehrszeit mit oder ohne Stauzuschlag) zu bestimmen.

Beispiel 2.1 : Das Verkehrsangebot um „Netzheim“ ist in Abbildung 2.5 ohne die Beschriftung und die Anbindungen dargestellt.

Das Straßen-Netz $IVNet = (V, E, IVCon, time)$ hat die Straßenverkehrsknoten $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Die Verbindungsmenge E wird ebenso wie die Anbindungsmenge $IVCon$ gemeinsam mit der Fahrdauer $time$ in Tabelle 2.2 dargestellt.

$e \in E$	$time(e)/60$ in Minuten
(1,2)	3
(2,1)	3
(2,3)	10
(3,2)	10
(2,4)	8
(4,2)	8
(3,4)	9
(4,3)	9
(4,5)	2
(5,4)	2
(5,6)	4
(6,5)	4
(5,8)	4
(8,5)	4
(6,7)	3
(7,6)	3
(8,9)	4
(9,8)	4

$ivcon \in IVCon$	$time(ivcon)$ in Minuten
(NH_Mitte,5)	1
(NH_NW,4)	2
(NH_O,5)	1
(NH_SW,4)	1
(Nahheim,2)	1
(Fernheim,6)	2
(Beiheim,8)	1
(Stadt,3)	5

Tabelle 2.2: Fahrdauer auf den Verbindungen (li.) und Anbindungen (re.) für das Beispiel „Netzheim“

Durch E werden je zwei Straßenverkehrsknoten $cross \in V$ miteinander verbunden. Die Straßenverkehrsknoten 6 und 5 sind durch ein Straßenstück miteinander verbunden. Die durchschnittliche Fahrdauer beträgt vier Minuten.

Durch $IVCon$ wird jeder Verkehrszelle $cell \in U$ ein Straßenverkehrsknoten $cross \in V$ zugeordnet (Anbindung). Die Verkehrszelle *Fernheim* ist an den Straßenverkehrsknoten 6 angebunden. Die durchschnittliche Anfahrt dauert dabei zwei Minuten. Die Anbindungen von Verkehrszellen, die nicht besiedelt sind (NW, N, NO, O, S) wurden hier nicht dargestellt, obwohl sie gemäß Definition natürlich existieren.

Es wurden nicht alle Straßen und Straßenverkehrsknoten innerhalb des Planungsraumes dargestellt. Abhängig vom Planungsgegenstand wird entschieden,

welche Straßen und Straßenverkehrsknoten für die Planung relevant sind und deshalb dargestellt werden müssen. Im Beispiel wurden in Netzheim nur die Durchgangsstraßen modelliert. Außerdem wurden alle Strecken, die aus dem Untersuchungsraum herausführen, nicht berücksichtigt.

2.3.2.2 Abstandsmodell im IV

Nun sollen mit Hilfe dieses Straßenverkehrsnetzes die Kosten und besten Wege zwischen je zwei Verkehrszellen berechnet werden. Das motiviert zu

Definition 2.5

Zu einem Untersuchungsraum U und einem Straßen-Netz $IVNet = (V, E, IVCon, time)$ wird definiert:

- Eine Liste $r_{iv} = [cross_1, \dots, cross_{r_max}]$ mit $\forall_{1 \leq i \leq r_max} : cross_i \in V$ heißt **IV-Route** $\Leftrightarrow \forall_{1 \leq i \leq r_max} : (cross_i, cross_{i+1}) \in E$.
(Eine IV-Route ist ein Folge von Verbindungen)
- Eine IV-Route $r_{iv} = [cross_1, \dots, cross_{r_max}]$ heißt **IV-Verbindung** von $cell_s$ nach $cell_z$, falls gilt
 1. $(cell_s, cross_1) \in IVCon$,
(Die Start-Verkehrszelle ist an den Startknoten der Route angebunden)
 2. $(cell_z, cross_{r_max}) \in IVCon$,
(Die Ziel-Verkehrszelle ist an den Zielknoten der Route angebunden)
 3. $cell_s \neq cell_z$.
(IV-Routen innerhalb von Verkehrszellen werden nicht berücksichtigt)
- Die **Menge aller IV-Verbindungen von Verkehrszelle s nach Verkehrszelle z** ist definiert als
 $R_{iv}(s, z) = \{r \mid r \text{ ist IV-Verbindung von } s \text{ nach } z\}$.
- Die **Kosten** einer IV-Verbindung $r = [cross_1, \dots, cross_{r_max}] \in R_{iv}(s, z)$ ergeben sich zu $Cost_{iv}(s, z, r) = time(s, cross_1) + \sum_{1 \leq j < r_max} time(cross_j, cross_{j+1}) + time(z, cross_{r_max})$.
- Der **IV-Abstand** zwischen zwei Verkehrszellen $s, z \in U$ ist definiert als
 $D_{iv}(s, z) = \text{Min} \{Cost_{iv}(s, z, r) \mid r \in R_{iv}(s, z)\}$.
- Eine IV-Verbindung $r \in R_{iv}(s, z)$ heißt **bester IV-Weg** von Verkehrszelle s nach Verkehrszelle $z \Leftrightarrow Cost_{iv}(s, z, r) = D_{iv}(s, z)$.

Bemerkung zur Definition: Die Kostenfunktion $Cost_{iv}$ stellt das Kostenmodell dar. Ich habe eine lineare Summierung der Einzelverbindungen gewählt, dies entspricht der Vorgehensweise am IEUV.

Beispiel 2.2 : In unserem Beispiel gibt es zwischen den Verkehrszellen „Nahheim“ und „Stadt“ (Netzstadt) folgende IV-Verbindungen:

1. $[2, 3] \in R_{iv}(\text{Nahheim}, \text{Stadt})$ wegen
 - $(\text{Nahheim}, 2) \in IVCon$,
 - $(\text{Stadt}, 3) \in IVCon$,
 - $(2, 3) \in E$.
2. $[2, 4, 3] \in R_{iv}(\text{Nahheim}, \text{Stadt})$ mit analoger Begründung.

Daher ist $R_{iv}(\text{Nahheim}, \text{Stadt}) = \{[2, 3], [2, 4, 3]\}$.

Für die IV-Verbindung $[2, 3]$ errechnen sich die Kosten wie folgt:

$$Cost_{iv}(\text{Nahheim}, \text{Stadt}, [2, 3]) = time(\text{Nahheim}, 2) + time(2, 3) + time(\text{Stadt}, 3)$$

$$Cost_{iv}(\text{Nahheim}, \text{Stadt}, [2, 3]) = 1 + 10 + 5 = 16$$

Analog für die IV-Verbindung $[2, 4, 3]$:

$$Cost_{iv}(\text{Nahheim}, \text{Stadt}, [2, 4, 3]) = 1 + 8 + 9 + 5 = 23$$

Die Kosten der IV-Verbindungen sind also 16 bzw. 23 (als Reisezeit in Minuten), der IV-Abstand zwischen „Nahheim“ und „Stadt“ ergibt sich zu

$$D_{iv}(\text{Nahheim}, \text{Stadt}) = 16$$

$[2, 3]$ ist bester IV-Weg von „Nahheim“ nach „Stadt“.

2.3.2.3 Modell für den öffentlichen Verkehr

Das Liniennetz für den ÖV besteht aus Haltestellen und Linien, die durch einen Linienvverlauf und Fahrtdauern gekennzeichnet sind. Ebenso wie beim Straßennetz gibt es außerdem Anschlüsse an die Verkehrszellen und Beschriftungen für Haltestellen, Linien und Anschlüsse.

Definition 2.6

Ein **Liniennetz** $LNet = (S, L, LCon, name, koord, art, takt, abstand)$ zu einem Untersuchungsraum U besteht aus

- einer endlichen Menge von **Haltestellen** S ,
- einer endlichen Menge von **Linien** L . Dabei besteht $l \in L$ mit $l = (R, time, dist)$ aus
 1. einem **Linienvverlauf** $R = [s_1, \dots, s_n]$ mit $\forall_{1 \leq i \leq n} : s_i \in S$,
 2. den **Fahrtdauern** $time : \{1, \dots, n - 1\} \rightarrow \mathbb{N}$ zwischen den Haltestellen,
 3. den **Streckenlängen** $dist : \{1, \dots, n - 1\} \rightarrow \mathbb{N}$,

($time$ und $dist$ bilden die Indices des Linienvverlaufs auf die entsprechenden Werte ab.)

- einer Menge von ungerichteten **ÖV-Anbindungen** der Verkehrszellen an Haltestellen $LCon \subset U \times S$. Dabei wird jede Verkehrszelle angebunden: $\forall_{cell \in U} \exists_{stop \in S} : (cell, stop) \in LCon$,
- den **Namen** der Haltestellen $name : S \rightarrow N$,
- den **Koordinaten** der Haltestellen $koord : S \rightarrow \mathbb{N} \times \mathbb{N}$,
- der **Verkehrsmittelart** je Linie $art : L \rightarrow \{\text{S-Bahn, U-Bahn, Straßenbahn, Zug, Bus}\}$,
- der **Taktzeit** je Linie $takt : L \rightarrow \mathbb{N}$ in Minuten und
- der **Anbindungslänge** $abstand : LCon \rightarrow \mathbb{N}$ in Metern.

Bemerkungen zur Definition:

1. Da Linien dieselbe Haltestelle mehrmals anfahren können, ist es nicht möglich, jede Linie durch eine Kantenmenge auf der Haltestellenmenge darzustellen (als Multigraph).
2. Reale Linien werden durch das Modell abgebildet, indem Hin- und Rückrichtung durch jeweils eine Linie dargestellt werden.

Beispiel 2.3 :

Das Liniennetz $LNet = (S, L, LCon, name, koord, art, takt, abstand)$ zu Abbildung 2.5 besteht vor der Planung (Analyse, indiziert durch hochgestelltes A) aus der Haltestellenmenge $S^A = \{1, 2, 3, 4, 5, 6, 9\}$ und der Linienmenge $L^A = \{l_1, l_2, l_3, l_4\}$. Die Linienmenge ist in Abbildung 2.3 zusammen mit der Taktzeit $takt$ dargestellt. Die Anbindungen der Verkehrszellen an die Haltestellen $LCon$ werden gemeinsam mit den Anbindungslängen $abstand$ in Tabelle 2.4 (links) dargestellt.

Das Liniennetz für die Planung (Prognose, indiziert durch hochgestelltes P) besteht aus der Haltestellenmenge $S^P = S^A \cup \{7, 8\}$ und der Linienmenge $L^P = L^A \cup \{l_5, l_6\}$. Die zusätzlichen Linien sind in Tabelle 2.3 (untere Hälfte) dargestellt. Die Anbindungen der Verkehrszellen an die Haltestellen werden in Tabelle 2.4 (rechts) dargestellt.

Die Zug- und die Bushaltestelle in „Fernheim“ werden als eine einzige identische Haltestelle (6) betrachtet. Dies ist notwendig, da das Umsteigen nur an Haltestellen erfolgen kann.

Linien gehen hier immer nur in eine Richtung, reale Linien werden also in je eine Linie für Hin- und Rückrichtung aufgespalten. So ist die Buslinie von „Netzstadt“ nach „Fernheim“ zerlegt worden in l_1 und l_2 . Linie l_1 fährt dabei von Haltestelle 1 nach Haltestelle 2 in 3 Minuten, anschließend nach 3 in 5 Minuten, usw..

Jede Linie ist mit einer Taktzeit ($takt$) beschriftet. Die Buslinie von „Netzstadt“ nach „Fernheim“ hat eine Taktzeit von 30 Minuten.

$l \in L^A$							$takt(l)$	
l_1	R	1	2	3	4	5	6	30
	$time/60$	3	5	4	2	6		
l_2	R	6	5	4	3	2	1	30
	$time/60$	6	2	4	5	3		
l_3	R	10	11	12	13			30
	$time/60$	2	3	2				
l_4	R	13	12	11	10			30
	$time/60$	2	3	2				
$l \in L^P$								
l_5	R	7	8	5	9			30
	$time/60$	5	3	5				
l_6	R	9	5	8	7			30
	$time/60$	5	3	5				

Tabelle 2.3: Die Linien und ihre Beschriftung in der Umgebung von „Netzheim“

$lcon \in LCon^A$	$abstand(lcon)$	$lcon \in LCon^P$	$abstand(lcon)$
(NH_Mitte,4)	350	(NH_Mitte,4)	350
(NH_NW,4)	500	(NH_NW,8)	200
(NH_O,5)	200	(NH_O,5)	200
(NH_SW,4)	150	(NH_SW,4)	150
(Nahheim,4)	6000	(Nahheim,7)	250
(Fernheim,6)	300	(Fernheim,6)	300
(Beiheim,9)	200	(Beiheim,9)	200
(Stadt,1)	350	(Stadt,1)	350
(Stadt,2)	450	(Stadt,2)	450

Tabelle 2.4: Anbindungen und ihre Beschriftung vor der Planung (links) und für die Planung (rechts)

Jede Anbindung ist mit der Weglänge $abstand$ in Metern beschriftet. Vor der Planung ist „Nahheim“ mit öffentlichen Verkehrsmitteln nicht erreichbar. Die nächste Bushaltestelle ist in „Netzheim“. In der Tabelle 2.4 (links) wird dies durch einen Anbindungsfußweg von 6000 Metern deutlich. Mit der geplanten neuen Linie schmilzt dieser Fußweg auf 250 Meter zur neuen Haltestelle 7 in „Nahheim“. Am IEUV werden Linien mit verschiedenen Taktzeiten für verschiedene Verkehrszeiten (Haupt-, Neben-, oder Spätverkehrszeit) beschriftet. So ist es möglich, die Wartezeit für verschiedene Verkehrszeiten unterschiedlich zu berechnen.

2.3.2.4 Abstandsmodell im ÖV

Für die Berechnung der Verbindungskosten und besten Wege im ÖV gibt es

Definition 2.7

Zu einem Untersuchungsraum U und einem Liniennetz $LNet = (S, L, LCon, name, koord, art, takt, abstand)$ wird definiert:

- Eine nicht-leere Liste

$r_{\text{öv}} = [(hst_1, l_1, hst_2), (hst_2, l_2, hst_3), \dots, (hst_{r_{\text{max}}-1}, l_{r_{\text{max}}-1}, hst_{r_{\text{max}}})]$
von drei-Tupeln aus je zwei Haltestellen und einer verbindenden Linie
heißt **ÖV-Route**, wenn gilt:

- $\forall 1 \leq i < r_{\text{max}} \exists k, m \in \mathbb{N} :$
 $l_i.R = [s_1, \dots, s_{k-1}, hst_i, s_{k+1}, \dots, s_{m-1}, hst_{i+1}, s_{m+1}, \dots]$
(Eine ÖV-Route ist eine Folge von Linienverbindungen zwischen Haltestellen)
- $\forall 1 \leq i, j \leq r_{\text{max}} : hst_i \neq hst_j$
(Eine ÖV-Route enthält keine doppelten Haltestellen)

- Eine ÖV-Route

$r_{\text{öv}} = [(hst_1, l_1, hst_2), \dots, (hst_{r_{\text{max}}-1}, l_{r_{\text{max}}-1}, hst_{r_{\text{max}}})]$ heißt **ÖV-Verbindung** von $cell_s$ nach $cell_z$, falls gilt

1. $(cell_s, hst_1) \in LCon$
(Die Start-Verkehrszelle ist an den Startknoten der Route angebunden)
2. $(cell_z, hst_{r_{\text{max}}}) \in LCon$
(Die Ziel-Verkehrszelle ist an den Zielknoten der Route angebunden)
3. $cell_s \neq cell_z$
(ÖV-Routen innerhalb von Verkehrszellen werden nicht berücksichtigt)

- Die **Menge aller ÖV-Verbindungen** von Verkehrszelle s nach Verkehrszelle z ist definiert als

$$R_{\text{öv}}(s, z) = \{r \mid r \text{ ist ÖV-Verbindung von } s \text{ nach } z\}$$

- Die **Kosten** einer Einzelfahrt (hst_i, l_i, hst_{i+1}) mit $\pi(l_i.R, k) = hst_i \wedge \pi(l_i.R, m) = hst_{i+1} \wedge k < m$ ergeben sich zu

$$Cost_f(hst_i, l_i, hst_{i+1}) = \sum_{k \leq j < m} l_i.time(j)$$

- Die **Kosten** einer ÖV-Verbindung

$r = [(hst_1, l_1, hst_2), \dots, (hst_{r_{\text{max}}-1}, l_{r_{\text{max}}-1}, hst_{r_{\text{max}}})] \in R_{\text{öv}}(s, z)$ ergeben sich zu $Cost_{\text{öv}}(s, z, r) = Cost_s(s, hst_1) + Cost_r(r) + Cost_z(z, hst_{r_{\text{max}}})$ und bestehen aus

1. Startkosten für die Anbindung der Start-Verkehrszelle
 $Cost_s(s, hst_1) = \frac{abstand(cell_s, hst_1)}{1.5}$.
(unter Annahme einer mittleren Gehgeschwindigkeit von 1.5 m/s)

2. Routenkosten für die Route

$$Cost_r(r) = \sum_{1 \leq i < r_max} Cost_f(hst_i, r_i, hst_{i+1})$$

3. Zielkosten für die Anbindung der Ziel-Verkehrszelle

$$Cost_z(z, hst_{r_max}) = \frac{abstand(cell_s, hst_{r_max})}{1.5}.$$

(unter Annahme einer mittleren Gehgeschwindigkeit von 1.5 m/s)

- Der **ÖV-Abstand** zwischen zwei Verkehrszellen $s, z \in U$ ist definiert als $D_{\ddot{O}V}(s, z) = Min\{Cost_{\ddot{O}V}(s, z, r) \mid r \in R_{\ddot{O}V}(s, z)\}$.
- Eine ÖV-Verbindung $r \in R_{\ddot{O}V}(s, z)$ heißt **besten ÖV-Weg** von Verkehrszelle s nach Verkehrszelle $z \Leftrightarrow Cost_{\ddot{O}V}(s, z, r) = D_{\ddot{O}V}(s, z)$.

Bemerkung zur Definition: Die Kostenfunktion $Cost_{\ddot{O}V}$ stellt auch hier das Kostenmodell dar. Ich habe auch hier eine lineare Summierung der Einzelverbindungen gewählt. Am IEUV werden teilweise auch andere Parameter berücksichtigt (z.B. Umsteigehäufigkeit).

Beispiel 2.4 : In unserem Beispiel müssen wir zwischen Analyse und Prognose unterscheiden. Dabei betrachte ich nur die jeweils besten ÖV-Wege.

Vor der Planung (Analyse) ist die ÖV-Verbindung $[(4, l_2, 2)]$ der beste ÖV-Weg zwischen den Verkehrszellen „Nahheim“ und „Stadt“ (Netzstadt). Die Kosten errechnen sich wie folgt:

$$\begin{aligned} Cost_{\ddot{O}V}(\text{Nahheim, Stadt}, [(4, l_2, 2)]) &= Cost_s(\text{Nahheim}, 4) + Cost_f(4, l_2, 2) \\ &\quad + Cost_s(\text{Stadt}, 2) \\ &= \frac{6000}{1.5} + 240 + 300 + \frac{450}{1.5} \\ &= 4000 + 540 + 300 = 4840 \end{aligned}$$

Für die Planung (Prognose) ergibt sich die ÖV-Verbindung $[(7, l_5, 5), (5, l_2, 2)]$ als besten ÖV-Weg zwischen den Verkehrszellen „Nahheim“ und „Stadt“ (Netzstadt). Die Kosten errechnen sich analog:

$$\begin{aligned} Cost_{\ddot{O}V}(\text{Nahheim, Stadt}, [(7, l_5, 5), (5, l_2, 2)]) &= Cost_s(\text{Nahheim}, 7) \\ &\quad + Cost_f(7, l_5, 5) \\ &\quad + Cost_f(5, l_2, 2) \\ &\quad + Cost_s(\text{Stadt}, 2) \\ &= \frac{250}{1.5} + 300 + 180 + 120 \\ &\quad + 240 + 300 + \frac{450}{1.5} \\ &= 167 + 480 + 660 + 300 \\ &= 1607 \end{aligned}$$

Der ÖV-Abstand zwischen „Nahheim“ und „Stadt“ verbessert sich durch die Planung der neuen Buslinie von 4840 auf 1607 (als Reisezeit in Sekunden).

2.3.3 Verkehrsnachfrage

Die Verkehrsnachfrage als Zielgröße der Verkehrsprognose wird bezüglich Start, Ziel, Verkehrsmittel und Fahrtroute differenziert.

Definition 2.8 Verkehrsnachfrage

Sei $U = \{cell_1, \dots, cell_{u_max}\}$ ein Untersuchungsraum, $IVNet = (V, E, IVCon, time)$ ein Straßennetz zu U und $LNet = (S, L, LCon, name, koord, art, takt, abstand)$ ein Liniennetz zu U .

- $f(s, z, m, r) \in \mathbf{N}$ ist die Fahrtenzahl im Untersuchungszeitraum mit
 - der **Startverkehrszelle** $s \in U$,
 - der **Zielverkehrszelle** $z \in U$,
 - dem **Verkehrsmodus** $m \in \{iv, öv\}$ und
 - der **Route** $r \in R_{iv}(s, z)$, falls $m = iv$ bzw. $r \in R_{öv}(s, z)$, falls $m = öv$.
- Die **Verkehrsnachfrage** F_{ijmr} ist die vierdimensionale Matrix aller $f(s, z, m, r)$.
- Der **Verkehr zwischen s und z mit Verkehrsmittel m** $f(s, z, m) = \sum_{r \in R_m(s, z)} f(s, z, m, r)$ ist die Zahl aller Fahrten von Startverkehrszelle s nach Zielverkehrszelle z mit Verkehrsmodus m .
- Der **Modal-Split** F_{ijm} ist die dreidimensionale Matrix aller $f(s, z, m)$.
- Der **Verkehr zwischen s und z** $f(s, z) = f(s, z, iv) + f(s, z, öv)$ ist die Zahl aller Fahrten von Startverkehrszelle s nach Zielverkehrszelle z .
- Die **Fahrtenmatrix**

$$F_{ij} = \begin{pmatrix} f(cell_1, cell_1) & \dots & f(cell_1, cell_{u_max}) \\ \vdots & \ddots & \vdots \\ f(cell_{u_max}, cell_1) & \dots & f(cell_{u_max}, cell_{u_max}) \end{pmatrix}$$

ist die Matrix aller Quell-/Zielverkehre.

- Der **Quellverkehr der Verkehrszelle s** $f_Q(s) = \sum_{z \in U} f(s, z)$ ist die Zahl aller Fahrten mit Startverkehrszelle s .
- Der **Quellverkehr** $F_Q = (f_Q(cell_1), \dots, f_Q(cell_{u_max}))$ ist der Vektor aller Quellverkehre.
- Der **Zielverkehr der Verkehrszelle z** $f_Z(z) = \sum_{s \in U} f(s, z)$ ist die Zahl aller Fahrten mit Zielverkehrszelle z .
- Der **Zielverkehr** $F_Z = (f_Z(cell_1), \dots, f_Z(cell_{u_max}))$ ist der Vektor der Zielverkehre.

Start s	Ziel z	Modal-Split		Fahrtenmatrix
		$f_A(s, z, iv)$	$f_A(s, z, öv)$	$f_A(s, z) = f_P(s, z)$
NH_Mitte	NH_NW	50	-	50
NH_NW	NH_Mitte	50	-	50
NH_Mitte	NH_O	50	-	50
NH_O	NH_Mitte	50	-	50
NH_Mitte	NH_SW	50	-	50
NH_SW	NH_Mitte	50	-	50
NH_Mitte	Stadt	200	100	300
Stadt	NH_Mitte	200	100	300
NH_Mitte	Nahheim	200	-	200
Nahheim	NH_Mitte	200	-	200
NH_Mitte	Fernheim	20	10	30
Fernheim	NH_Mitte	20	10	30
NH_Mitte	Beiheim	30	-	30
Beiheim	NH_Mitte	30	-	30

Tabelle 2.5: Der Modal-Split (Analyse) sowie die Fahrtenmatrix (Analyse = Prognose) für das Beispiel „Netzheim“

Beispiel 2.5 : Der Modal-Split vor der Planung (Verkehrsnachfrage nach Start, Ziel und Verkehrsmittel) ist in Tabelle 2.5 dargestellt. Dabei wurden nur diejenigen Start-Ziel-Beziehungen berücksichtigt, bei denen „NH_Mitte“ entweder Start oder Ziel ist.

Zwischen „NH_Mitte“ und „Fernheim“ gibt es also in beiden Richtungen eine Fahrtenzahl von 20 im Individualverkehr und 10 im Öffentlichen Verkehr.

Für die Prognose wird in unserem Beispiel davon ausgegangen, daß sich die Gesamtfahrtenzahl nicht ändert. Nur der Modal-Split muß aufgrund der neuen Buslinie neu berechnet werden. Quell- und Zielverkehr sowie die Fahrtenmatrix ändern sich nicht. Die Fahrtenmatrix für Analyse und Prognose ergibt sich daher als Summe aus Individualverkehr und Öffentlichem Verkehr.

2.4 Prognose mit dem Vierstufenmodell

Der Vierstufenalgorithmus berechnet die Verkehrsnachfrage in vier Schritten. Statt einer formalen Beschreibung folgt hier die genaue Aufteilung in die vier Teilmodelle.

Verkehrserzeugung: Modell für den Zusammenhang zwischen statistischen Größen je Verkehrszelle (z.B. Beschäftigten- oder Einwohnerzahl) und den Vektoren der Quellverkehre F_Q und der Zielverkehre F_Z .
(siehe auch Studienarbeit von Thomas Zeitler [Zei95])

Verkehrsverteilung: Modell für den Zusammenhang zwischen den Abständen zwischen Verkehrszellen (D_{iv} , $D_{öv}$) einerseits und der

Fahrtenmatrix andererseits. Dieser Modellzusammenhang soll meist auch die Randbedingungen $f_Q(s) = \sum_{z \in U} f(s, z)$ und $f_Z(z) = \sum_{s \in U} f(s, z)$ erfüllen.

(ebenfalls in der Studienarbeit von Thomas Zeitler [Zei95])

Verkehrsmittelwahl: Modell für die Aufteilung der Fahrtenmatrix F_{ij} in den Modal-Split F_{ijm} . Diese Aufteilung wird in Abhängigkeit von den Kosten je Verkehrsmodus modelliert.

(z.B. mit dem Logit/Probit-Entscheidungsmodell aus der Diplomarbeit von Werner Schumacher [Sch95])

Verkehrsumlegung: Modell für die Verteilung des Modal-Split F_{ijm} auf die einzelnen ÖV- und IV-Routen. Das Ergebnis ist die Verkehrsnachfrage F_{ijmr} , die Umlegung berücksichtigt die k-kürzesten Wege.

(siehe auch Diplomarbeit von Friedrich Schedel [Sch90])

Bemerkungen:

- Die Eichung der einzelnen Modellzusammenhänge erfolgt mit Hilfe der Daten aus der Analyse.
- Die Teilmodelle können in beliebiger Reihenfolge geeicht werden.
- Aus den Modellen für das Verkehrsangebot werden nur die Kosten und der Verlauf der jeweils besten Route (oder Routen) benötigt. Der Ablauf der Bestwegsuche kann der Diplomarbeit von Jürgen Hoffmann [Hof94] entnommen werden.

Bei der Verkehrsprognose müssen natürlich nicht alle Teilmodelle geeicht oder gerechnet werden. Vielmehr wird die Prognose den bekannten Eingangsgrößen aus der Analyse und den geforderten Zielgrößen angepaßt. In Abbildung 2.6 ist der Ablauf einer vollständigen Prognose dargestellt. Die Daten der Analyse sind mit einem tiefgestellten A indiziert (Bsp.: $F_{ijm,A}$), die Größen der Prognose mit einem tiefgestellten P (Bsp.: $Struc_P$).

Wie man leicht sehen kann, gelangt man über die verschiedenen Teilmodelle zu den immer feiner differenzierten Verkehrsnachfragen. Schließlich ist die Verkehrsnachfrage bekannt und nach Start, Ziel, Verkehrsmodus und Route differenziert. Das jeweilige Teilmodell muß vor dem Einsatz geeicht werden.

Vielfach wird jedoch keine vollständige Prognose verlangt. Es genügt dann die Verkehrsnachfrage mit Differenzierung nach Start, Ziel und Verkehrsmittel. Entsprechend sind dann auch Analysedaten ohne Umlegung (Modal-Split) ausreichend, das Teilmodell „Umlegung“ muß nicht geeicht werden.

Oft wird auch die Entwicklung der Fahrtenmatrix direkt prognostiziert, so daß sich die Durchführung der Verkehrserzeugung und der Verkehrsverteilung erübrigt.

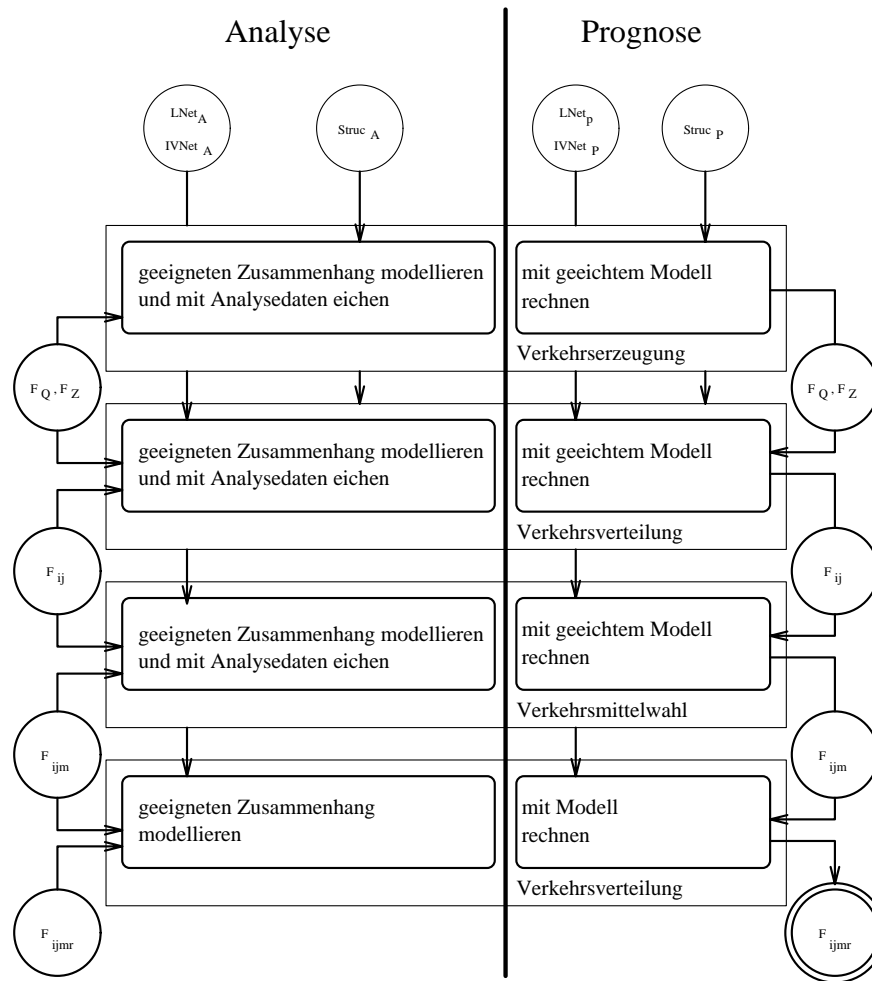


Abbildung 2.6: Prognose der Verkehrsnachfrage mit dem Vierstufenmodell

2.5 Kritik

Das Vierstufenmodell stellt die Teilmodelle Verkehrsentstehung, Verkehrsverteilung, Verkehrsmittelwahl und Verkehrsumlegung in eine bestimmte Reihenfolge. Obwohl es viele Variationen dieses Modells gibt, die mehrere Teilmodelle zusammenfassen (s.a. Steierwald [Ste94]), bleibt die Grundreihenfolge doch immer bestehen. Dabei wird vernachlässigt, daß die Verkehrsentstehung auch durch die Verkehrsmittelwahl und die Routenwahl beeinflusst wird, da die Attraktivität von Verbindungen bei Überlast stark nachläßt. In Wirklichkeit befinden sich die modellierten Teilentscheidungen der Individuen alle in einem globalen Gleichgewicht. Dieser Kritikpunkt führt in [Opp95] auf einen anderen Ansatz zur Modellierung dieses Gleichgewichtes.

Außerdem gehen durch die grobe Modellierung des Verkehrsangebotes Nachteile einer Verbindung unter, die nur eine bestimmte Personengruppe oder einen bestimmten Zeitraum betreffen.

Dazu gehören unter anderem:

- verspätete oder überfüllte Busse,
- ÖV-Mittel, die nachts nicht fahren,
- Umsteigevorgänge, die nachts von Frauen gemieden werden,
- schlechte Umsteigemöglichkeiten,
- Wettereinflüsse.

Busse sollten daher in Konkurrenz zum normalen Straßenverkehr modelliert werden. ÖV-Mittel sollten nicht nach Taktzeiten sondern nach Fahrplan und mit Verspätungen modelliert werden. Die Benutzung von Wegen und Verkehrsangeboten sollte von den individuellen Eigenschaften der Personen abhängig modelliert werden.

Ein Simulationsmodell für die Abläufe im Verkehr kann viele dieser Aspekte berücksichtigen und ist deshalb als Analysewerkzeug eine sinnvolle Ergänzung zur Verkehrsnachfrageprognose mit dem Vierstufenmodell.

Kapitel 3

Simulationsmodell für den ÖPNV

In diesem Kapitel wird ein Simulationsmodell für den Verkehrsablauf im ÖPNV entwickelt. Wesentlicher Unterschied zu anderen Simulationsmodellen ist dabei die getrennte Simulation von Personen und unterschiedlichen Fahrzeugen. Damit wird die Modellierung größerer Realitätsausschnitte als bisher möglich. Die Bewegung von Fahrzeugen kann nicht als Selbstzweck fehlinterpretiert werden, stattdessen wird die Bewegung von Personen (und Gütern) und damit die eigentliche Ursache von Fahrzeugbewegungen abgebildet.

Die Modellbildung erfolgt in zwei Schritten. In Abschnitt 3.1 werden zunächst die abstrakten Konzepte informal beschrieben, die abgebildet werden sollen. Die Idee zur graphischen Darstellung des Fahrtverlaufs wurde dabei der Diplomarbeit von Stefan Preuß [Pre93] entnommen. In Abschnitt 3.2 wird dann ein formales Modell für die Simulation des ÖPNV aufgestellt und mit einfachen Beispielen erläutert.

3.1 Die Konzepte

Das Modell soll so allgemein sein, daß die Feinkörnigkeit der Abbildung des realen Verkehrsangebots nicht festgelegt ist, sondern dem Untersuchungsgegenstand angepaßt werden kann.

Folgende Konzepte sind die tragenden Ideen des Simulationsmodells:

- Verkehrsobjekt ist nur ein Oberbegriff für verschiedene Arten von dynamischen Objekten wie Personen, IV- und ÖV-Mittel, die alle getrennt modelliert werden.
- Verkehrsnetze sollen nach ihren betrieblichen Eigenschaften für die zugehörigen Verkehrsobjekte modelliert werden.
- Verkehrsobjekte tragen eine Beschreibung ihrer Route. Die Beschreibungsgenauigkeit ist variabel.

- Verkehrsobjekte (und damit auch die benutzten Verkehrsnetze) können in einer Mitnahmebeziehung zueinander stehen (z.B. nehmen Züge Personen mit).
- Die Anzahl von Verkehrsobjekten, die sich auf Verkehrsnetzen oder in anderen Verkehrsobjekten bewegen kann, ist beschränkt.

Wesentliches Ziel einer Simulation ist die anschließende Analyse der gewonnenen Daten. Folgende Daten sollen nach einem Simulationslauf zur Verfügung stehen:

- zeitlicher Fahrtverlauf für alle Verkehrsobjekte.
- zeitlicher Verlauf der Belastung von Verkehrsnetzen und Fahrzeugen.

In den folgenden Teilen dieses Abschnitts werden die einzelnen Konzepte genauer vorgestellt und motiviert, dabei werden neu eingeführte Begriffe *kursiv* geschrieben.

3.1.1 Verkehrsobjekte und Verkehrsnetze

Die beweglichen Einheiten einer Verkehrssimulation heißen *Verkehrsobjekte*. Verkehrsobjekte werden in *Quellen* erzeugt, bewegen sich durch das *Verkehrsnetz* und werden in *Senken* wieder gelöscht. In Abbildung 3.1 ist ein Verkehrsnetz zusammen mit Quelle und Senke abgebildet.

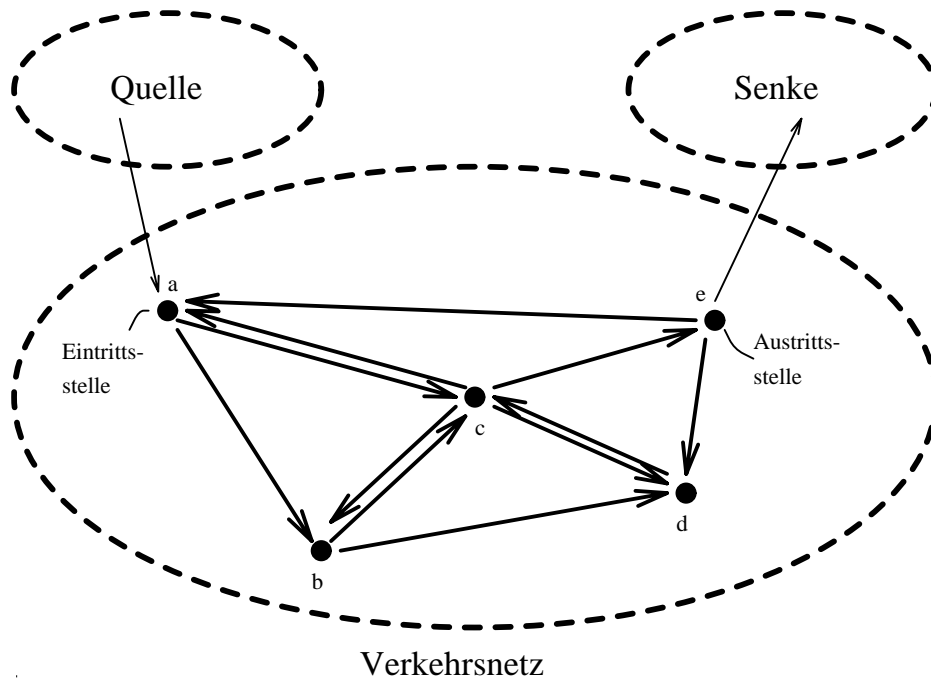


Abbildung 3.1: Abstraktion eines Verkehrsnetzes

Die Knoten im Verkehrsnetz heißen *Orte* und sind ein Modell für reale Aufenthaltspunkte oder -bereiche von Verkehrsobjekten. Die Kanten im Verkehrsnetz sind gerichtet und heißen *Verbindungen*. Die Verbindungen sind ein Modell für die Bewegungsmöglichkeiten der Verkehrsobjekte zwischen den Orten. Ein Verkehrsnetz ist also ein Multigraph aus Orten und Verbindungen. Ein Verkehrsnetz wird immer für eine bestimmte *Art* von Verkehrsobjekten festgelegt. Die Verbindungen zwischen Orten können daher speziell für diese Art festgelegt werden. Den Orten kann eine Kapazität zugeordnet werden, die sich auf die jeweilige Art bezieht. Dabei wird die Kapazität von Verbindungen nicht explizit modelliert, sondern den Orten zugeschlagen. Beispiele für Arten sind Person, Bus, Zug, usw.. Die Bewegung von Verkehrsobjekten ist nur innerhalb eines Verkehrsnetzes möglich.

Ein Ort, an dem neue Verkehrsobjekte in das Verkehrsnetz eintreten, heißt *Eintrittsstelle*. So ist die Schnittstelle zwischen Quelle und Verkehrsnetz eine Eintrittsstelle. Analog heißt ein Ort, an dem Verkehrsobjekte aus dem Verkehrsnetz austreten, *Austrittsstelle*.

Die Bewegung der Verkehrsobjekte innerhalb der Verkehrsnetze erfolgt nur entlang der gerichteten Verbindungen des Verkehrsnetzes.

3.1.2 Routen

Die Verkehrsobjekte sollen sich nicht zufällig durch das Verkehrsnetz bewegen, da bei zufälliger Bewegung keine Effekte beobachtet werden können, die auf der planvollen Bewegung zwischen bestimmten Orten beruhen. Daher besitzt jedes Verkehrsobjekt eine *Routenbeschreibung*. Die Routenbeschreibung ist dabei eine Liste von Orten, den *Routenpunkten*, die nacheinander angesteuert werden sollen (geplante Route). Da sich Verkehrsobjekte nur entlang der Verbindungen bewegen können, soll eine Routenbeschreibung in eine *Route* umgesetzt werden können. Eine Route ist dabei ein Weg (d.h. eine Folge von Verbindungen) durch das Verkehrsnetz. Die Route beginnt an der Eintrittsstelle des Verkehrsobjektes und endet an der Austrittsstelle des Verkehrsobjektes. Eine Sonderstellung nehmen die Routenbeschreibungen von Personen ein, da sie ihr Verkehrsnetz auch verlassen können, um ein Verkehrsmittel zu benutzen und ihr Verkehrsnetz dann an anderer Stelle wieder zu betreten. Auf diesen Fall wird in Abschnitt 3.1.4 eingegangen.

In Abbildung 3.2 ist ein Ausschnitt aus einem Verkehrsnetz zusammen mit einer möglichen Route abgebildet. Dabei wurde mit Absicht eine Schleife eingebaut, um den Fall so allgemein wie möglich zu halten. In der Realität werden Schleifen z.B. von öffentlichen Verkehrsmitteln gefahren.

Die Routenbeschreibung zur gezeigten Route ([a,b,c,d,b,e,f]) könnte so aussehen:

Routenbeschreibung = [a, b, c, d, b, e, f]

Die Umsetzung zu einer Route erfolgt mit einer *Routentabelle*, die aus dem aktuellen Ort und dem nächsten Routenpunkt eine mögliche Verbindung ab dem aktuellen Ort bestimmt.

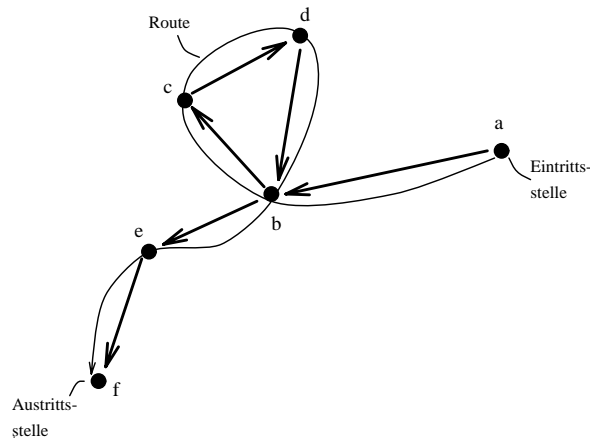


Abbildung 3.2: Route im Verkehrsnetz

Für den Ort b könnte die Routentabelle z.B. so aussehen:

aktueller Ort	nächster Routenpunkt	Verbindung
b	c	(b,c)
b	e	(b,e)

Die Auswahl der Verbindung hängt jetzt davon ab, ob die Schleife schon durchfahren wurde oder nicht. Im ersten Fall ist c der nächste Routenpunkt, im zweiten Fall ist es e.

Manchmal möchte man mit einer Routenbeschreibung vielleicht nicht den ganzen Weg, sondern nur einige wichtige Zwischenpunkte aufzählen. In diesem Fall muß die Routentabelle erweitert werden.

Will man z.B mit der eingeschränkten Routenbeschreibung [a, b, d, b, f] auskommen, so benötigt man folgende Routentabelle:

aktueller Ort	nächster Routenpunkt	Verbindung
b	c	(b,c)
b	d	(b,c)
b	e	(b,e)
b	f	(b,e)

Befindet sich das Verkehrsobjekt auf Ort b und ist der nächste Routenpunkt f, so kann die richtige Verbindung (b,e) aufgrund der Routentabelle gewählt werden.

Bemerkung: Die Routentabelle verkörpert das Routenwissen, die Genauigkeit der Routenbeschreibung orientiert sich am Umfang des Routenwissens. Kann zum aktuellen Ort und dem nächsten Routenpunkt kein Routentabelleneintrag gefunden werden, so ist die Routenbeschreibung nicht genau genug.

3.1.3 Das Zeitkonzept

Die Bewegung von Verkehrsobjekten hat einen bestimmten zeitlichen Verlauf, der in Abbildung 3.3 exemplarisch dargestellt ist.

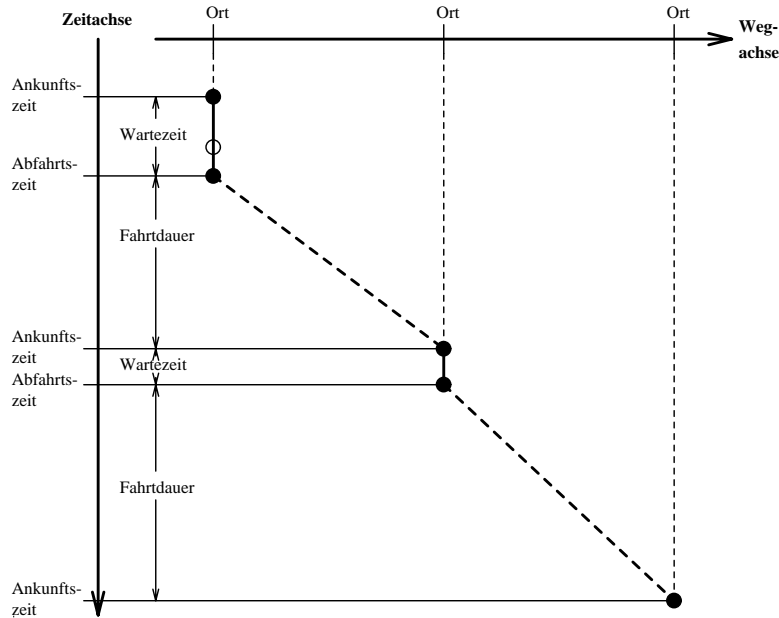


Abbildung 3.3: Zeitlicher Fahrtverlauf

Dieser zeitliche Verlauf heißt *zeitlicher Fahrtverlauf* und wird von *Ankunftszeit* und *Abfahrtszeit* zerlegt in

- *Wartezeiten* (an einem Ort zwischen Ankunfts- und Abfahrtszeit) und
- *Fahrtdauern* (als Zeitverbrauch für Verbindungen oder Mitfahrten von Ort zu Ort).

Der Zeitpunkt, an dem Verkehrsobjekte in das Verkehrsnetz eintreten, heißt *Eintrittszeit* und ist zugleich Ankunftszeit am jeweiligen Ort. Analog heißt der Zeitpunkt, an dem Verkehrsobjekte aus dem Verkehrsnetz austreten, *Austrittszeit*.

Zeit wird durch diskrete *Zeitpunkte* modelliert. Der Abstand zwischen zwei Zeitpunkten heißt *Zeiteinheit* und kann frei und je nach Modellierungsgenauigkeit gewählt werden.

3.1.4 Das Mitnahme-/Mitfahrkonzept

Manche Verkehrsobjekte (z.B. Personen) können von anderen Verkehrsobjekten (z.B. Zug) mitgenommen werden. Es mußte ein Konzept entwickelt werden, das einen entsprechenden Bezug zwischen den jeweiligen Netzen herstellt und mit den Routenbeschreibungen verträglich ist.

Das Verkehrsnetz der mitnehmenden Verkehrsobjekte heißt *Mitnahmenetz* und das Verkehrsnetz der mitfahrenden Verkehrsobjekte heißt *Mitfahrnetz*. In Abbildung 3.4 wird das Zusammenspiel der beiden Verkehrsnetze veranschaulicht.

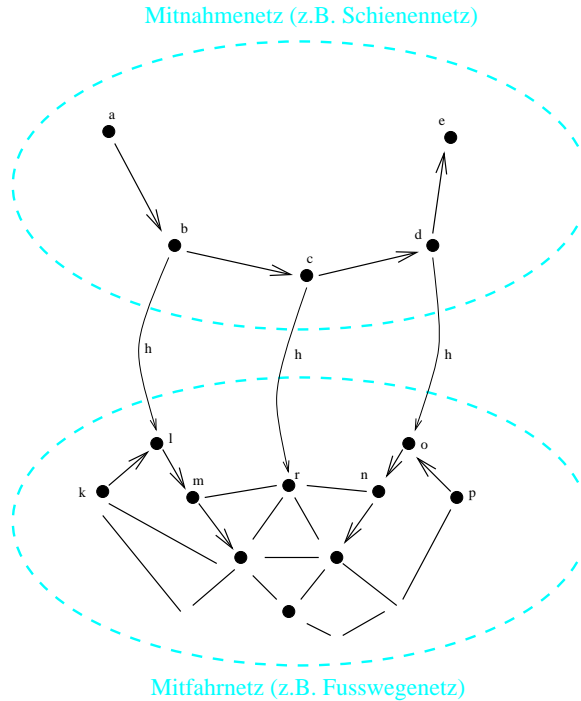


Abbildung 3.4: Mitfahrnetz und Mitnahmenetz

Die dargestellte Mitnahmebeziehung h bildet Orte des Mitnahmenetzes auf Orte des Mitfahrnetzes ab. Alle Orte des Mitnahmenetzes, auf denen h definiert ist, heißen *Haltepunkte*. Alle Orte des Mitfahrnetzes im Bildbereich der Abbildung heißen *Haltestellen*. Natürlich können auch mehrere Mitnahmenetze zum gleichen Mitfahrnetz existieren. Orte im Mitfahrnetz können dann gleichzeitig Haltestelle von mehreren Mitnahmenetzen sein (Abb. 3.5).

Gelangt ein Verkehrsobjekt mo des Mitnahmenetzes an einen Ort x seiner Routenbeschreibung¹, so wird jedes Verkehrsobjekt am Ort $h(x)$, dessen nächster Routenpunkt als Mitnahmebeziehung in der Routenbeschreibung von mo vorkommt, aus dem Mitfahrnetz entfernt und begleitet mo durch das Mitnahmenetz. Der Zeitpunkt, zu dem dies geschieht, heißt *Mitnahmezeitpunkt*. Der Vorgang heißt *Einstieg* und der Ort $h(x)$ *Einstiegsstelle*.

Gelangt ein Verkehrsobjekt mo des Mitnahmenetzes an einen Ort y seiner Routenbeschreibung und es wird von einem Verkehrsobjekt fo des Mitfahrnetzes begleitet, dessen nächster Routenpunkt $h(y)$ ist, so wird das Verkehrsobjekt fo in das Mitfahrnetz auf dem Ort $h(y)$ abgesetzt. Der Zeitpunkt, zu dem dies geschieht, heißt *Absetzzeitpunkt*. Der Vorgang heißt *Ausstieg* und der Ort $h(y)$ *Ausstiegsstelle*.

¹jeder Ort der Routenbeschreibung im Mitnahmenetz ist Haltepunkt

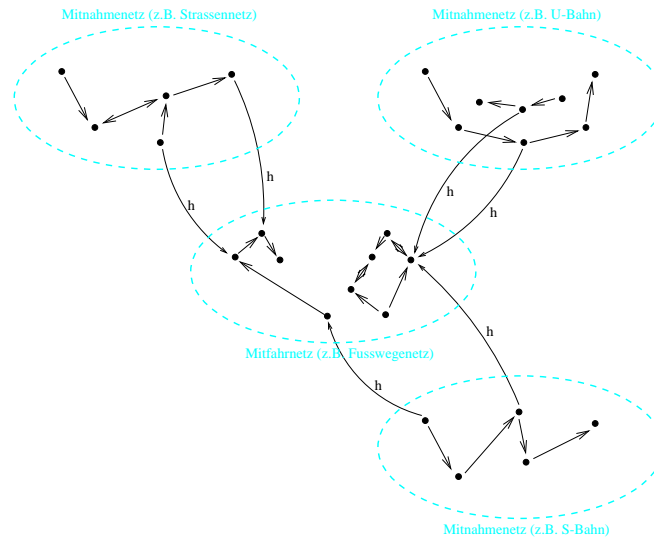


Abbildung 3.5: Drei Mitnahmenetze zu einem Mitfahrnetz

3.1.5 Fahrtverlauf

Für alle Arten von Verkehrsobjekten berechnet sich die Fahrtdauer zwischen zwei Orten aus den Eigenschaften der Verbindung (i.a. der Länge) und den Eigenschaften des Verkehrsobjekts (i.a. der Geschwindigkeit). Zur Vereinfachung wird jede Verbindung jedoch zunächst nur mit der Fahrtdauer beschriftet.

3.1.5.1 Person

Das wichtigste Verkehrsobjekt in einer Verkehrssimulation ist die *Person*. Das Verkehrsnetz für Personen heißt *Fußwegenetz*. Die Verbindungen heißen *Fußwege*.

Der Fahrtverlauf von Personen wird in Abbildung 3.6 dargestellt.

Wenn man die Reisedauer ohne Simulation berechnet, so kann man als Wartezeit die mittlere Wartezeit für eine bestimmte Linie verwenden. Die Fahrtdauer kann dem Fahrplan entnommen werden. Auch diese fiktive Reisedauer ist in der Abbildung angedeutet.

An normalen Routenpunkten haben Personen keine Wartezeit. Ist der nächste Routenpunkt nicht in der Routentabelle, so wartet die Person, bis sie von einem Verkehrsobjekt des Mitnahmenetzes mitgenommen wird.

Die Fahrtdauer im Mitnahmenetz muß nicht berechnet werden, sondern ergibt sich aus der Mitfahrt in den öffentlichen Verkehrsmitteln.

3.1.5.2 Die öffentlichen Verkehrsmittel

Obwohl einige öffentliche Verkehrsbetriebe auch bedarfsgesteuerte Fahrzeuge einsetzen, beschränkt sich dieses Modell auf Fahrzeuge wie z.B. Busse, Züge, S-

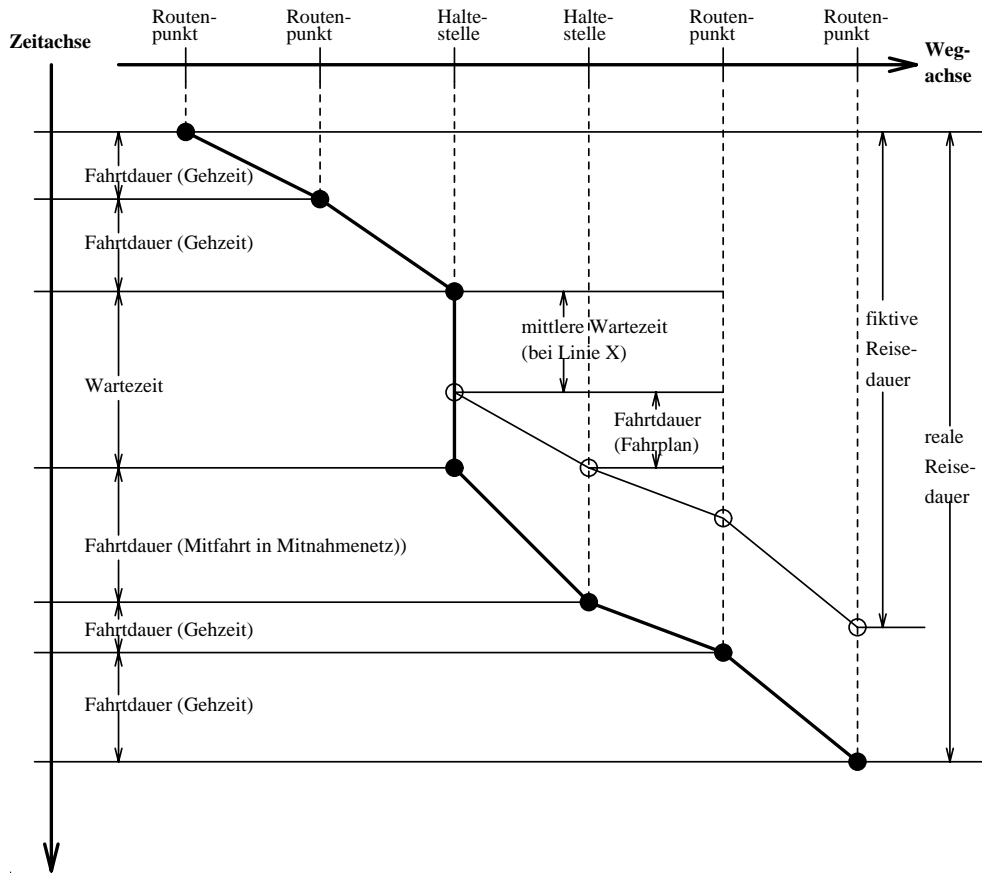


Abbildung 3.6: Der Fahrtverlauf von Personen

, U- und Straßenbahnen, die gemäß feststehendem Fahrplan eingesetzt werden. Solche Verkehrsobjekte werden von nun an mit *Fahrplanmittel* bezeichnet.

Der Fahrtverlauf von Fahrplanmitteln wird in Abbildung 3.7 dargestellt.

Die Haltezeiten ergeben sich bei Fahrplanmitteln aus

- der Zeit für Ein- und Aussteigen (minimale Wartezeit) und
- dem Abfahrtszeit nach Fahrplan (frühester Abfahrtszeitpunkt).

Fahrplanmittel haben daher neben einer Routenbeschreibung auch eine Beschriftung der einzelnen Routenpunkte mit *Planabfahrtszeit* und *Planankunftszeit*. Planankunfts- und Planabfahrtszeit ergeben sich unmittelbar aus dem Fahrplan. Die Haltezeit an einem Haltepunkt ist also

1. **mindestens** so lang wie die Zeit für Ein- und Aussteigen
2. **mindestens** so lang, bis die Modellzeit größer ist als die Abfahrtszeit.

Der letzte Haltepunkt eines Fahrplanmittels ist Austrittspunkt zu einer Senke. Die Eintrittszeit des Fahrplanmittels ist der Ankunftszeitpunkt am ersten Haltepunkt, die Austrittszeit ist der Abfahrtszeitpunkt am letzten Haltepunkt.

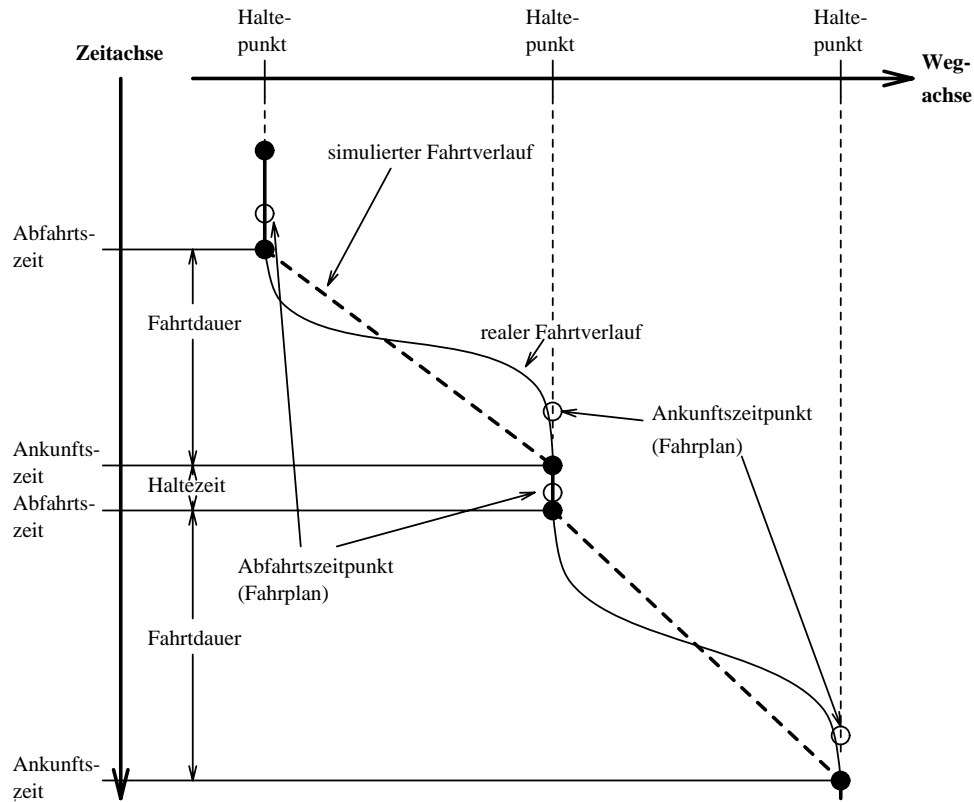


Abbildung 3.7: Fahrtverlauf von Fahrplanmitteln

3.1.6 Kapazitäten

Orte in Verkehrsnetzen haben eine *Kapazität*, Fahrplanmittel eine *Mitnahmekapazität*. Die Abfahrt von einem Ort ist nur möglich, wenn die Kapazität des folgenden Ortes nicht ausgeschöpft ist². Der Einstieg in ein Fahrplanmittel ist nur möglich, wenn die Mitnahmekapazität des Fahrplanmittels nicht ausgeschöpft ist. Der Ausstieg aus einem Fahrplanmittel ist nur möglich, wenn die Kapazität der Haltestelle nicht ausgeschöpft ist. Durch das starre Mitnahmekonzept verschwinden Verkehrsobjekte, die nicht aussteigen können, da sie ja später nie mehr an der Ausstiegsstelle vorbeikommen. Auch der Eintritt in Verkehrsnetze ist nur möglich, wenn noch Kapazitäten frei sind. Durch die Verwendung von Kapazitäten ist es z.B. auch möglich, Warteschlangen auf einer Schienenstrecke oder einer einspurigen Straße zu modellieren.

3.1.7 Erweiterungen

Folgende weitere Konzepte halte ich bei der Modellierung von Verkehr für wesentlich. In mein Grundmodell konnten sie nicht Eingang finden.

- Wettereinflüsse auf Verkehrsobjekte und -netze.

²Die Kapazität von Verbindungen wird dem Folgeort zugeschlagen.

- Kreuzungspunkte zwischen verschiedenen Verkehrsnetzen.
- Verbindungen werden mit Länge, Steigung und Qualität modelliert, die Fahrtdauer ergibt sich aus Eigenschaften der Verkehrsobjekte.
- Steuerungsregeln und -anlagen (z.B. Ampeln oder Vorfahrtsregeln), die bei einer zunehmenden Verfeinerung von Verkehrsnetzen notwendig werden.

3.2 Formalisierung der Konzepte (Spezifikation)

In diesem Abschnitt wird mein Modell für den ÖPNV und dessen Simulation formal gefaßt. Die exakte Spezifikation dient dabei einerseits als Grundlage für die Realisierung des Modells. Andererseits macht sie eine genaue Abbildung gegebener Daten in das Modell überhaupt erst möglich.

Neben mengentheoretischen Ausdrücken werden dabei prädikatenlogische Ausdrücke erster Ordnung verwendet. Zusätzlich benötige ich folgende

Notationen:

1. Statt partielle Abbildungen zu verwenden, wird ein ausgezeichnetes Element \mathbf{u} (undefined) eingeführt, auf das ebenfalls abgebildet werden kann.
2. Namen werden immer als Element der universalen Namenmenge $N = \{n \mid n \text{ ist Name}\}$ dargestellt.
3. Tupel werden rund $(())$ und Listen eckig $([])$ geklammert.
4. Auf Elemente eines Tupels kann mit der Punktschreibweise zugegriffen werden, z.B. besteht $o = (R, n)$ aus $o.R$ und $o.n$.

3.2.1 Verkehrsnetze

Zuächst sollen Verkehrsnetze als Modell für den Bewegungsraum von Verkehrsobjekten formal gefaßt werden. Dies geschieht in mehreren Schritten. Zunächst

Definition 3.1

(V, E) heißt **Verkehrsgraph** $:\Leftrightarrow$

- V ist eine endliche Menge von **Orten**.
- $E \subseteq (V \times V) \setminus \{(v, v) \mid v \in V\}$ ist eine Menge von **Verbindungen**.

Bemerkungen:

1. Ein Verkehrsgraph ist ein gerichteter, schleifenfreier Graph mit Orten als Knoten und Verbindungen als Kanten.
2. Die Verbindungen eines Verkehrsgraphen stellen die Bewegungsmöglichkeiten für zugehörige Verkehrsobjekte dar.

Eine Routenbeschreibung zu einer Ortsmenge V ist eine Liste von Orten aus V mit den üblichen Funktionen für Listen. Deshalb

Definition 3.2

Sei V ein Menge von Orten.

- $\mathfrak{R}(V) = \bigcup_{i \in \mathbb{N}} V^i$ ist die **Menge aller Listen über V** . $R \in V^k \subset \mathfrak{R}(V)$ heißt **Routenbeschreibung** über V .
Schreibweise: $R = [r_1, \dots, r_k] \in V^k$.
- $r_i \in V, 1 \leq i \leq k$ heißt **i -ter Routenpunkt** von $R = [r_1, \dots, r_k]$
- $\max(R) = k$ heißt **Länge** von $R : \Leftrightarrow R \in V^k$.
- $I(R) = \{1, \dots, k\}$ heißt **Indexmenge** von $R : \Leftrightarrow R \in V^k$.
- $v \in V$ **liegt in** $R = [r_1, \dots, r_k] : \Leftrightarrow \exists_{1 \leq i \leq k} : r_i = v$.
- $R_3 = R_1 \cdot R_2$ für $R_1, R_2 \in \mathfrak{R}(V)$ ist die **Konkatenation** von R_1 und R_2 .
- $\pi : \mathfrak{R}(V) \times \mathbb{N} \rightarrow V \cup \{u\}$ mit

$$\pi([r_1, \dots, r_k], i) = \begin{cases} r_i & 1 \leq i \leq k \\ u & i > k \end{cases}$$
ist die Projektion der Routenbeschreibung R auf ihren i -ten Routenpunkt.

Bemerkung: Routenbeschreibungen sind also Listen von Orten, die zunächst keinen Bezug zueinander haben.

Hat man einen Verkehrsgraph und eine Routenbeschreibung über dessen Ortsmenge, so ist es möglich, daß die Routenbeschreibung ein Weg in diesem Verkehrsgraph ist. Im folgenden werde ich sagen, die Routenbeschreibung liegt in der Kantenmenge.

Definition 3.3

Sei (V, E) ein Verkehrsgraph. Sei $R = [r_1, \dots, r_k] \in \mathfrak{R}(V)$. R **liegt in** $E : \Leftrightarrow \forall_{1 \leq i \leq k} : (r_i, r_{i+1}) \in E$.

Ein Verkehrsnetz muß neben dem Verkehrsgraph eine Routentabelle besitzen, die es erlaubt, Routenbeschreibungen auf Wege im Verkehrsgraphen abzubilden. Zunächst definieren wir Tabellen auf Ortsmengen.

Definition 3.4

(T, λ) heißt **Tabelle** zur Ortsmenge $V : \Leftrightarrow$

- $T \subseteq (V \times V) \setminus \{(v, v) \mid v \in V\}$ ist eine Menge von **Tabelleneinträgen** und
- $\lambda : T \rightarrow V$ bildet jeden Tabelleneintrag auf einen Ort ab.

Bemerkung: (V, T) ist ein gerichteter, schleifenfreier Graph mit Tabelleneinträgen als Kanten. λ bildet jede Kante auf einen weiteren Knoten ab.

Eine Routentabelle ist nun eine Tabelle, bei der man durch wiederholtes Anwenden der Abbildung λ weder in eine Sackgasse noch in einen Zyklus gerät. Mit ihrer Hilfe sollen im weiteren Verlauf dieses Abschnitts Routenbeschreibungen zu Wegen im Verkehrsgraphen ergänzt werden.

Beispiel 3.1 : Die Tabelle

$e \in T$	$\lambda(e)$
(a,d)	b
(b,d)	c
(c,d)	d

ist in Abbildung 3.8 (Rechts) illustriert. Orte sind durch Punkte, Tabelleneinträge durch gestrichelte dicke Pfeile und die Abbildung λ durch dünne Pfeile dargestellt. Zunächst sei der aktuelle Ort a und der nächste Routenpunkt d. Mit Hilfe der Tabelle kann festgestellt werden, daß man dazu zuerst zum Ort b muß. Nun ist b der aktuelle Ort und noch immer ist d der nächste Routenpunkt. Wegen $\lambda(b,d) = c$ muß man also zunächst nach c. Nach einem weiteren Schritt gelangt man zu d. Der nächste Routenpunkt konnte erreicht werden.

Um sicherzustellen, daß man mit einem Tabelleneintrag (a, b) immer von a nach b kommt, formuliere ich zunächst folgende Eigenschaften und dann Routentabellen.

Definition 3.5

Sei (T, λ) eine Tabelle zur Ortsmenge V .

- (T, λ) heißt **sackgassenfrei** $:\Leftrightarrow \forall (v_1, v_2) \in T : v_2 = \lambda(v_1, v_2) \vee (\lambda(v_1, v_2), v_2) \in T$.
- (T, λ) heißt **zyklisch** $:\Leftrightarrow \exists k \in \mathbb{N} \exists (v_1, v), \dots, (v_k, v) \in T : \lambda(v_1, v) = v_2 \wedge \dots \wedge \lambda(v_{k-1}, v) = v_k \wedge \lambda(v_k, v) = v_1$.

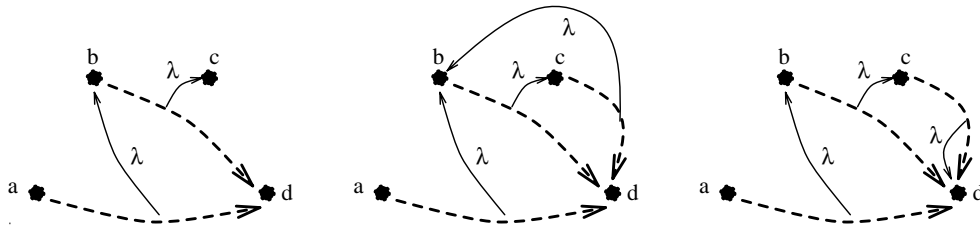


Abbildung 3.8: Beispiele für Tabellen auf Ortsmengen

In Abbildung 3.8 ist die linke Tabelle nicht sackgassenfrei, weil $\lambda(b,d) = c$, aber nicht $(c,d) \in T$. Die mittlere ist zyklisch. Die rechte Tabelle ist sackgassenfrei und nicht zyklisch.

Definition 3.6

Sei V eine Ortsmenge. Eine sackgassenfreie und nicht zyklische Tabelle zu V heißt **Routentabelle** zu V .

Verkehrsgraph und Routentabelle werden nun zum Routengraph zusammengefaßt. Dabei sollen nur mit dem aktuellen Ort verbundene Orte in die Tabelle eingetragen sein (Die Tabelle soll zum Verkehrsgraph passen).

Definition 3.7

(V, E, T, λ) heißt **Routengraph** $:\Leftrightarrow$

- (V, E) ist ein Verkehrsgraph,
- (T, λ) ist eine Routentabelle zu V und
- $\forall_{(v_1, v_2) \in T} : (v_1, \lambda(v_1, v_2)) \in E$.

Routenbeschreibungen zur Ortsmenge V können in E liegen, dann handelt es sich um eine Route, oder auch in T liegen, dann kann eine Route daraus gewonnen werden. Diese Eigenschaften sollen zunächst gefaßt werden.

Definition 3.8

Die Routenbeschreibung $R = [r_1, \dots, r_k]$ heißt **Route** im Routengraph $(V, E, T, \lambda) :\Leftrightarrow \forall_{1 \leq i < k} : (r_i, r_{i+1}) \in E$

Die Routenbeschreibung $R = [r_1, \dots, r_k]$ heißt **günstig** im Routengraph $(V, E, T, \lambda) :\Leftrightarrow \forall_{1 \leq i < k} : (r_i, r_{i+1}) \in T$

Bemerkung Eine Route ist also ein Weg auf den Verbindungen, eine günstige Routenbeschreibung ein Weg auf den Routentabelleneinträgen eines Routengraphen. In Abbildung 3.9 ist $[a, b, d, e, b, f]$ eine günstige Routenbeschreibung, aber keine Route (wg. $(b, d) \notin E$). $[a, b, c, d, e, b, f]$ ist zwar eine Route, aber keine Routenbeschreibung (wg. $(b, c) \notin T$).

Es ist klar, daß die notwendige Genauigkeit für Routenbeschreibungen in einem Routengraph direkt von der Routentabelle abhängt. Eine Routentabelle, die alle Routenbeschreibungen zur Route ergänzen kann, ist zu groß, um sie realisieren zu können. Leider lassen sich die Erkenntnisse zum Routingproblem (siehe z.B. Tanenbaum [Tan89]) nicht direkt anwenden, insbesondere lassen sich reale Verkehrsnetze nicht willkürlich hierarchisieren.

Nun können wir die Erzeugung einer Route aus einem Routentabelleneintrag formal fassen.

Definition 3.9

Sei (V, E, T, λ) ein Routengraph.

$\lambda^* : T \rightarrow \mathfrak{R}(V)$ heißt **Routenbildung zu T** $:\Leftrightarrow$

$$\lambda^*(v_1, v_2) = \begin{cases} [v_2] & \lambda(v_1, v_2) = v_2 \\ [\lambda(v_1, v_2)] \cdot \lambda^*(\lambda(v_1, v_2), v_2) & \text{sonst} \end{cases}$$

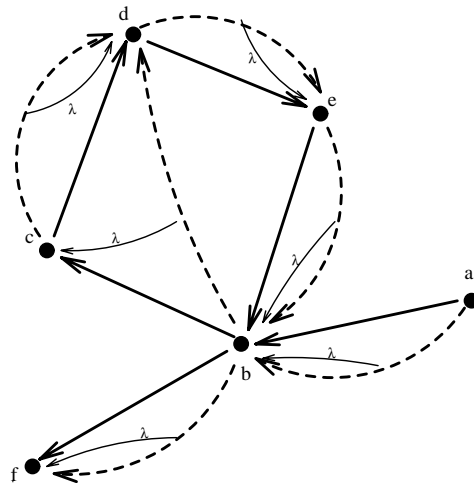


Abbildung 3.9: Routen und günstige Routenbeschreibungen

Bemerkungen:

1. Die Sackgassenfreiheit von (T, λ) stellt sicher, daß ein Routeneintrag einen Nachfolgeeintrag besitzt $((\lambda(v_1, v_2), v_2) \in T)$ falls er nicht unmittelbar ans Ziel führt $(\lambda(v_1, v_2) \neq v_2)$.
2. Wegen der Schleifenfreiheit von λ und der Endlichkeit der Ortmenge V ist ein Abbruch der rekursiven Definition von λ^* gewährleistet.
3. Der Startpunkt v_1 ist nicht Teil von λ^* , um die Routenbildung auf mehrere Abschnitte anwenden zu können.
4. $\lambda^*(v_1, v_2)$ ist eine Route im Routengraph.

Definition 3.10

Sei (V, E, T, λ) ein Routengraph und $R = [r_1, \dots, r_k] \in \mathfrak{R}(V)$ eine günstige Routenbeschreibung im Routengraph. $\lambda^*(R) = [r_1] \cdot \lambda^*(r_1, r_2) \cdot \dots \cdot \lambda^*(r_{k-1}, r_k)$ heißt **Routenbildung zu R** .

Bemerkung: $\lambda^*(R)$ ist eine Route im Routengraph.

Beispiel 3.2 : In Abbildung 3.9 ist $[a, b, c, d, e, b, f]$ Routenbildung von $[a, b, d, e, b, f]$ wegen $\lambda^*(a, b) = [b]$, $\lambda^*(b, d) = [c, d]$, usw.

Ein Routengraph wird zum Verkehrsnetz als Bewegungsraum bestimmter Verkehrsobjekte, indem die Orte mit Koordinaten, Name und Kapazität sowie die Verbindungen mit Länge und Fahrtdauer beschriftet werden. Da das Simulationsmodell auf einer diskreten Zeitskala basiert, werden Zeiträume und Zeitpunkte durch natürliche Zahlen dargestellt.

Definition 3.11

$\vartheta = (V, E, T, \mu, \nu, \kappa, \delta, \tau, \lambda)$ heißt **Verkehrsnetz** : \Leftrightarrow

- (V, E, T, λ) ist ein Routengraph.
- $\mu : V \rightarrow \mathbb{R} \times \mathbb{R}$ ordnet jedem Ort seine **Koordinaten** zu.
- $\nu : V \rightarrow N$ beschriftet jeden Ort mit einem **Namen**.
- $\kappa : V \rightarrow \mathbb{N} \cup \{\infty\}$ ordnet jedem Ort seine **Kapazität** zu.
- $\delta : E \rightarrow \mathbb{R}^+$ ordnet jeder Verbindung eine **Länge** zu.
- $\tau : E \rightarrow \mathbb{N}$ ordnet jeder Verbindung eine **Fahrtdauer** zu.

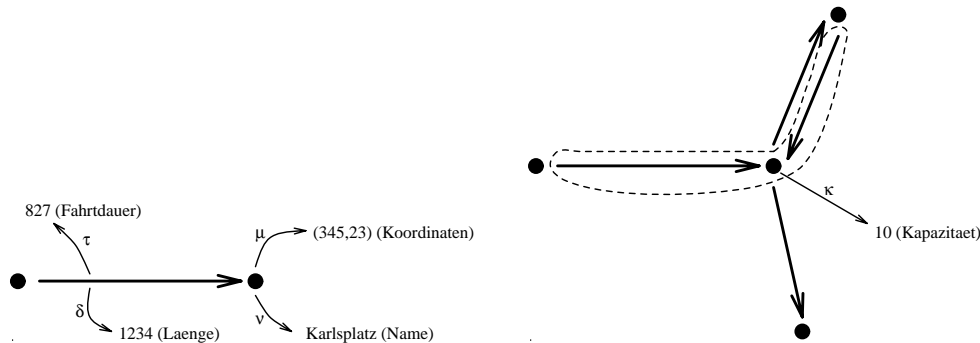


Abbildung 3.10: Verdeutlichung von Definition 3.11

Bemerkungen:

1. Ein Ort modelliert einen Aufenthaltsraum für Verkehrsobjekte. Die Koordinaten eines Ortes dienen der Darstellung und lokalisieren einen Punkt des Aufenthaltsraums. Ein Ort hat eine Kapazität, die die Menge der Verkehrsobjekte auf diesem Ort beschränkt. Da die Verbindungen keine Kapazität besitzen, wird die Menge der Verkehrsobjekte, die sich auf adjazenten einführenden Verbindungen bewegen können, dem jeweiligen Ort als Kapazität zugeschlagen (siehe Abbildung 3.10 rechts). Dadurch entsteht eine Modellierungsschwäche für Verbindungen, die in beiden Richtungen verwendet werden (siehe auch Verklemmungen in 4.2.1).
2. Verbindungen sind mit Fahrtdauer und Länge beschriftet. In meinem Modell wird tatsächlich zunächst nur die Fahrtdauer verwendet.

3.2.2 Verkehrsobjekte

Die beweglichen Einheiten einer Verkehrssimulation sind die Verkehrsobjekte. Sie bewegen sich in einem Verkehrsnetz und benötigen daher eine Routenbeschreibung über der jeweiligen Ortsmenge. Zu jedem Routenpunkt der Ortsbeschreibung haben Verkehrsobjekte eine Planankunftszeit, eine Planabfahrtszeit

und eine Mindestwartezeit. Weiterhin sind sie mit einem Namen, einer Art und einer Mitnahmekapazität beschriftet.

Definition 3.12 Verkehrsobjekt

$o = (R, n, a, m, \gamma, \zeta, \eta)$ ist **Verkehrsobjekt** über der Ortsmenge V : \Leftrightarrow

- $R \in \mathfrak{R}(V)$ ist eine Routenbeschreibung
- $n \in N$ ist ein Name
- $a \in \{\text{person, fahrplanmittel}\}$ ist die **Art** des Verkehrsobjekts.
- $m \in \mathbb{N}$ ist die **Mitnahmekapazität** und bezieht sich auf die Anzahl der Verkehrsobjekte, die mitgenommen werden können. Es gilt:
 $a = \text{person} \Rightarrow m = 0$
(Personen besitzen keine Mitnahmekapazität)
- $\gamma : I(R) \rightarrow \mathbb{N}$ bildet die Indexmenge der Routenbeschreibung auf die **Planankunftszeit** am zugehörigen Routenpunkt ab.
- $\zeta : I(R) \rightarrow \mathbb{N}$ bildet die Indexmenge der Routenbeschreibung auf die **Planabfahrtszeit** am zugehörigen Routenpunkt ab.
- $\eta : I(R) \rightarrow \mathbb{N}$ bildet die Indexmenge der Routenbeschreibung auf die **Mindestwartezeit** am zugehörigen Routenpunkt ab.

$\Omega(V) = \{o \mid o \text{ ist Verkehrsobjekt über der Ortsmenge } V\}$ heißt **Menge aller Verkehrsobjekte** über der Ortsmenge V .

Die Planankunftszeit beschreibt den spätesten Zeitpunkt, zu dem der entsprechende Routenpunkt erreicht werden soll und wird in der Simulation nur beim Eintritt in des Verkehrsnetz verwendet. Im weiteren Verlauf der Simulation kann die Einhaltung der Planankunftszeit nicht garantiert werden. Hingegen kann die Einhaltung der Planabfahrtszeit als frühester Abfahrtszeitpunkt ebenso wie die Einhaltung der Mindestwartedauer garantiert werden. Dies ist im Modell der Fall.

Verkehrsobjekte der Art „person“ (**Personen**) modellieren natürliche Personen. Verkehrsobjekte der Art „fahrplanmittel“ (**Fahrplanmittel**) modellieren öffentliche Verkehrsmittel, die eine bestimmte Zahl von Personen befördern (mitnehmen) können.

Definition 3.12 garantiert nicht, daß die Planankunfts- und -abfahrtszeiten plausibel sind. Insbesondere kann ein späterer Routenpunkt eine frühere Planankunfts- oder -abfahrtszeit haben als ein früherer bzw. die Planabfahrtszeit kann vor der Planankunftszeit liegen. Die folgende Tabelle gibt ein Beispiel für plausible γ , ζ und η :

x	1	2	3
$\pi(o.R, x)$	a	c	d
$o.\gamma(x)$	2	6	10
$o.\zeta(x)$	4	9	13
$o.\eta(x)$	1	2	1

Das Verkehrsobjekt o soll also am 2. Routenpunkt (Ort c) zum Zeitpunkt 6 ankommen, frühestens zum Zeitpunkt 9 abfahren und mindestens zwei Zeiteinheiten warten.

Planankunftszeit, Planabfahrtszeit und Mindestwartezeit sind auch bei Personen sinnvoll, um z.B. den Zeitverbrauch für eine Tätigkeit oder eine bestimmte Aufenthaltsdauer zu modellieren.

3.2.3 Zustände von Verkehrsobjekten

Im folgenden werden nun Zustände von Verkehrsobjekten definiert. Im Zustand eines Verkehrsobjektes ist nur das Vorankommen bezüglich der Routenbeschreibung, nicht aber die Position im Verkehrsnetz gespeichert. Daher kann der Zustand von Verkehrsobjekten unabhängig vom Verkehrsnetz definiert werden.

Der Zustand enthält den zuletzt erreichten Index der Routenbeschreibung sowie den bisherigen Fahrtverlauf in Form tatsächlicher Ankunfts- und Abfahrtszeiten. Bei Fahrplanmitteln gehört auch die Mitnahmezahl bei Abfahrt vom jeweiligen Routenpunkt zum Fahrtverlauf.

Definition 3.13 Zustand von Verkehrsobjekten

$\psi = (i, \alpha, \beta, \varrho)$ heißt **Zustand** von Verkehrsobjekt $o = (R, n, a, m, \gamma, \zeta, \eta) : \Leftrightarrow$

- $i \in I(R)$ ist der **Index** des letzten erreichten Routenpunktes.
- $\alpha : I(R) \rightarrow \mathbf{N}$ bildet die Indexmenge der Routenbeschreibung auf die (tatsächliche) **Ankunftszeit** am zugehörigen Routenpunkt ab.
- $\beta : I(R) \rightarrow \mathbf{N}$ bildet die Indexmenge der Routenbeschreibung auf die (tatsächliche) **Abfahrtszeit** am zugehörigen Routenpunkt ab.
- $\varrho : I(R) \rightarrow \mathbf{N}$ bildet die Indexmenge der Routenbeschreibung auf die (tatsächliche) **Mitnahmezahl** ab dem zugehörigen Routenpunkt ab. Es gilt: $a = \text{person} \Rightarrow \varrho(x) \equiv 0$
(Personen nehmen keine anderen Verkehrsobjekte mit).

$\Psi(o) = \{\psi \mid \psi \text{ ist Zustand von Verkehrsobjekt } o\}$ heißt **Menge aller Zustände** von Verkehrsobjekt o .

$\Psi(V) = \{\psi \mid \exists o \in \Omega(V) : \psi \text{ ist Zustand von Verkehrsobjekt } o\}$ heißt **Menge aller Zustände** von Verkehrsobjekten über der Ortsmenge V .

Der Index zeigt an, wie weit das Verkehrsobjekt auf seiner Routenbeschreibung schon gekommen ist. Ankunftszeit, Abfahrtszeit und Mitnahmezahl je Routenpunkt müssen bis zu diesem Index dem tatsächlichen Fahrtverlauf entsprechen.

3.2.4 Übergänge auf Zuständen

Der Zustand eines Verkehrsobjektes ändert sich im Laufe einer Simulation. Der zuletzt erreichte Routenpunkt wird aktualisiert, wenn der nächste erreicht ist.

Außerdem werden die Ankunfts- und Abfahrtszeiten sowie die Mitnahmezahlen an den jeweiligen Routenpunkten gesetzt. Im Rahmen einer Simulation sollen diese Änderungen einem klaren Protokoll folgen. Beim Eintritt in das Verkehrsnetz wird der Zustand mit dem Eintrittszeitpunkt initialisiert. Bei der Ankunft am nächsten Routenpunkt wird die Ankunftszeit gesetzt und der Index aktualisiert. Bei der Ankunft an einem anderen Ort als dem nächsten Routenpunkt ändert sich der Zustand nicht. Bei der Abfahrt vom aktuellen Routenpunkt wird die Abfahrtszeit und die Mitnahmezahl gesetzt. Bei der Abfahrt von einem anderen Ort ändert sich der Zustand nicht.

Definition 3.14 Übergänge von Zuständen

Auf der Zustandsmenge $\Psi(o)$ eines Verkehrsobjekts $o \in \Omega(V)$ sind folgende Übergänge definiert:

- $Init_o : \mathbb{N} \rightarrow \Psi(o)$ durch $Init_o(\bar{t}) = (i', \alpha', \beta', \varrho')$ mit

$$i' = 1$$

$$\alpha'(x) = \begin{cases} \bar{t} & x = 1 \\ 0 & x \in I(o.R) \setminus \{1\} \end{cases}$$

$$\beta'(x) = 0 \text{ für alle } x \in I(o.R)$$

$$\varrho'(x) = 0 \text{ für alle } x \in I(o.R).$$

(Initialisiert den Zustand des Verkehrsobjekts o , dabei wird die Ankunftszeit am ersten Routenpunkt auf \bar{t} gesetzt.)

- $An_o : \Psi(o) \times V \times \mathbb{N} \rightarrow \Psi(o)$ durch $An_o((i, \alpha, \beta, \varrho), v, \bar{t}) = (i', \alpha', \beta, \varrho)$ mit
 1. **Fall:** $\pi(o.R, i + 1) = v$ (nächster Routenpunkt erreicht)

$$i' = i + 1$$

$$\alpha'(x) = \begin{cases} \bar{t} & x = i + 1 \\ \alpha(x) & x \in I(o.R) \setminus \{i + 1\} \end{cases}$$
 (Index erhöhen und die Ankunftszeit setzen)
 2. **Fall:** $\pi(o.R, i + 1) \neq v$ (nächster Routenpunkt noch nicht erreicht)

$$i' = i$$

$$\alpha'(x) = \alpha(x) \text{ für alle } x \in I(o.R)$$
 (keine Änderung des Zustandes).

(Erhöht den Index und setzt die Abfahrtszeit am (neuen) aktuellen Routenpunkt auf \bar{t} , falls der nächste Routenpunkt dem Ort v entspricht.)

- $Ab_o : \Psi(o) \times V \times \mathbb{N} \times \mathbb{N} \rightarrow \Psi(o)$ durch $Ab_o((i, \alpha, \beta, \varrho), v, \bar{t}, n) = (i, \alpha, \beta', \varrho')$ mit
 1. **Fall:** $\pi(o.R, i) = v$ (Routenpunkt noch aktuell)

$$\beta'(x) = \begin{cases} \bar{t} & x = i \\ \beta(x) & x \in I(o.R) \setminus \{i\} \end{cases}$$

$$\varrho'(x) = \begin{cases} n & x = i \wedge o.a = \text{fahrplanmittel} \\ 0 & x = i \wedge o.a = \text{person} \\ \varrho(x) & x \in I(o.R) \setminus \{i\} \end{cases}$$
 (Abfahrtszeit und Mitnahmezahl setzen)

2. **Fall:** $\pi(o.R, i) \neq v$ (Routenpunkt schon verlassen)
 $\beta'(x) = \beta(x)$ für alle $x \in I(o.R)$
 $\varrho'(x) = \varrho(x)$ für alle $x \in I(o.R)$
 (keine Änderung des Zustandes).

(Setzt Abfahrtszeit und Mitnahmezahl am aktuellen Routenpunkt auf \bar{t} bzw. i , falls der aktuellen Routenpunkt mit dem Ort v übereinstimmt.)

Beispiel 3.3 :

Ein Verkehrsobjekt $o = ([a, c, d], "S1", fahrplanmittel, 100, \gamma, \zeta, \eta)$ mit

x	1	2	3
$o.\gamma(x)$	2	6	10
$o.\zeta(x)$	4	9	13
$o.\eta(x)$	1	2	1

tritt zum Zeitpunkt 1 in das Verkehrsnetz ein, wird initialisiert und bekommt den Zustand $\psi_1 = Init_o(1) = (1, \alpha_1, \beta_1, \varrho_1)$ mit

x	1	2	3
$\alpha_1(x)$	1	0	0
$\beta_1(x)$	0	0	0
$\varrho_1(x)$	0	0	0

Der Index **1** ist fettgedruckt, um anzudeuten, welches der aktuelle Index des Zustandes ist. Der Zustand sollte nun kein weiteres Mal initialisiert werden. Als nächstes verläßt das Verkehrsobjekt vielleicht den Routenpunkt a zum Zeitpunkt 4 und bekommt den Zustand $\psi_2 = Ab_o(\psi_1, a, 4, 20) = (1, \alpha_2, \beta_2, \varrho_2)$ mit

x	1	2	3
$\alpha_2(x)$	1	0	0
$\beta_2(x)$	4	0	0
$\varrho_2(x)$	20	0	0

Durch diesen Zustandsübergang wurde die Abfahrtszeit (4) und die Mitnahmezahl (20) am aktuellen Routenpunkt (a) gesetzt. Dies kann selbstverständlich auch mehrmals geschehen. Erst die Ankunft am nächsten Routenpunkt (c) setzt den Index hoch. Dies kann z.B. so aussehen: $\psi_3 = An_o(\psi_2, c, 7) = (2, \alpha_3, \beta_3, \varrho_3)$ mit

	x	1	2	3		x	1	2	3		
$\psi_3:$	$\alpha_3(x)$	1	7	0	\Rightarrow	$Ab_o(\psi_3, c, 9, 30)$	$\psi_4:$	$\alpha_4(x)$	1	7	0
	$\beta_3(x)$	4	0	0				$\beta_4(x)$	4	9	0
	$\varrho_3(x)$	20	0	0				$\varrho_4(x)$	20	30	0

Nächster Routenpunkt ist nun d . Jede Ankunft an einem anderen Ort ändert den Zustand nicht.

Es gibt auch ein klares Protokoll für die Informationen, die die Simulation vom Zustand eines Verkehrsobjekts benötigt. Die Abfahrzeit am zuletzt erreichten Routenpunkt berechnet sich dabei als Maximum von

- (tatsächlicher) Ankunftszeit + Mindestwartezeit am Routenpunkt und
- Planabfahrtszeit am Routenpunkt.

Der nächste Routenpunkt ergibt sich aus dem Index des Zustandes. Dann gibt es noch eine Abbildung, die prüft, ob ein vorgegebener Routenpunkt noch angesteuert werden soll oder nicht.

Definition 3.15

Auf der Zustandsmenge $\Psi(o)$ eines Verkehrsobjekts $o \in \Omega(V)$ gibt es folgende Hilfsabbildungen:

- *Abfahrzeit* : $\Psi(o) \rightarrow \mathbb{N}$ durch

$$\text{Abfahrzeit}(i, \alpha, \beta, \varrho) = \begin{cases} \alpha(i) + o.\eta(i) & \alpha(i) + o.\eta(i) > o.\zeta(i) \\ o.\zeta(i) & \text{sonst} \end{cases}$$
- *Naechst* : $\Psi(o) \rightarrow V$ durch $\text{Naechst}(i, \alpha, \beta, \varrho) = \pi(o.R, i + 1)$
- *Geht_nach* : $\Psi(o) \times V \rightarrow V \cup \{u\}$ durch

$$\text{Geht_nach}((o, i, \alpha, \beta, \varrho), v) = \begin{cases} v & \exists j \in \{i+1, \dots, \max(o.R)\} : v = \pi(o.R, j) \\ u & \text{sonst} \end{cases}$$

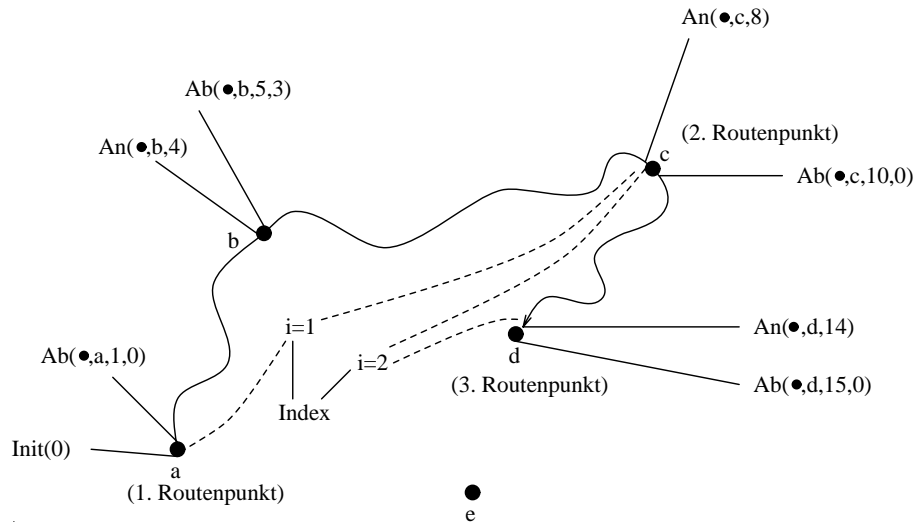


Abbildung 3.11: Bewegung eines Verkehrsobjekts in einer Ortsmenge

In Abbildung 3.11 ist dargestellt, was mit einem Verkehrsobjekt bei der Simulation passieren soll. In diesem Fall ist die Ortsmenge $V = \{a,b,c,d,e\}$. Das Verkehrsobjekt hat die Routenbeschreibung $R = [a,c,d]$. Tatsächlich

bewegt sich das Verkehrsobjekt entlang der als Schlangenlinie dargestellten Route. Dabei erfährt es an jedem Ort die angegebenen Zustandsübergänge. Am Ort b verändert sich der Zustand jedoch nicht, weil b nicht nächster Routenpunkt des Verkehrsobjekts ist. Als Endzustand ergibt sich $(3, \alpha, \beta, \varrho)$ mit

x	1	2	3
$\alpha(x)$	0	8	14
$\beta(x)$	1	10	15
$\varrho(x)$	0	0	0

3.2.5 Mitnahmesystem

Die Bewegung von Personen in Fahrplanmitteln erfordert einen Bezug zwischen den jeweiligen Verkehrsnetzen. Die Verkehrsnetze werden völlig getrennt modelliert und sind nur dadurch verbunden, daß jedem Haltepunkt im Verkehrsnetz für Fahrplanmittel eine Haltestelle im Verkehrsnetz für Personen zugeordnet ist. An dieser Stelle wird ein Mitnahmesystem mit genau einem Mitnahmenetz spezifiziert. Natürlich kann man Mitnahmesysteme mit mehreren Mitnahmenetzen analog definieren.

Definition 3.16 Mitnahmesystem

$\Sigma = (\vartheta_p, P, \vartheta_f, F, h)$ heißt **Mitnahmesystem** $:\Leftrightarrow$

- $\vartheta_p = (V_p, E_p, T_p, \mu_p, \nu_p, \kappa_p, \delta_p, \tau_p, \lambda_p)$ und $\vartheta_f = (V_f, E_f, T_f, \mu_f, \nu_f, \kappa_f, \delta_f, \tau_f, \lambda_f)$ sind Verkehrsnetze mit $V_p \cap V_f = \emptyset$. ϑ_p heißt **Fußwegenetz** und ϑ_f heißt **ÖV-Netz**.
- $P \in \Omega(V_p)$ mit $\forall p \in P : p.a = \text{person}$ ist eine endliche Menge von **Personen** über der Ortsmenge V_p .
- $F \in \Omega(V_f)$ mit $\forall f \in F : f.a = \text{fahrplanmittel}$ ist eine endliche Menge von **Fahrplanmitteln** über der Ortsmenge V_f .
- alle Verkehrsobjekte haben einen eindeutigen Namen:
 $\forall o \in P \cup F : o_1 \neq o_2 \Rightarrow o_1.n \neq o_2.n$.
- $h : V_f \rightarrow V_p \cup \{u\}$ bildet die **Haltepunktmenge** $(\{v \in V_f \mid h(v) \in V_p\})$ auf die **Haltestellenmenge** $(\{v \in V_p \mid \exists w \in V_f : h(w) = v\})$ ab. h heißt **Mitnahmebeziehung** zwischen ÖV-Netz ϑ_f und Fußwegenetz ϑ_p .

In einem Mitnahmesystem gibt es folgende gemeinsame Beschriftungen:

- $\mu : V_p \cup V_f \rightarrow \mathbb{R} \times \mathbb{R}$ mit $\mu(x) = \begin{cases} \mu_p(x) & x \in V_p \\ \mu_f(x) & x \in V_f \end{cases}$
für die Koordinaten

- $\nu : V_p \cup V_f \rightarrow \mathbb{N}$ mit $\nu(x) = \begin{cases} \nu_p(x) & x \in V_p \\ \nu_f(x) & x \in V_f \end{cases}$
für die Namen.
- $\kappa : V_p \cup V_f \rightarrow \mathbb{N} \cup \{\infty\}$ mit $\kappa(x) = \begin{cases} \kappa_p(x) & x \in V_p \\ \kappa_f(x) & x \in V_f \end{cases}$
für die Kapazität.
- $\delta : E_p \cup E_f \rightarrow \mathbb{R}_+$ mit $\delta(x) = \begin{cases} \delta_p(x) & x \in E_p \\ \delta_f(x) & x \in E_f \end{cases}$
für die Länge.
- $\tau : E_p \cup E_f \rightarrow \mathbb{N}$ mit $\tau(x) = \begin{cases} \tau_p(x) & x \in E_p \\ \tau_f(x) & x \in E_f \end{cases}$
für die Fahrtdauer.
- $\lambda : T_p \cup T_f \rightarrow V_p \cup V_f$ mit $\lambda(x) = \begin{cases} \lambda_p(x) & x \in T_p \\ \lambda_f(x) & x \in T_f \end{cases}$
für die Routentabelle.

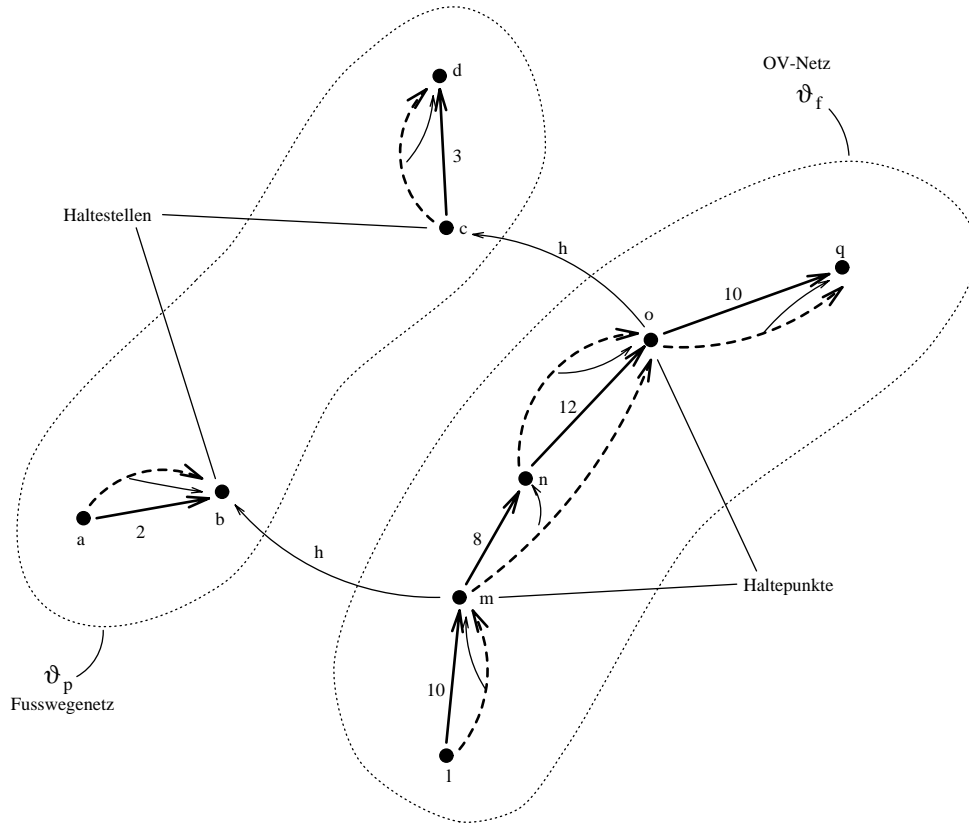


Abbildung 3.12: Ein Mitnahmesystem

In Abbildung 3.12 sind Fußwegenetz ϑ_p , ÖV-Netz ϑ_f und die Mitnahmebeziehung h zwischen den Netzen exemplarisch dargestellt. Verbindungen sind durch

dicke Pfeile, Routentabelleneinträge durch dicke, gestrichelte Pfeile und Abbildungen durch dünne Pfeile dargestellt. Die Verbindungen sind dabei nur mit der jeweiligen Fahrtdauer beschriftet.

3.2.6 Konfiguration von Mitnahmesystemen

Die Konfiguration eines Mitnahmesystems enthält für alle Verkehrsobjekte, die sich in ihrem Verkehrsnetz bewegen, einen Zustand, einen Status, einen Ereigniszeitpunkt und den aktuellen Ort. Der Status gibt dabei an, ob sich das Verkehrsobjekt gerade am aktuellen Ort aufhält oder sich auf diesen zubewegt. Der Ereigniszeitpunkt ist der früheste Zeitpunkt, an dem für dieses Objekt ein neues Ereignis eintritt.

Definition 3.17 Konfiguration

$\Phi = (\varphi, \sigma, \chi, M, t)$ heißt **Konfiguration** von Mitnahmesystem $\Sigma = (\vartheta_p, P, \vartheta_f, F, h)$ mit $\vartheta_p = (V_p, E_p, T_p, \mu_p, \nu_p, \kappa_p, \delta_p, \tau_p, \lambda_p)$ und $\vartheta_f = (V_f, E_f, T_f, \mu_f, \nu_f, \kappa_f, \delta_f, \tau_f, \lambda_f) :\Leftrightarrow$

- $\varphi : P \cup F \rightarrow \Psi(V_p) \cup \Psi(V_f)$ mit $\forall o \in P \cup F : \varphi(o) \in \Psi(o)$ ordnet den Verkehrsobjekten einen Zustand zu.
- $\sigma : P \cup F \rightarrow \{\text{wartend, unterwegs}\}$ ordnet den Verkehrsobjekten ihren aktuellen **Status** zu.
- $\chi : P \cup F \rightarrow \mathbb{N}$ bildet die Verkehrsobjekte auf den nächsten **Ereigniszeitpunkt** ab.
- $M : (P \rightarrow V_p \cup F \cup \{\text{quelle, senke}\}) \cup (F \rightarrow V_f \cup \{\text{quelle, senke}\})$ ordnet jedem Verkehrsobjekt seinen aktuellen Ort bzw. die ausgezeichneten Werte *quelle* oder *senke* zu. Der aktuelle Ort kann dabei nicht in einem fremden Verkehrsnetz liegen. Der aktuelle Ort von Personen kann auch ein Fahrplanmittel sein.
- $t \in \mathbb{N}$ ist der Zeitpunkt der Konfiguration.

Alle Verkehrsobjekte können sich außer bei Orten auch in *quelle* oder *senke* befinden. Diese ausgezeichneten Aufenthaltsorte geben an, daß sich das Verkehrsobjekt noch nicht bzw. nicht mehr im Verkehrsnetz bewegt. Personen können sich außerdem in Fahrplanmitteln befinden.

Ist der Status eines Verkehrsobjekts *unterwegs*, so bewegt es sich gerade auf den aktuellen Ort zu. Ist der Status *wartend*, so befindet es sich an einem Ort, in einem anderen Verkehrsobjekt oder außerhalb des Verkehrsnetzes. Ein Verkehrsobjekt o_1 mit $\sigma(o_1) = \text{unterwegs}$, $\chi(o_1) = 234$ und $M(o_1) = a \in V$ bewegt sich gerade auf den Ort a zu und wird dort zum Zeitpunkt $T = 234$ ankommen. Ein Verkehrsobjekt o_2 mit $\sigma(o_2) = \text{wartend}$, $\chi(o_2) = 237$ und $M(o_2) = f \in F$ befindet sich gerade im Fahrplanmittel f . Der nächste Ereigniszeitpunkt ist in diesem Fall bedeutungslos, da der Ausstieg von der Bewegung des Fahrplanmittels f abhängt.

3.2.7 Simulation

Die Simulation von Verkehr in einem Mitnahmesystem besteht aus einer Folge von Konfigurationen. Der Übergang auf eine Folgekonfiguration ist dabei entweder ein Zeitschritt oder eines der folgenden Ereignisse:

- Austritt eines Verkehrsobjekts aus dem Verkehrsnetz (Ziel)
- Eintritt eines Verkehrsobjekts in das Verkehrsnetz (Start)
- Abfahrt eines Verkehrsobjekts von einem Ort (Abfahrt)
- Ankunft eines Verkehrsobjekts an einem Ort (Ankunft)
- Einstieg einer Person in ein Fahrplanmittel (Einstieg)
- Ausstieg einer Person aus einem Fahrplanmittel (Ausstieg)

Zu jedem dieser Ereignisse läßt sich die zugehörige Ereignismenge bestimmen (Name in Klammern). Dabei werden alle Ereignisse berechnet, die überhaupt jetzt auftreten können, unabhängig davon, ob sie vielleicht miteinander in Konflikt stehen.

Die Bewegung zu einem Ort ist nur möglich, wenn dessen Kapazität nicht ausgeschöpft ist. Ähnliches gilt auch für das Einsteigen in Fahrplanmittel. Um die Ereignismengen leichter berechnen zu können, hier zunächst einige Hilfsmengen auf Konfigurationen.

Definition 3.18

Zu einer Konfiguration $\Phi = (\varphi, \sigma, \chi, M, t)$ von Mitnahmesystem $\Sigma = (\vartheta_p, P, \vartheta_f, F, h)$ mit $\vartheta_p = (V_p, E_p, T_p, \mu_p, \nu_p, \kappa_p, \delta_p, \tau_p, \lambda_p)$ und $\vartheta_f = (V_f, E_f, T_f, \mu_f, \nu_f, \kappa_f, \delta_f, \tau_f, \lambda_f)$ gibt es folgende **Hilfsmengen**:

1. $Voll(\Phi, \Sigma) = \{v \in V_p \cup V_f \mid \#M^{-1}(v) = \kappa(v)\}$
(Hilfsmenge der Orte, deren Kapazität ausgeschöpft ist.)
2. $Besetzt(\Phi, \Sigma) = \{f \in F \mid \#M^{-1}(f) = f.m\}$
(Hilfsmenge der Fahrplanmittel, deren Mitnahmekapazität ausgeschöpft ist.)

Die eigentlichen Ereignismengen können nun leichter berechnet werden.

Definition 3.19

Zu einer Konfiguration $\Phi = (\varphi, \sigma, \chi, M, t)$ von Mitnahmesystem $\Sigma = (\vartheta_p, P, \vartheta_f, F, h)$ mit $\vartheta_p = (V_p, E_p, T_p, \mu_p, \nu_p, \kappa_p, \delta_p, \tau_p, \lambda_p)$ und $\vartheta_f = (V_f, E_f, T_f, \mu_f, \nu_f, \kappa_f, \delta_f, \tau_f, \lambda_f)$ gibt es folgende **Ereignismengen**:

1. $Ziel(\Phi, \Sigma) = \{o \in P \cup F \mid \sigma(o) = \text{wartend} \wedge \chi(o) \leq t \wedge \varphi(o).i = \max(o.R)\}$
(Ereignismenge der Verkehrsobjekte, die am Ende ihrer Routenbeschreibung angelangt sind und dort ihre Wartezeit beendet haben)

2. $Start(\Phi, \Sigma) = \{o \in P \cup F \mid M(o) = \text{quelle} \wedge \chi(o) \leq t \wedge \pi(o.R, 1) \notin Voll(\Phi, \Sigma)\}$
(Ereignismenge der Verkehrsobjekte, die noch in der Quelle sind, ihren Eintrittszeitpunkt erreicht haben und in das Verkehrsnetz eintreten können)
3. $Abfahrt(\Phi, \Sigma) = \{o \in P \cup F \mid \sigma(o) = \text{wartend} \wedge \chi(o) \leq t \wedge \exists_{(v_1, v_2) \in T} : (v_1 = M(o) \wedge v_2 = Naechst(\varphi(o)) \wedge \lambda(v_1, v_2) \notin Voll(\Phi, \Sigma))\}$
(Ereignismenge der Verkehrsobjekte, die an einem Ort sind, ihre Wartezeit beendet haben und sich nun mit Hilfe der Routentabelle weiterbewegen können)
4. $Ankunft(\Phi, \Sigma) = \{o \in P \cup F \mid \sigma(o) = \text{unterwegs} \wedge \chi(o) \leq t\}$
(Ereignismenge der Verkehrsobjekte, die gerade zu einem Ort unterwegs sind und ihre Fahrtdauer beendet haben)
5. $Einstieg(\Phi, \Sigma) = \{(p, f) \in P \times F \mid \sigma(p) = \text{wartend} \wedge \sigma(f) = \text{wartend} \wedge \chi(p) \leq t \wedge f \notin Besetzt(\Phi, \Sigma) \wedge \exists_{v \in V_f} : (v = M(f) \wedge h(v) = M(p)) \wedge \exists_{w \in V_f} : (w = Geht_nach(\varphi(f), w) \wedge h(w) = Naechst(\varphi(p)))\}$
(Ereignismenge der Personen–Fahrplanmittel–Tupel, bei denen die Person einsteigen kann und das Fahrplanmittel den nächsten Routenpunkt der Person ansteuern wird)
6. $Ausstieg(\Phi, \Sigma) = \{(p, f) \in P \times F \mid \sigma(p) = \text{wartend} \wedge \sigma(f) = \text{wartend} \wedge M(p) = f \wedge \exists v \in V_f : (v = M(f) \wedge h(v) = Naechst(\varphi(p)) \wedge h(v) \notin Voll(\Phi, \Sigma))\}$
(Ereignismenge der Personen–Fahrplanmittel–Tupel, bei denen die Person aus einem Fahrplanmittel aussteigen kann und ihren nächsten Routenpunkt durch Mitfahrt erreicht hat)

Diese sechs Ereignismengen sind nicht konfliktfrei. Die Durchführung eines Ereignisses kann andere Ereignisse verschwinden lassen oder erzeugen. Die Ankunft eines Fahrplanmittels an einem Ort erzeugt z.B. neue Ereignisse in *Ausstieg*. Die Abfahrt eines beliebigen Verkehrsobjekts von einem Ort kann neue Abfahrten auslösen, da die Kapazität des Ortes vielleicht erschöpft war.

Die Ereignismengen sind auch nicht unbedingt disjunkt. Eine Person kann z.B. zugleich in den Ereignismengen *Abfahrt* und *Einstieg* sein, wenn ihr nächster Routenpunkt sowohl mit einem passenden Fahrplanmittel als auch direkt über einen Routentableneintrag erreicht werden kann.

Eine Folgekonfiguration entsteht deshalb durch ein einziges Ereignis. Danach werden die Ereignismengen neu berechnet. Die Zeit wird erst fortgeschaltet, wenn alle Ereignismengen leer sind (in Anlehnung an die Simulation von THOR-Netzen, vgl. 5.2).

Die Auswahl eines Ereignisses geschieht innerhalb von Ereignismengen mit einer Auswahlfunktion *First*, die auf jeder Ereignismenge $E \in \{\text{Ziel}, \text{Start}, \text{Abfahrt}, \text{Ankunft}, \text{Einstieg}, \text{Ausstieg}\}$ mit $First(E) = e \in E$ definiert ist.

Die Auswahl der Ereignismenge erfolgt durch eine Fallunterscheidung in der Definition der Folgekonfiguration. Die sieben disjunkten Fälle für die sechs Ereignismengen und das Fortschalten der Zeit stellen die Behandlung der Ereignismengen in folgender Reihenfolge sicher:

- | | |
|------------|-------------|
| 1. Ziel | 4. Ankunft |
| 2. Start | 5. Einstieg |
| 3. Abfahrt | 6. Ausstieg |

Erst wenn alle Ereignismengen leer sind, wird die Zeit fortgeschaltet.

Definition 3.20

$\Phi' = (\varphi', \sigma', \chi', M', t')$ heißt **Folgekonfiguration** von $\Phi = (\varphi, \sigma, \chi, M, t)$ auf dem Mitnahmesystem $\Sigma = (\vartheta_p, P, \vartheta_f, F, h) : \Leftrightarrow$

1. **Fall:** $Ziel(\Phi, \Sigma) \neq \emptyset$. $o = First(Ziel(\Phi, \Sigma))$.

$$\begin{aligned} \varphi'(x) &= \varphi(x) \text{ für alle } x \in P \cup F \\ \sigma'(x) &= \sigma(x) \text{ für alle } x \in P \cup F \\ \chi'(x) &= \chi(x) \text{ für alle } x \in P \cup F \\ M'(x) &= \begin{cases} \text{senke} & x = o \\ M(x) & x \in (P \cup F) \setminus \{o\} \end{cases} \\ t' &= t \end{aligned}$$

2. **Fall:** $Ziel(\Phi, \Sigma) = \emptyset \wedge Start(\Phi, \Sigma) \neq \emptyset$. $o = First(Start(\Phi, \Sigma))$.

$$\begin{aligned} \varphi'(x) &= \begin{cases} Init_o(t) & x = o \\ \varphi(x) & x \in (P \cup F) \setminus \{o\} \end{cases} \\ \sigma'(x) &= \begin{cases} \text{wartend} & x = o \\ \sigma(x) & x \in (P \cup F) \setminus \{o\} \end{cases} \\ \chi'(x) &= \begin{cases} Abfahrzeit(\varphi'(x)) & x = o \\ \chi(x) & x \in (P \cup F) \setminus \{o\} \end{cases} \\ M'(x) &= \begin{cases} \pi(x.R, 1) & x = o \\ M(x) & x \in (P \cup F) \setminus \{o\} \end{cases} \\ t' &= t \end{aligned}$$

3. **Fall:** $Ziel(\Phi, \Sigma) = \emptyset \wedge Start(\Phi, \Sigma) = \emptyset \wedge Abfahrt(\Phi, \Sigma) \neq \emptyset$.
 $o = First(Abfahrt(\Phi, \Sigma))$.

$$\begin{aligned} \varphi'(x) &= \begin{cases} Ab_o(\varphi(x), M(x), t, \#M^{-1}(o)) & x = o \\ \varphi(x) & x \in (P \cup F) \setminus \{o\} \end{cases} \\ \sigma'(x) &= \begin{cases} \text{unterwegs} & x = o \\ \sigma(x) & x \in (P \cup F) \setminus \{o\} \end{cases} \\ \chi'(x) &= \begin{cases} t + \tau(M(x), M'(x)) & x = o \\ \chi(x) & x \in (P \cup F) \setminus \{o\} \end{cases} \\ M'(x) &= \begin{cases} \lambda(M(x), Naechst(\varphi(o))) & x = o \\ M(x) & x \in (P \cup F) \setminus \{o\} \end{cases} \end{aligned}$$

$$t' = t$$

4. **Fall:** $Ziel(\Phi, \Sigma) = \emptyset \wedge Start(\Phi, \Sigma) = \emptyset \wedge Abfahrt(\Phi, \Sigma) = \emptyset \wedge Ankunft(\Phi, \Sigma) \neq \emptyset$. $o = First(Aankunft(\Phi, \Sigma))$.

$$\varphi'(x) = \begin{cases} An_o(\varphi(x), M(x), t) & x = o \\ \varphi(x) & x \in (P \cup F) \setminus \{o\} \end{cases}$$

$$\sigma'(x) = \begin{cases} \text{wartend} & x = o \\ \sigma(x) & x \in (P \cup F) \setminus \{o\} \end{cases}$$

$$\chi'(x) = \begin{cases} Abfahrzeit(\varphi'(x)) & x = o \\ \chi(x) & x \in (P \cup F) \setminus \{o\} \end{cases}$$

$$M'(x) = M(x) \text{ für alle } x \in P \cup F$$

$$t' = t$$

5. **Fall:** $Ziel(\Phi, \Sigma) = \emptyset \wedge Start(\Phi, \Sigma) = \emptyset \wedge Abfahrt(\Phi, \Sigma) = \emptyset \wedge Ankunft(\Phi, \Sigma) = \emptyset \wedge Einstieg(\Phi, \Sigma) \neq \emptyset$. $(p, f) = First(Einstieg(\Phi, \Sigma))$.

$$\varphi'(x) = \begin{cases} Ab_o(\varphi(x), M(x), t, 0) & x = p \\ \varphi(x) & x \in (P \cup F) \setminus \{p\} \end{cases}$$

$$\sigma'(x) = \sigma(x) \text{ für alle } x \in P \cup F$$

$$\chi'(x) = \chi(x) \text{ für alle } x \in P \cup F$$

$$M'(x) = \begin{cases} f & x = p \\ M(x) & x \in (P \cup F) \setminus \{p\} \end{cases}$$

$$t' = t$$

6. **Fall:** $Ziel(\Phi, \Sigma) = \emptyset \wedge Start(\Phi, \Sigma) = \emptyset \wedge Abfahrt(\Phi, \Sigma) = \emptyset \wedge Ankunft(\Phi, \Sigma) = \emptyset \wedge Einstieg(\Phi, \Sigma) = \emptyset \wedge Ausstieg(\Phi, \Sigma) \neq \emptyset$. $(p, f) = First(Ausstieg(\Phi, \Sigma))$

$$\varphi'(x) = \begin{cases} An_o(\varphi(x), h(M(f)), t) & x = p \\ \varphi(x) & x \in (P \cup F) \setminus \{p\} \end{cases}$$

$$\sigma'(x) = \sigma(x) \text{ für alle } x \in P \cup F$$

$$\chi'(x) = \begin{cases} Abfahrzeit(\varphi'(x)) & x = p \\ \chi(x) & x \in (P \cup F) \setminus \{p\} \end{cases}$$

$$M'(x) = \begin{cases} h(M(f)) & x = p \\ M(x) & x \in (P \cup F) \setminus \{p\} \end{cases}$$

$$t' = t$$

7. **Fall:** $Ziel(\Phi, \Sigma) = \emptyset \wedge Start(\Phi, \Sigma) = \emptyset \wedge Abfahrt(\Phi, \Sigma) = \emptyset \wedge Ankunft(\Phi, \Sigma) = \emptyset \wedge Einstieg(\Phi, \Sigma) = \emptyset \wedge Ausstieg(\Phi, \Sigma) = \emptyset$.

$$\varphi'(x) = \varphi(x) \text{ für alle } x \in P \cup F$$

$$\sigma'(x) = \sigma(x) \text{ für alle } x \in P \cup F$$

$$\chi'(x) = \chi(x) \text{ für alle } x \in P \cup F$$

$$M'(x) = M(x) \text{ für alle } x \in P \cup F$$

$$t' = t + 1$$

Für eine Simulation benötigen wir nun noch Start- und Zielkonfigurationen.

Definition 3.21

Eine Konfiguration $\Phi = (\varphi, \sigma, \chi, M, t)$ von Mitnahmesystem $\Sigma = (\vartheta_p, P, \vartheta_f, F, h)$ heißt **Startkonfiguration** $:\Leftrightarrow \forall_{o \in P \cup F} : M(o) = \text{quelle} \wedge \chi(o) = o.\gamma(1)$.

Eine Konfiguration $\Phi = (\varphi, \sigma, \chi, M, t)$ von Mitnahmesystem $\Sigma = (\vartheta_p, P, \vartheta_f, F, h)$ heißt **Zielkonfiguration** $:\Leftrightarrow$

1. $Ziel(\Phi, \Sigma) = \emptyset \wedge Start(\Phi, \Sigma) = \emptyset \wedge Abfahrt(\Phi, \Sigma) = \emptyset \wedge Ankunft(\Phi, \Sigma) = \emptyset \wedge Einstieg(\Phi, \Sigma) = \emptyset \wedge Ausstieg(\Phi, \Sigma) = \emptyset$.
2. $\forall_{o \in P \cup F} : \chi(o) \leq t$.

Definition 3.22

$\Phi_1 \Phi_2 \dots \Phi_n$ heißt **Simulation** des Mitnahmesystems $\Sigma = (\vartheta_p, P, \vartheta_f, F, h)$ $:\Leftrightarrow$

1. $\forall_{1 \leq i \leq n} : \Phi_i$ ist Konfiguration von Σ .
2. Φ_1 ist Startkonfiguration von Σ .
3. $\forall_{1 \leq i < n} : \Phi_{i+1}$ ist Folgekonfiguration von Φ_i auf Σ .

Definition 3.23

Eine Simulation $\Phi_1 \Phi_2 \dots \Phi_n$ von Mitnahmesystem $\Sigma = (\vartheta_p, P, \vartheta_f, F, h)$ heißt **vollständig** $:\Leftrightarrow \Phi_n$ ist Zielkonfiguration von Σ .

Beispiel 3.4 : Ein Mitnahmesystem $\Sigma = (\vartheta_p, \{p\}, \vartheta_f, \{f\}, h)$ besteht aus den Verkehrsnetzen und der Mitnahmebeziehung aus Abbildung 3.12 sowie der Person $p = ([a, b, c, d], \text{“Max“}, \text{person}, 0, \gamma_p, \zeta_p, \eta_p)$ und dem Fahrplanmittel $f = ([l, m, o, q], \text{“S1“}, \text{fahrplanmittel}, 10, \gamma_f, \zeta_f, \eta_f)$ mit

x	1	2	3	4	x	1	2	3	4
$\pi([a, b, c, d], x)$	a	b	c	d	$\pi([l, m, o, q], x)$	l	m	o	q
$\gamma_p(x)$	1	0	0	0	$\gamma_f(x)$	1	12	33	45
$\zeta_p(x)$	0	0	0	0	$\zeta_f(x)$	2	14	35	47
$\eta_p(x)$	0	0	0	0	$\eta_f(x)$	1	1	2	1

Für diese beiden Verkehrsobjekte ist der Ablauf einer Simulation in Tabelle 3.13 dargestellt. Vom Zustand der Objekte ist dabei nur der aktuelle Index $\varphi.i$ sichtbar. In der Spalte „Ereignis“ werden die Ereignismengen aufgezählt, in denen das jeweilige Verkehrsobjekt auftaucht.

Erläuterungen zum Simulationsbeispiel (Tabelle 3.13):

- Unterstrichen sind die Ereignisse, die für die Folgekonfiguration geschaltet werden. Bei Φ_1 liegen zwar beide Verkehrsobjekte in *Start* ($Start = \{p, f\}$), aber p wird ausgewählt ($p = First(Start)$). Sind alle Ereignismengen leer bzw. liegt keines der Verkehrsobjekte in irgendeiner Ereignismenge, so ist die Zeit unterstrichen und wird für den Folgezustand inkrementiert. Dabei werden im Beispiel gegebenenfalls mehrere Zeitschritte zusammengefaßt (z.B. $\Phi_9 - \Phi_{17}$).

t	$\varphi(p).i$	$\sigma(p)$	$\chi(p)$	$M(p)$	Ereignis	$\varphi(f).i$	$\sigma(f)$	$\chi(f)$	$M(f)$	Ereignis
Φ_0	$\frac{0}{1}$	1	wart.	1	quelle	1	wart.	1	quelle	–
Φ_1	$\frac{1}{1}$	1	wart.	1	quelle	1	wart.	1	quelle	<u>Start</u>
Φ_2	$\frac{1}{1}$	1	wart.	1	a	1	wart.	1	quelle	<u>Start</u>
Φ_3	$\frac{1}{1}$	1	wart.	1	a	1	wart.	2	l	–
Φ_4	$\frac{1}{1}$	1	fahr.	3	b	1	wart.	2	l	–
Φ_5	$\frac{2}{2}$	1	fahr.	3	b	1	wart.	2	l	<u>Abfahrt</u>
Φ_6	$\frac{2}{3}$	1	fahr.	3	b	1	fahr.	12	m	–
Φ_7	$\frac{3}{3}$	1	fahr.	3	b	1	fahr.	12	m	–
Φ_8	$\frac{3}{2}$	2	wart.	3	b	1	fahr.	12	m	–
Φ_{17}	$\frac{12}{12}$	2	wart.	3	b	1	fahr.	12	m	<u>Ankunft</u>
Φ_{18}	$\frac{12}{2}$	2	wart.	3	b	2	wart.	14	m	–
Φ_{19}	$\frac{12}{2}$	2	wart.	3	f	2	wart.	14	m	–
Φ_{21}	$\frac{14}{14}$	2	wart.	3	f	2	wart.	14	m	<u>Abfahrt</u>
Φ_{22}	$\frac{14}{2}$	2	wart.	3	f	2	fahr.	22	n	–
Φ_{30}	$\frac{22}{22}$	2	wart.	3	f	2	fahr.	22	n	<u>Ankunft</u>
Φ_{31}	$\frac{22}{2}$	2	wart.	3	f	2	wart.	14	n	<u>Abfahrt</u>
Φ_{32}	$\frac{22}{2}$	2	wart.	3	f	2	fahr.	34	o	–
Φ_{44}	$\frac{34}{34}$	2	wart.	3	f	2	fahr.	34	o	<u>Ankunft</u>
Φ_{45}	$\frac{34}{2}$	2	wart.	3	f	3	wart.	36	o	–
Φ_{46}	$\frac{34}{3}$	3	wart.	34	c	3	wart.	36	o	–
Φ_{47}	$\frac{34}{3}$	3	fahr.	37	d	3	wart.	36	o	–
Φ_{49}	$\frac{36}{36}$	3	fahr.	37	d	3	wart.	36	o	<u>Abfahrt</u>
Φ_{50}	$\frac{36}{3}$	3	fahr.	37	d	3	fahr.	46	q	–
Φ_{51}	$\frac{37}{37}$	3	fahr.	37	d	3	fahr.	46	q	–
Φ_{52}	$\frac{37}{4}$	4	wart.	37	d	3	fahr.	46	q	–
Φ_{53}	$\frac{37}{4}$	4	wart.	37	senke	3	fahr.	46	q	–
Φ_{62}	$\frac{46}{46}$	4	wart.	37	senke	3	fahr.	46	q	<u>Ankunft</u>
Φ_{63}	$\frac{46}{4}$	4	wart.	37	senke	4	wart.	47	q	–
Φ_{64}	$\frac{47}{47}$	4	wart.	37	senke	4	wart.	47	q	<u>Ziel</u>
Φ_{65}	$\frac{47}{4}$	4	wart.	37	senke	4	wart.	47	senke	–

Abbildung 3.13: Simulationsbeispiel

- Sind mehrere Ereignismengen nichtleer (wie z.B. in Konfiguration Φ_2), so legt die Fallunterscheidung in Definition 3.20 die Ereignismenge fest, die zuerst bearbeitet wird (*Abfahrt* wird erst bearbeitet (3. Fall), wenn *Start* und *Ziel* leer sind).
- Beim Einstieg einer Person p (z.B. bei $\Phi_{18} \rightarrow \Phi_{19}$) wird der Zeitpunkt $\chi(p)$ des nächsten Ereignisses nicht neu berechnet, da er nur von der Bewegung des mitnehmenden Fahrplanmittels abhängt. *Ausstieg* ist daher auch die einzige Ereignismenge, in die ein Verkehrsobjekt aufgenommen wird, ohne diesen Zeitpunkt zu berücksichtigen.

In der Zielkonfiguration Φ_{65} sehen die Zustände von p und f so aus:

	x		1	2	3	4		x		1	2	3	4
$\varphi(p)$:	$\alpha(x)$		1	3	34	37	$\varphi(f)$:	$\alpha(x)$		1	12	34	46
	$\beta(x)$		1	12	34	37		$\beta(x)$		2	14	36	47
	$\rho(x)$		0	0	0	0		$\rho(x)$		0	1	0	0

starke Vermutung: Für alle Startkonfigurationen Φ_1 von Mitnahmesystem Σ existiert ein $n \in \mathbb{N}$, so daß $\Phi_1 \Phi_2 \dots \Phi_n$ vollständige Simulation von Σ , d.h. die Simulation terminiert mit jeder Startkonfiguration.

Kapitel 4

Modellierung am Beispiel Stuttgart

Um ein etwas größeres Beispiel zu bekommen, wurde auf eine feinkörnige Modellierung der Verkehrswege zunächst verzichtet. Stattdessen wurde bei der Modellierung des öffentlichen Verkehrs in Stuttgart eine Art Top-Down-Ansatz verfolgt. Die Verkehrsangebotsdaten des Instituts für Eisenbahn- und Verkehrswesen (IEUV) dienen als Grundlage der Modellierung, Verfeinerungen wurden nur dort vorgenommen, wo dies für die Simulation unerlässlich ist. Dadurch steht ein großer Teil der benötigten Daten in elektronischer Form zur Verfügung und kann automatisch weiterverarbeitet werden. Folgende Daten wurden dabei verwendet (siehe auch Kapitel 2):

- Der Untersuchungsraum $U = \{cell_1, \dots, cell_{u_max}\}$ und
- das Liniennetz $LNet = (S, L, LCon, name, koord, art, takt, abstand)$ zum Untersuchungsraum U .

Das resultierende Fußwegenetz besteht nur aus den Haltestellen als Orten und wird in dieser Arbeit ohne Verbindungen modelliert. Die Bewegung von Personen erfolgt also nur durch Mitnahme in Fahrplanmitteln. Die Modellierung vom Fußwegenetz wird in Abschnitt 4.1 beschrieben.

Die resultierenden ÖV-Netze für U- und S-Bahn bestehen aus

- Haltepunkten (Orten) und
- den notwendigen Verbindungen, um zu gewährleisten, daß die Routenbeschreibungen aller Linien auch Routen im jeweiligen Netz sind.

Die Modellierung der S- und U-Bahn-Netze sowie die zugehörigen Routenbeschreibungen wird in Abschnitt 4.2 beschrieben. Dabei wird auch die Mitnahmebeziehung h für beide Netze modelliert, die die Verbindung zwischen dem jeweiligen öffentlichen Verkehrsnetz und dem Fußwegenetz herstellt. Weitere öffentliche Netze (z.B. das Straßennetz als Verkehrsnetz für Bus und IV) werden in dieser Arbeit nicht modelliert.

In den Abschnitten 4.4 und 4.5 wird die Modellierung von Fahrplanmitteln und Personen zu den jeweiligen Netzen beschrieben.

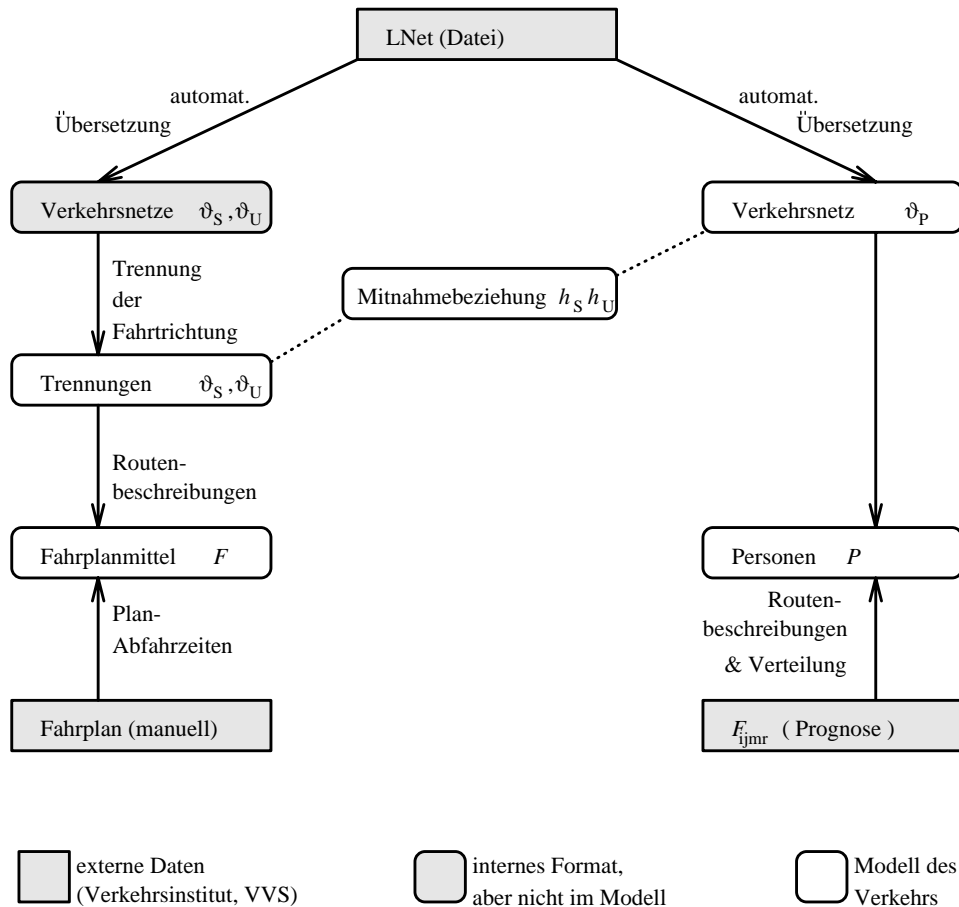


Abbildung 4.1: grobe Modellierung aus den Daten des IEUV und des VVS

Für diese Modellierung der Verkehrsobjekte sind neben den oben genannten Daten selbst bei größter Modellierung noch folgende Informationen nötig:

- Fahrpläne von S- und U-Bahn (zur Modellierung der Fahrplanmittel) und
- die Verkehrsnachfrage F_{ijmr} (zur Modellierung der Personen).

Eine Übersicht der verwendeten Daten und der daraus gewonnenen Informationen bietet Abbildung 4.1. Die Fahrpläne liegen beim IEUV nicht vor. Deshalb wurden die öffentlichen Fahrpläne des Verkehrs- und Tarifverbunds Stuttgart (VVS) von Hand¹ eingegeben.

¹Gegen Ende meiner Arbeit erhielt ich die Zusage vom VVS, mir diese Daten elektronisch zur Verfügung zu stellen. Leider sind sie nicht rechtzeitig eingetroffen. Die Eingabe von Hand geschah natürlich nur ungefähr und ohne Berücksichtigung von Ausnahmen vom normalen Takt.

In dieser Arbeit werden die Routentabellen immer so einfach wie möglich gehalten. Routentabellen verweisen daher nur auf unmittelbare Nachbarorte. Solche Routentabellen nenne ich elementar.

Definition 4.1

Sei (V, E) ein Verkehrsgraph. (T, λ) heißt **elementare Routentabelle zu (V, E)** : $\Leftrightarrow T = E \wedge \forall_{(v_1, v_2) \in T} : \lambda(v_1, v_2) = v_2$.

Bemerkung: Verkehrsnetze, deren Routentabellen elementar zum Verkehrsgraph (V, E) sind, kennen keinen Unterschied zwischen günstigen Routenbeschreibungen und Routen. (vgl. Def. 3.8)

4.1 Das Fußwegenetz

Das Verkehrsnetz für Fußgänger besteht für diese konkrete Modellierung nur aus den Haltestellen, weil diese direkt aus dem Liniennetz übernommen werden können.

Definition 4.2

Sei $LNet = (S, L, LCon, name, koord, art, takt, abstand)$ Liniennetz zum Untersuchungsraum $U = \{cell_1, \dots, cell_{u-max}\}$. Das Verkehrsnetz $\vartheta_P = (V_P, E_P, T_P, \mu_P, \nu_P, \kappa_P, \delta_P, \tau_P, \lambda_P)$ heißt **Fußwegenetz zum Liniennetz $LNet$** : \Leftrightarrow

- $V_P = S$.
- $T_P = E_P = \emptyset$.
- $\forall_{v \in V_P} : \mu_P(v) = koord(v)$.
- $\forall_{v \in V_P} : \nu_P(v) = name(v)$.
- $\kappa_P \equiv \infty$.

Bemerkungen:

- δ_P, τ_P und λ_P müssen nicht festgelegt werden, weil ihre Urbildmengen (E_P bzw. T_P) leer sind.
- Bewegungen im Fußwegenetz können nur mit Hilfe eines Mitnahmenetzes durchgeführt werden.

4.2 Verkehrsnetze für U- und S-Bahn

In diesem Abschnitt wird die Modellierung der Schienenverkehrsnetze aus dem Liniennetz dargestellt. Alle Definitionen heben dabei auf das U-Bahn-Verkehrsnetz ab, selbstverständlich wird das S-Bahn-Verkehrsnetz vollkommen analog modelliert.

In Teil 4.2.1 wird zunächst ein Verkehrsnetz ohne Trennung der Fahrtrichtungen definiert. Leider ist es jedoch kaum möglich, die Bewegung von Schienenfahrzeugen ohne eine Trennung der Fahrtrichtungen zu modellieren (Teil 4.2.2). In den Teilen 4.2.3 bis 4.2.6 wird daher eine künstliche Trennung der Fahrtrichtungen formalisiert, die zu einer bestimmten Menge von Routenbeschreibungen paßt. Diese Formalisierung wird nicht für Verkehrsnetze, sondern nur für Verkehrsgraphen durchgeführt. In Teil 4.2.7 werden die Ergebnisse dann auf die Berechnung eines getrennten Verkehrsnetzes aus einem Liniennetz übertragen.

Die Formalisierung von getrennten Verkehrsgraphen basiert dabei auf einer Ortsmenge V und einer Routenbeschreibungsmenge $\mathcal{R} \in \mathfrak{R}(V)$. Diese Basis nenne ich im folgenden Routennetz.

Definition 4.3 Routennetz

Sei V eine Ortsmenge und $\mathcal{R} \in \mathfrak{R}(V)$ eine Menge von Routenbeschreibungen. (V, \mathcal{R}) heißt **Routennetz**.

4.2.1 Verkehrsnetz ohne Trennung der Fahrtrichtungen

Da das Liniennetz direkt vorgibt, welche Verbindungen im entsprechenden Verkehrsnetz existieren müssen, definiere ich im folgenden nun die Umsetzung von dem Liniennetz auf das U-Bahn-Netz.

Definition 4.4

Sei $LNet = (S, L, LCon, name, koord, art, takt, abstand)$ ein Liniennetz zum Untersuchungsraum $U = \{cell_1, \dots, cell_{u_max}\}$. Das Verkehrsnetz $\vartheta_U = (V_U, E_U, T_U, \mu_U, \nu_U, \kappa_U, \delta_U, \tau_U, \lambda_U)$ heißt **U-Bahn-Verkehrsnetz zum Liniennetz $LNet$** $:\Leftrightarrow$

- $V_U = \{s \in S \mid \exists_{(R,time,dist) \in L} : s \text{ liegt in } R \wedge art(R, time, dist) = \text{U-Bahn}\}$.
- $E_U = \{(v_1, v_2) \in V_U \mid \exists_{(R,time,dist) \in \mathcal{R}_{bahn}} : art(R, time, dist) = \text{U-Bahn} \wedge \exists_{1 \leq j < max(R)} : \pi(R, j) = v_1 \wedge \pi(R, j + 1) = v_2\}$.
- (T_U, λ_U) ist elementare Routentabelle zu (V_U, E_U) .
- $\forall_{v \in V_U} : \mu_U(v) = koord(v)$.
- $\forall_{v \in V_U} : \nu_U(v) = name(v)$.
- $\kappa_U \equiv 2$.
- $\forall_{(v_1, v_2) \in E_U} \exists_{(R,time,dist) \in L} \exists_{1 \leq j < max(R)} : \pi(R, j) = v_1 \wedge \pi(R, j + 1) = v_2 \wedge art(R, time, dist) = \text{U-Bahn} \wedge \delta_U(v_1, v_2) = dist(j) \wedge \tau_U(v_1, v_2) = time(j)$.

Bemerkungen:

1. Je zwei Routenpunkte einer beliebigen Routenbeschreibung von U-Bahn-Linien erzeugen also eine Verbindung. Da eine Verbindung auch mehrmals in verschiedenen Routenbeschreibungen vorkommen kann, muß für die Bestimmung von Länge und Fahrzeit einer Verbindung eine bestimmte Routenbeschreibung herangezogen werden.
2. Die Kapazität eines Ortes für U-Bahnen wurde auf zwei festgelegt, damit sich Verkehrsobjekte in unterschiedlichen Richtungen bewegen können, ohne sich zu blockieren.

Beispiel 4.1 : Gegeben ist ein Liniennetz $LNet = (S, L, LCon, name, koord, art, takt, abstand)$ mit der Haltestellenmenge $S = \{\text{Vaihinger Str., SSB-Zentrum, Möhringen, Leinfelden}\}$ und der Linienmenge $L = \{(R_1, time_1, dist_1), (R_2, time_2, dist_2), (R_3, time_3, dist_3), (R_4, time_4, dist_4)\}$ mit:

j	1	2	3
$\pi(R_1, j)$	SSB-Zentrum	Vaihinger Str.	Möhringen
$dist_1(j)$	-
$time_1(j)$	120	60	-
$\pi(R_2, j)$	Möhringen	Vaihinger Str.	SSB-Zentrum
$dist_2(j)$	-
$time_2(j)$	60	120	-
$\pi(R_3, j)$	Leinfelden	Vaihinger Str.	Möhringen
$dist_3(j)$	-
$time_3(j)$	480	61	-
$\pi(R_4, j)$	Möhringen	Vaihinger Str.	Leinfelden
$dist_4(j)$	-
$time_4(j)$	61	480	-

Es gibt also offensichtlich eine Linie von „SSB-Zentrum“ über „Vaihinger Str.“ nach „Möhringen“ sowie eine Linie von „Leinfelden“ über „Vaihinger Str.“ nach „Möhringen“ und die jeweiligen Rückrichtungen. Daraus wird ein Verkehrsnetz gewonnen, das die entsprechenden Verbindungen besitzt und in Abbildung 4.2 dargestellt ist. Die Fahrzeit von 60 Sekunden zwischen „Vaihinger Str.“ und „Möhringen“ entstand dabei durch zufällige Auswahl der Routenbeschreibungen R_1 und R_2 als maßgebliche Linien.

Modelliert man die Bewegung der entsprechenden Fahrplanmittel in diesem Verkehrsnetz, so kommt es sehr leicht zu Verklemmungen. Befinden sich z.B. zwei Verkehrsobjekte auf b , die a als nächsten Routenpunkt besitzen und außerdem zwei Verkehrsobjekte auf a , die b als nächsten Routenpunkt besitzen, so kann kein Verkehrsobjekt abfahren.

Bei Schienennetzen ist es daher wünschenswert, je ein Gleis für jede Fahrtrichtung zu modellieren, so daß alle Orte mit Kapazität eins versehen werden können und der Verkehr auf diesen Gleisen als Warteschlange modelliert werden kann.

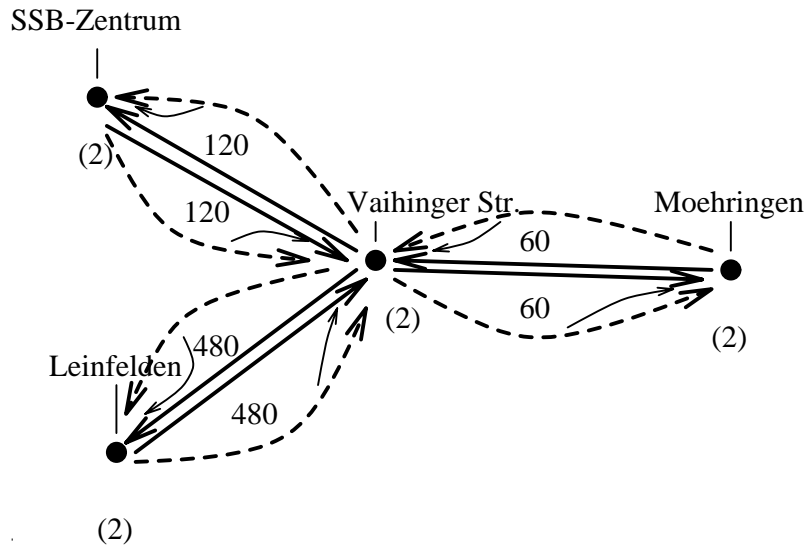


Abbildung 4.2: Ausschnitt eines U-Bahn-Verkehrsnetzes

4.2.2 Trennung der Fahrtrichtungen (Problem)

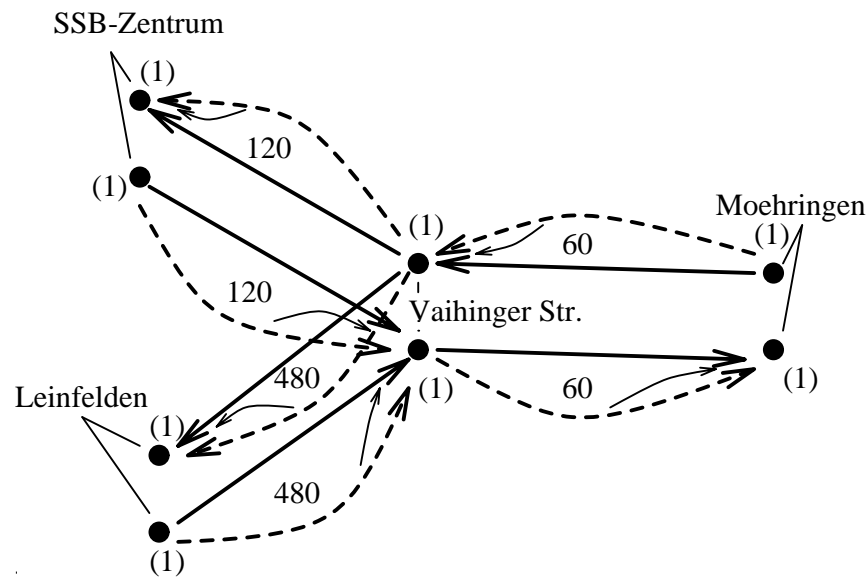


Abbildung 4.3: Intuitive Trennung der Fahrtrichtungen

Intuitiv sieht das Verkehrsnetz für das Beispiel dann wie in Abbildung 4.3 aus. Dadurch zerfällt jede der ursprünglichen Haltestellen in zwei verschiedene Orte (einen für jede Fahrtrichtung). Der wesentliche Vorteil liegt jedoch darin, daß die Abbiegebeziehungen mitmodelliert werden. Im einfachen Fall (Abbildung 4.2) kann sich ein Verkehrsobjekt auch von „Leinfelden“ über „Vaihinger Str.“ nach „SSB-Zentrum“ bewegen, obwohl dies aufgrund des Liniennetzes nicht möglich

ist. Bei getrennten Fahrtrichtungen (Abbildung 4.3) ist diese zusätzliche Abbiegemöglichkeit nicht mehr vorhanden.

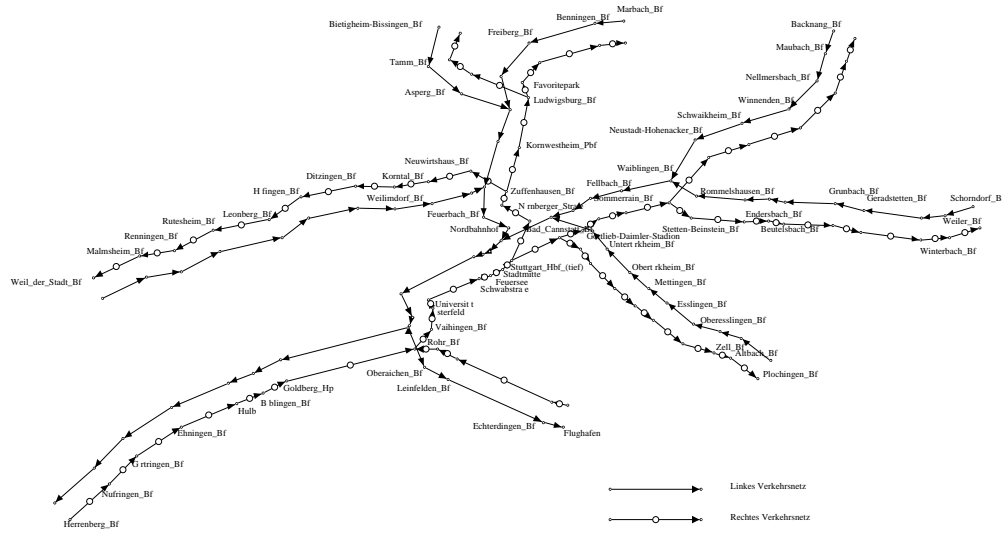


Abbildung 4.4: Zweigeteiltes S-Bahn-Netz in Stuttgart

Diese einfache Zweiteilung genügt im Fall des S-Bahn-Verkehrsnetzes von Stuttgart. Wie man in Abbildung 4.4 sieht, zerfällt der Verkehrsgraph vollständig in zwei Teile. Dies geht jedoch im allgemeinen nicht. Wollte man z.B. in Abbildung 4.3 noch eine Abbiegebeziehung von „Leinfelden“ über „Vaihinger Str.“ nach „SSB-Zentrum“ einführen, so kommt man nicht umhin, die beiden Verkehrsgraphen miteinander zu verbinden.

4.2.3 Verfeinerung von Orten

Im folgenden wird nun beschrieben, wie eine bestehende Ortsmenge vervielfacht (verfeinert) werden kann, was dabei mit den Routenbeschreibungen geschieht und welche Anforderungen ich an den resultierenden Verkehrsgraphen zur Vermeidung von Verklemmungen stelle.

Eine Verfeinerung von Orten beschreibt die folgende

Definition 4.5

(\bar{V}, f, V) heißt **Verfeinerung von Orten** $:\Leftrightarrow$

- \bar{V}, V sind Ortsmengen,
- $f : \bar{V} \rightarrow V$ ordnet jedem Ort aus \bar{V} einen Ort aus V zu.

Bemerkung: Alle Orte $\bar{v} \in \bar{V}$, die von der Abbildung f auf den Ort $v \in V$ abgebildet werden, stellen die Aufspaltung des Ortes v in mehrere Orte dar.

Routenbeschreibungen, die auf einer Ortsmenge definiert sind, benötigen bei einer Aufspaltung dieser Ortsmenge eine korrespondierende Routenbeschreibung. Daher die folgende

Definition 4.6

Sei (\bar{V}, f, V) eine Verfeinerung von Orten. Sei $\bar{R} = [\bar{r}_1, \dots, \bar{r}_n] \in \mathfrak{R}(\bar{V})$. Dann **korrespondiert \bar{R} bezüglich f zu $R \in \mathfrak{R}(V) : \Leftrightarrow R = [f(\bar{r}_1), \dots, f(\bar{r}_n)]$.**

Schreibweise: $\bar{R} \sim_f R$.

Bemerkung: Zwei Routenbeschreibungen korrespondieren bezüglich einer Abbildung genau dann, wenn die erste Routenbeschreibung durch die Abbildung in die andere Routenbeschreibung überführt werden kann.

Hat man ein Routennetz (V, \mathcal{R}) und leitet man aus der Aufspaltung von Orten einen Verkehrsgraphen ab, so sollen natürlich alle Routenbeschreibungen $R \in \mathcal{R}$ eine korrespondierende Routenbeschreibung haben, die in der Kantenmenge liegt. Für Verkehrsgraphen, die von Verfeinerungen von Orten abgeleitet sind, definieren wir diese Eigenschaft als „passend“.

Definition 4.7

Sei (\bar{V}, f, V) eine Verfeinerung von Orten. $\mathcal{R} \subset \mathfrak{R}(V)$ eine Routenbeschreibungsmenge. Der Verkehrsgraph (\bar{V}, \bar{E}) **paßt zu \mathcal{R} bezüglich f : \Leftrightarrow**

$$\forall R \in \mathcal{R} \exists \bar{R} \in \mathfrak{R}(\bar{V}) : (\bar{R} \sim_f R) \wedge (\bar{R} \text{ liegt in } \bar{E}).$$

Der eigentliche Sinn einer Verfeinerung besteht darin, genügend Orte für jeden ursprünglichen Ort einzuführen, so daß die Verbindungsrichtungen getrennt sind. Diese Trennung kann für Verkehrsgraphen, die von Verfeinerungen von Orten abgeleitet sind, auf mehrere Arten ausgedrückt werden. Meine Vorstellung verlangt, daß die Verbindung unterschiedlicher Richtung zwischen zwei Orten vollständig getrennt sind. Deshalb

Definition 4.8

Sei (\bar{V}, f, V) eine Verfeinerung von Orten. Der Verkehrsgraph (\bar{V}, \bar{E}) ist **getrennt bezüglich V : \Leftrightarrow**

- $\forall_{(\bar{v}_1, \bar{v}_2) \in \bar{E}} : f(\bar{v}_1) \neq f(\bar{v}_2)$
- $\forall_{(\bar{v}_1, \bar{v}_2), (\bar{v}_3, \bar{v}_4) \in \bar{E}} : ((f(\bar{v}_1) = f(\bar{v}_4) \wedge (f(\bar{v}_2) = f(\bar{v}_3))) \Rightarrow \bar{v}_1 \neq \bar{v}_4 \wedge \bar{v}_2 \neq \bar{v}_3)$

Beispiel 4.2 : Die vorangegangenen Definitionen werden in Abbildung 4.5 illustriert. (\bar{V}, f, V) ist dabei eine Verfeinerung von Orten und $\mathcal{R} = \{[A, B, C]\} \in \mathfrak{R}(V)$ eine einelementige Routenmenge. Die Verkehrsgraphen (\bar{V}, \bar{E}) in a) und c) sind getrennt bezüglich V . In b) ist (\bar{V}, \bar{E}) nicht getrennt, weil (q, p) und (p, r) in \bar{E} . Die Verkehrsgraphen (\bar{V}, \bar{E}) in b) und c) passen zu \mathcal{R} bezüglich f . In a) paßt (\bar{V}, \bar{E}) nicht zu \mathcal{R} bezüglich f . Zwar gibt es eine bezüglich f zu $[A, B, C]$ korrespondierende Routenbeschreibung (z.B. $[n, o, q]$), jedoch keine, die gleichzeitig in \bar{E} liegt.

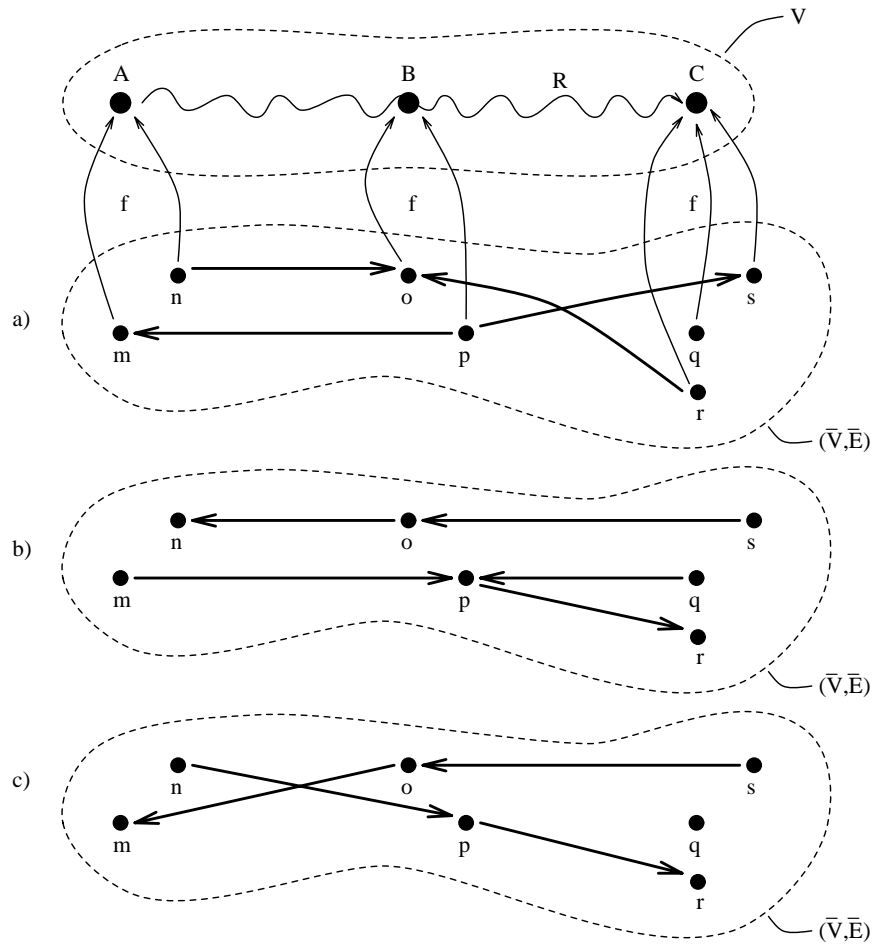


Abbildung 4.5: **a)** getrennt, aber nicht passend **b)** passend, aber nicht getrennt **c)** getrennt und passend

Ich fasse „passend“ und „getrennt“ nun zusammen in

Definition 4.9

Sei (\bar{V}, f, V) eine Verfeinerung von Orten und (V, \mathcal{R}) ein Routennetz. Der Verkehrsgraph (\bar{V}, \bar{E}) ist **getrennter Verkehrsgraph zu (V, \mathcal{R})** $:\Leftrightarrow$

- (\bar{V}, \bar{E}) paßt zu \mathcal{R} bezüglich f und
- (\bar{V}, \bar{E}) ist getrennt bezüglich V .

4.2.4 Abbiegehaltepunkte

Es stellt sich nun die Frage, wie man die Orte am besten aufspaltet. Um Verkehrsgraphen zu erhalten, die zu einem gegebenen Routennetz passen, bietet es sich an, für jede Abbiegemöglichkeit an einem Ort einen eigenen Ort einzuführen.

Beispiel 4.3 : In Abbildung 4.6 (links) führt eine Routenbeschreibung von a

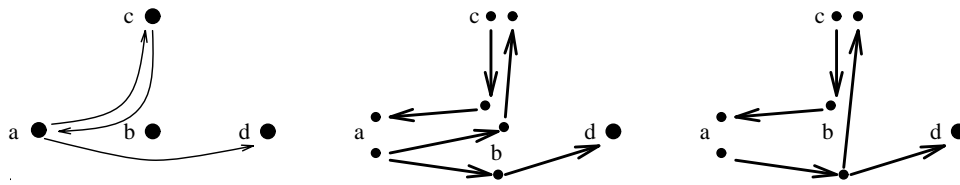


Abbildung 4.6: Beispiel für eine Verfeinerung von Orten

über b nach c , eine andere von a über b nach d und eine dritte von c über b nach a . b kann also in $[a, \mathbf{b}, c]$, $[a, \mathbf{b}, d]$ und $[c, \mathbf{b}, a]$ aufgespalten werden. Auch die Orte a und c sind entsprechend aufgespalten worden. Dies ist in der mittleren Abbildung zusammen mit den entsprechenden Kanten dargestellt. Dabei verkörpern $[a, \mathbf{b}, d]$ und $[c, \mathbf{b}, a]$ unterschiedliche Fahrtrichtungen, während $[a, \mathbf{b}, c]$ und $[a, \mathbf{b}, d]$ wohl auch zu $[a, \mathbf{b}, \{c, d\}]$ verschmolzen werden könnten (linke Abbildung).

Solche Abbiegebeziehungen führe ich allgemein als Abbiegehaltepunkte ein. Ein Abbiegehaltepunkt besteht dabei aus einem Vorbereich als Menge von Orten, einem Haltepunkt und einem Nachbereich als Menge von Orten.

Definition 4.10

Sei V ein Ortsmenge. Die Ortsmenge $AHP(V) = \wp(V) \times V \times \wp(V)$ ist die **Menge aller möglichen Abbiegehaltepunkte in V** . Mit $h = (V_{vor}, v, V_{nach}) \in AHP(V)$ heißt V_{vor} **Vorbereich von h** und V_{nach} **Nachbereich von h** .

Bemerkung: Elemente aus Mengen von Abbiegehaltepunkten werden von nun an häufig mit h bezeichnet. Diese Bezeichnung sollte nicht mit der Mitnahmebeziehung h verwechselt werden.

Durch die Abbildung auf die mittlere Komponente kann V zusammen mit $AHP(V)$ als Verfeinerung von Orten betrachtet werden. Dies ist im folgenden Lemma formuliert.

Lemma 4.1

Sei V ein Ortsmenge und $H \subseteq AHP(V)$. Dann ist (H, f, V) eine Verfeinerung von Orten mit $f : (V_{vor}, v, V_{nach}) \mapsto v$.

Beweis :

folgt direkt aus Definition 4.5 (Verfeinerung von Orten).

q.e.d.

Der Abbiegehaltepunkt $(V_{vor}, v, V_{nach}) \in AHP(V)$ drückt alle Abbiegemöglichkeiten von $v_1 \in V_{vor}$ über v nach $v_2 \in V_{nach}$ aus. Für den Verkehrsgraphen mit der entsprechenden Kantenmenge ergibt sich unmittelbar

Definition 4.11

Sei V ein Ortsmenge und $H \subseteq AHP(V)$. $(H, \bar{E}) = VG(H)$ heißt **assoziierter Verkehrsgraph** zu $H := \{(V_{vor,1}, v_1, V_{nach,1}), (V_{vor,2}, v_2, V_{nach,2})\} \in H \times H \mid v_1 \in V_{vor,2} \wedge v_2 \in V_{nach,1}$.

Zwei Abbiegehaltepunkte $(V_{vor,1}, v_1, V_{nach,1})$ und $(V_{vor,2}, v_2, V_{nach,2})$ werden genau dann verbunden, wenn der zugeordnete Ort v_1 des ersten Abbiegehaltepunkts im Vorbereich des zweiten liegt und der zugeordnete Ort v_2 im Nachbereich des ersten Abbiegehaltepunkts liegt.

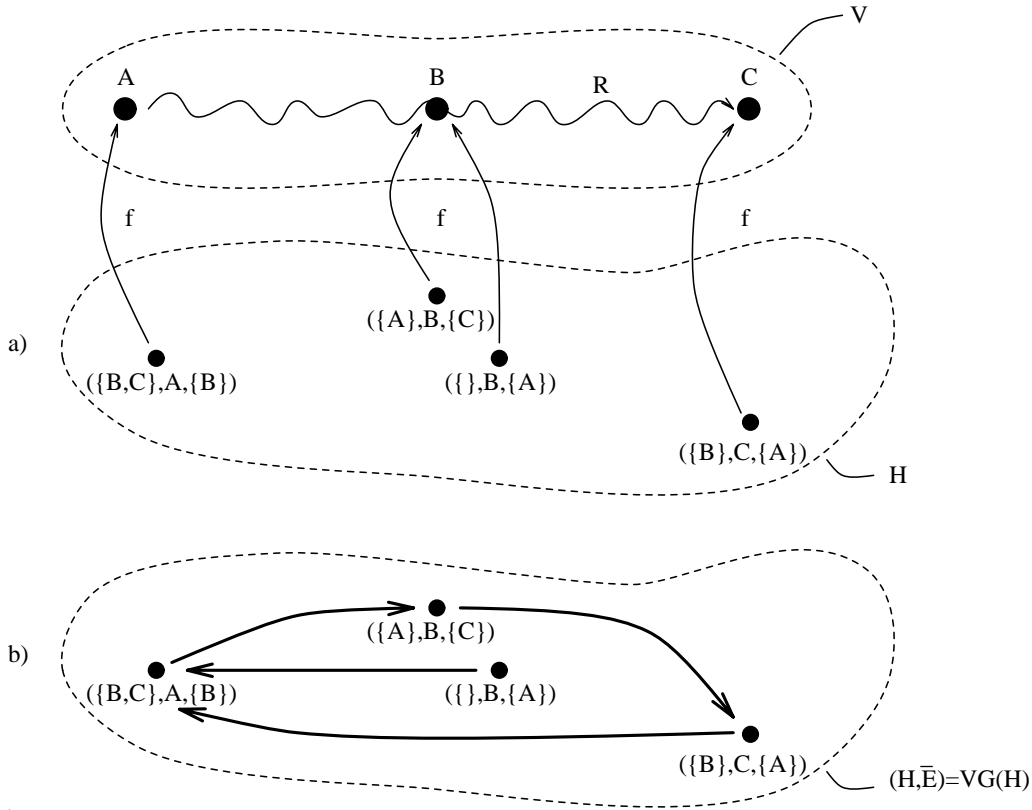


Abbildung 4.7: Abbiegehaltepunkte zu einer Ortsmenge $V = \{A, B, C\}$ und einer Routenmenge $\mathcal{R} = \{[A, B, C]\}$. **a)** Menge von Abbiegehaltepunkten $H = H(V, \mathcal{R})$. **b)** assoziierter Verkehrsgraph (H, \bar{E}) .

In Abbildung 4.7 sind eine Verfeinerung von Orten (H, f, V) und der zu H korrespondierende Verkehrsgraph (H, \bar{E}) abgebildet. Dabei ist $(\{\}, B, \{A\})$ ein Abbiegehaltepunkt, der nur eine auslaufende Kante und keine einlaufende Kante besitzt. Solche Abbiegehaltepunkte beschreiben eine Quellrichtung.

Bevor nun aufgrund eines Routennetzes (V, \mathcal{R}) eine Menge von Abbiegebeziehungen konstruiert wird, deren assoziierter Verkehrsgraph auch getrennter Verkehrsgraph zu (V, \mathcal{R}) ist, sollen zunächst die Bedingungen formuliert werden, unter denen eine Menge von Abbiegebeziehungen überhaupt zu einem assoziierten Verkehrsgraphen führt, der getrennt ist.

Definition 4.12

Sei V ein Ortsmenge und $H \subseteq AHP(V)$.

- H heißt **schleifenfrei** $:\Leftrightarrow \forall (V_{vor}, v, V_{nach}) \in H : v \notin V_{vor} \wedge v \notin V_{nach}$.
- H heißt **wendefrei** $:\Leftrightarrow \forall (V_{vor}, v, V_{nach}) \in H : V_{vor} \cap V_{nach} = \emptyset$.

Satz 4.2

Sei (H, f, V) eine Verfeinerung von Orten mit $H \subseteq AHP(V)$ und $f(V_{vor}, v, V_{nach}) \mapsto v$. Weiterhin sei H schleifen- und wendefrei. Dann ist der assoziierte Verkehrsgraph (H, \bar{E}) getrennt bezüglich V .

Beweis :

Sei $((V_{vor,1}, v_1, V_{nach,1}), (V_{vor,2}, v_2, V_{nach,2})), ((V_{vor,3}, v_3, V_{nach,3}), (V_{vor,4}, v_4, V_{nach,4})) \in \bar{E}$. Zu zeigen ist, daß

1. $f(V_{vor,1}, v_1, V_{nach,1}) \neq f(V_{vor,2}, v_2, V_{nach,2})$
2. $((v_1 = v_4) \wedge (v_2 = v_3)) \Rightarrow (((V_{vor,1}, v_1, V_{nach,1}) \neq (V_{vor,4}, v_4, V_{nach,4})) \wedge ((V_{vor,2}, v_2, V_{nach,2}) \neq (V_{vor,3}, v_3, V_{nach,3})))$

Zu 1. Beweis durch Widerspruch.

Annahme: $f(V_{vor,1}, v_1, V_{nach,1}) = f(V_{vor,2}, v_2, V_{nach,2})$. Also $v_1 = v_2 \in V_{vor,2}$. Widerspruch zur Schleifenfreiheit von H .

Zu 2. Sei $v_1 = v_4$. $V_{vor,2} \cap V_{nach,3} \supseteq \{v_1\}$ wegen $v_1 \in V_{vor,2}$ und $v_1 = v_4 \in V_{nach,3}$. Ich zeige, daß $((V_{vor,2}, v_2, V_{nach,2}) \neq (V_{vor,3}, v_3, V_{nach,3}))$. Beweis durch Widerspruch. Annahme: $((V_{vor,2}, v_2, V_{nach,2}) = (V_{vor,3}, v_3, V_{nach,3}))$. Also $V_{vor,2} = V_{vor,3}$. Damit $V_{vor,3} \cap V_{nach,3} \supseteq \{v_1\} \neq \emptyset$. Widerspruch zur Wendefreiheit von H . Sei nun $v_2 = v_3$. Dann folgt analog, daß $(V_{vor,1}, v_1, V_{nach,1}) \neq (V_{vor,4}, v_4, V_{nach,4})$.

q.e.d.

Spaltet man die Orte eines Routennetzes (V, \mathcal{R}) in eine Menge von Abbiegehaltepunkten H auf, so ist die Schleifen- und Wendefreiheit von H also hinreichendes Kriterium dafür, daß der zu H assoziierte Verkehrsgraph $VG(H)$ getrennt ist bezüglich V . Im nächsten Abschnitt konstruiere ich nun eine Menge von Abbiegehaltepunkten H zum Routennetzes (V, \mathcal{R}) , die zur Routenbeschreibungsmenge \mathcal{R} paßt.

4.2.5 Trennung der Fahrtrichtungen (Lösung)

Zu einem Routennetz definieren wir jetzt Mengen von Abbiegebeziehungen, Senk- und Quellrichtungen sowie Haltepunktbeziehungen als Sonderfälle von Abbiegehaltepunkten.

Definition 4.13 Haltepunktbeziehungen

Sei (V, \mathcal{R}) ein Routennetz. Folgende Mengen sind auf (V, \mathcal{R}) definiert:

1. Menge der **Abbiegebeziehungen** $ABB(V, \mathcal{R}) \subseteq AHP(V)$ mit
 $ABB(V, \mathcal{R}) = \{(\{v_1\}, v, \{v_2\}) \in \wp(V) \times V \times \wp(V) \mid \exists R \in \mathcal{R} \exists j \in \mathbb{N} : \pi(R, j) = v_1 \wedge \pi(R, j+1) = v \wedge \pi(R, j+2) = v_2\}$.
2. Menge der **Senkrichtungen** $SR(V, \mathcal{R}) \subseteq AHP(V)$ mit
 $SR(V, \mathcal{R}) = \{(\{v_1\}, v, \emptyset) \in \wp(V) \times V \times \{\emptyset\} \mid (\exists R \in \mathcal{R} : \pi(R, \max(R) - 1) = v_1 \wedge \pi(R, \max(R)) = v) \wedge (\forall v_2 \in V : (\{v_1\}, v, \{v_2\}) \notin ABB(V))\}$.
3. Menge der **Quellrichtungen** $QR(V, \mathcal{R}) \subseteq AHP(V)$ mit
 $QR(V, \mathcal{R}) = \{(\emptyset, v, \{v_2\}) \in \{\emptyset\} \times V \times \wp(V) \mid (\exists R \in \mathcal{R} : \pi(R, 1) = v \wedge \pi(R, 2) = v_2) \wedge (\forall v_1 \in V : (\{v_1\}, v, \{v_2\}) \notin ABB(V))\}$.
4. Menge der **Haltepunktbeziehungen** $H(V, \mathcal{R}) \subseteq AHP(V)$ mit
 $H(V, \mathcal{R}) = ABB(V, \mathcal{R}) \cup SR(V, \mathcal{R}) \cup QR(V, \mathcal{R})$.

Bemerkungen:

- Die Vor- und Nachbereiche aller Haltepunktbeziehungen sind einelementig oder leer.
- In die Menge der Senkrichtungen sind nur diejenigen Abbiegehaltepunkte mit leerem Nachbereich aufgenommen, für die es keine entsprechende Abbiegebeziehung mit einelementigem Nachbereich gibt.

Beispiel 4.4 : Sei (V, \mathcal{R}) ein Routennetz mit $V = \{a, b, c, d\}$ und einer einelementigen Routenmenge $\mathcal{R} = \{[a, b, c]\}$. Am Ort b gibt es eine Abbiegebeziehung, deswegen $ABB(V, \mathcal{R}) = \{(\{a\}, b, \{c\})\}$. Am Ort a gibt es eine Quellrichtung (insbesondere also keine Abbiegebeziehung), deswegen $QR(V, \mathcal{R}) = \{(\emptyset, a, \{b\})\}$. Am Ort c gibt es eine Senkrichtung (insbesondere also keine Abbiegebeziehung), deswegen $SR(V, \mathcal{R}) = \{(\{b\}, c, \emptyset)\}$.

Diese Konstruktion soll eine Menge von Abbiegehaltepunkten erzeugen, deren assoziierter Verkehrsgraph zur Menge der Routenbeschreibungen paßt. Es gibt jedoch eine kleine Einschränkung. Jede Routenbeschreibung muß mindestens die Länge zwei haben, da aus einelementigen Routenbeschreibungen keine Haltepunktbeziehung konstruiert wird. Diese Eigenschaft nenne ich „Echtheit“.

Definition 4.14

Sei V eine Ortmenge und $\mathcal{R} \subseteq \mathfrak{R}(V)$ eine Menge von Routenbeschreibungen. \mathcal{R} heißt **echt** $:\Leftrightarrow \forall R \in \mathcal{R} : \max(R) \geq 2$.

Satz 4.3

Sei (V, \mathcal{R}) ein Routennetz. Sei (H, f, V) eine Verfeinerung von Orten, wobei $H = H(V, \mathcal{R})$ die Menge von Haltepunktbeziehungen und $f : (V_{vor}, v, V_{nach}) \mapsto v$ ist. Weiterhin sei \mathcal{R} echt. Dann paßt der assoziierte Verkehrsgraph $(H, \bar{E}) = VG(H)$ zu (V, \mathcal{R}) bezüglich f .

Beweis :

Beweis durch Konstruktion. Sei $R = [v_1, \dots, v_n] \in \mathcal{R}$ beliebig. Dann ist $\bar{R} = [\bar{v}_1, \dots, \bar{v}_n] \in \mathfrak{R}(H)$ mit

- Für alle $1 < i < n$: $\bar{v}_i = (\{v_{i-1}\}, v_i, \{v_{i+1}\}) \in H$.
(Definition 4.13 (Abbiegebeziehungen))
- Für $i = 1$: $(\emptyset, v_1, \{v_2\}) \in H$ oder $\exists v \in V : (\{v\}, v_1, \{v_2\}) \in H$. Wähle \bar{v}_1 entsprechend.
(Definition 4.13 (Quellrichtungen) und Definition 4.14 (Echtheit))
- Für $i = n$: $(\{v_{n-1}\}, v_n, \emptyset) \in H$ oder $\exists v \in V : (\{v_{n-1}\}, v_n, \{v\}) \in H$. Wähle \bar{v}_n entsprechend.
(Definition 4.13 (Senkrichtungen) und Definition 4.14 (Echtheit))

Wegen $f(V_{vor}, v, V_{nach}) \mapsto v$ gilt offenbar: $\bar{R} \sim_f R$. Wegen Definition 4.11 (Assoziierter Verkehrsgraph) gilt offenbar: \bar{R} liegt in \bar{E} . **q.e.d.**

Nun muß ich noch zeigen, daß aus dieser Konstruktion auch ein getrennter, zu H assoziierter Verkehrsgraph entsteht. Voraussetzung dafür ist die Schleifen- und Wendefreiheit von H .

Diese Eigenschaften für die Menge der Haltepunktbeziehungen $H(V, \mathcal{R})$ hängen von entsprechenden Eigenschaften der Routenbeschreibungsmenge \mathcal{R} ab.

Definition 4.15

Sei V eine Ortsmenge und $\mathcal{R} \subseteq \mathfrak{R}(V)$ eine Menge von Routenbeschreibungen.

- \mathcal{R} heißt **schleifenfrei** : $\Leftrightarrow \forall R \in \mathcal{R} \forall 1 \leq i < \max(R) : \pi(R, i) \neq \pi(R, i + 1)$.
- \mathcal{R} heißt **wendefrei** : $\Leftrightarrow \forall R \in \mathcal{R} \forall 1 \leq i < \max(R) - 1 : \pi(R, i) \neq \pi(R, i + 2)$.

Lemma 4.4

Sei (V, \mathcal{R}) ein Routennetz. Sei \mathcal{R} schleifen- und wendefrei. Dann ist die Menge der Haltepunktbeziehungen $H(V, \mathcal{R})$ ebenfalls schleifen- und wendefrei.

Beweis :

Folgt direkt aus Definition 4.13.

q.e.d.

Korollar 4.5

Sei (V, \mathcal{R}) ein Routennetz. Sei (H, f, V) eine Verfeinerung von Orten, wobei $H = H(V, \mathcal{R})$ ist die Menge von Routenbeziehungen und $f : (V_{vor}, v, V_{nach}) \mapsto v$. Weiterhin sei \mathcal{R} echt sowie schleifen- und wendefrei. Dann ist der assoziierte Verkehrsgraph $(H, \bar{E}) = VG(H)$ getrennter Verkehrsgraph zum Routennetz (V, \mathcal{R}) .

Beispiel 4.5 : Zugrunde liege das Routennetz (V, \mathcal{R}) mit $V = \{a, b, c, d\}$ und $\mathcal{R} = \{[b, a, c], [c, a, b], [d, a, c], [c, a, d]\}$. In Abbildung 4.8 sind die Menge der Haltepunktbeziehungen $H = H(V, \mathcal{R})$ und der assoziierte Verkehrsgraph $(V, \bar{E}) = VG(H)$ gezeigt.

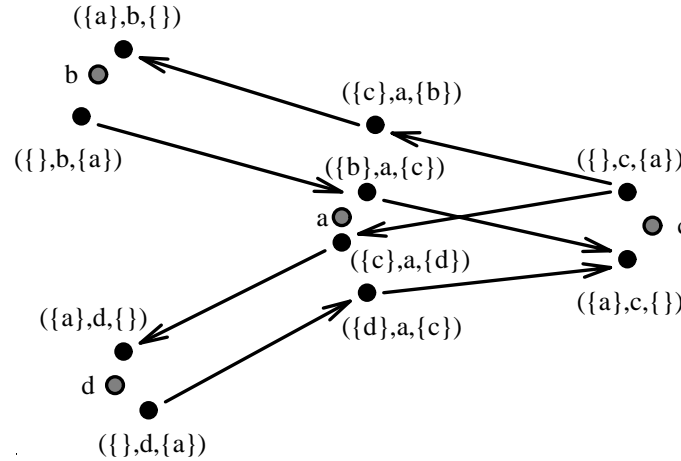


Abbildung 4.8: Der assoziierte Verkehrsgraph

Von der intuitiven Vorstellung eines getrennten Verkehrsgraphen (vgl. Abb. 4.3) sind wir mit dieser Theorie noch immer entfernt. Z.B. können die Orte $(\{c\}, a, \{b\})$ und $(\{c\}, a, \{d\})$ aus dem Beispiel zu $(\{c\}, a, \{b, d\})$ verschmolzen werden, ohne die geforderten Eigenschaften für getrennte Verkehrsgraphen zu verletzen. Um diese Lücke zu schließen, muß H nun noch reduziert werden. Diese Reduktion von Haltepunktbeziehungen ist Thema des nächsten Abschnitts.

4.2.6 Verschmelzung von Haltepunktbeziehungen

Reale Schienennetze haben oft einen gemeinsamen Haltepunkt für alle Linienverläufe, die aus der gleichen Richtung kommen oder in die gleiche Richtung führen. Um dies bei der abstrakten Modellierung getrennter Verkehrsgraphen widerspiegeln zu können, sollen Haltepunktbeziehungen verschmolzen werden, die gemeinsame Vor- oder Nachbereiche haben (und zum selben Ort gehören). Dabei darf die Trennung der Fahrtrichtungen nicht verlorengehen.

Um Ortszugehörigkeit sowie Vor- und Nachbereiche von Haltepunktbeziehungen vergleichen zu können, führe ich folgende Äquivalenzrelationen ein.

Definition 4.16

Sei V ein Ortsmenge und $H \subseteq AHP(V)$ eine Menge von Abbiegehaltepunkten. Seien $h_1, h_2 \in H$ mit $h_1 = (V_{vor,1}, v_1, V_{nach,1})$ und $h_2 = (V_{vor,2}, v_2, V_{nach,2})$. Dann gilt:

- h_1 ist **ortsäquivalent** zu h_2 ($h_1 \sim_O h_2$) $:\Leftrightarrow v_1 = v_2$.

- h_1 ist **linksäquivalent** zu h_2 ($h_1 \sim_L h_2$) $:\Leftrightarrow V_{vor,1} = V_{vor,2} \wedge v_1 = v_2$.
- h_1 ist **rechtsäquivalent** zu h_2 ($h_1 \sim_R h_2$) $:\Leftrightarrow v_1 = v_2 \wedge V_{nach,1} = V_{nach,2}$.

Bemerkung: Haltepunktbeziehungen mit gleicher Ortszugehörigkeit sind in der gleichen Äquivalenzklasse bezüglich \sim_O . Stimmen zudem die Vorbereiche (bzw. Nachbereiche) überein, so sind die Haltepunktbeziehungen auch bezüglich \sim_L (bzw. \sim_R) in der gleichen Äquivalenzklasse.

Definition 4.17 Reduziertheit

Sei V ein Ortsmenge und $H \subseteq AHP(V)$ eine Menge von Abbiegehaltepunkten. H heißt

- **linksreduziert** $:\Leftrightarrow \forall h \in H : \#[h]_L = 1$.
- **rechtsreduziert** $:\Leftrightarrow \forall h \in H : \#[h]_R = 1$.
- **reduziert** $:\Leftrightarrow H$ ist links- und rechtsreduziert.

Die Verschmelzung von rechts- oder linksäquivalenten Haltepunktbeziehungen aus einer Menge von Haltepunktbeziehungen soll zu einer reduzierten Menge von Haltepunktbeziehungen führen.

Definition 4.18 Verschmelzung

Sei V ein Ortsmenge und $H \subseteq AHP(V)$ eine Menge von Abbiegehaltepunkten. Zu H gibt es folgende Verschmelzungen:

$LV(H) = \{(V_{vor}, v, \bigcup_{(V'_{vor}, v', V'_{nach}) \in [(V_{vor}, v, V_{nach})]_L} V'_{nach}) \mid (V_{vor}, v, V_{nach}) \in H\}$
heißt **Linksverschmelzung** von H .

$RV(H) = \{(\bigcup_{(V'_{vor}, v', V'_{nach}) \in [(V_{vor}, v, V_{nach})]_L} V'_{vor}, v, V_{nach}) \mid (V_{vor}, v, V_{nach}) \in H\}$
heißt **Rechtsverschmelzung** von H .

Bemerkung: Sowohl $LV(H)$ als auch $RV(H)$ sind Teilmengen von $AHP(V)$ und daher Mengen von Abbiegehaltepunkten. $LV(H)$ ist linksreduziert und $RV(H)$ ist rechtsreduziert.

Für eine beliebige Menge von Abbiegehaltepunkten H ist es jedoch durchaus denkbar, daß $RV(LV(H))$ nicht linksreduziert ist. Genügt H folgender Bedingung, so kann man zeigen, daß $RV(LV(H))$ linksreduziert ist.

Definition 4.19 Einfachheit

Sei V ein Ortsmenge und $H \subseteq AHP(V)$ eine Menge von Abbiegehaltepunkten. H heißt **einfach** $:\Leftrightarrow \forall (V_{vor}, v, V_{nach}) \in H : \#V_{vor} \leq 1 \wedge \#V_{nach} \leq 1$.

Satz 4.6

Sei V ein Ortsmenge und $H \subseteq AHP(V)$ eine Menge von Abbiegehaltepunkten. Weiterhin sei H einfach. Dann ist $RV(LV(H))$ reduziert.

Beweis :

$RV(LV(H))$ ist rechtsreduziert. $LV(H)$ ist linksreduziert. Zu zeigen ist, daß auch $RV(LV(H))$ linksreduziert.

Beweis in drei Schritten:

1. Sei $h_1, h_2 \in LV(H), h_1 \sim_O h_2, h_1 \neq h_2$. Sei $h_1 = (V_{vor,1}, v, V_{nach,1})$ und $h_2 = (V_{vor,2}, v, V_{nach,2})$. Dann ist $V_{vor,1} \cap V_{vor,2} = \emptyset$.

Beweis durch Widerspruch: Annahme: $V_{vor,1} \cap V_{vor,2} \neq \emptyset$. Dann gilt: $V_{vor,1} = V_{vor,2}$, wegen Einfachheit von H . Dann gilt: $h_1 \sim_L h_2$. Dann gilt: $h_1 = h_2$, weil $LV(H)$ linksreduziert. Widerspruch. Also $V_{vor,1} \cap V_{vor,2} = \emptyset$.

2. Sei $h_1, h_2 \in RV(LV(H)), h_1 \sim_O h_2, h_1 \neq h_2$. Sei $h_1 = (V_{vor,1}, v, V_{nach,1})$ und $h_2 = (V_{vor,2}, v, V_{nach,2})$. Dann ist $V_{vor,1} \cap V_{vor,2} = \emptyset$.

Beweis durch Widerspruch: Annahme: $V_{vor,1} \cap V_{vor,2} \neq \emptyset$.

Dann gilt: $\exists_{(V'_{vor,1}, v, V'_{nach,1}), (V'_{vor,2}, v, V'_{nach,2}) \in LV(H)} : ((V'_{vor,1}, v, V'_{nach,1})]_R \neq [(V'_{vor,2}, v, V'_{nach,2})]_R) \wedge (V'_{vor,1} \cap V'_{vor,2} \neq \emptyset)$. Widerspruch. Also $V_{vor,1} \cap V_{vor,2} = \emptyset$.

3. Sei $h_1, h_2 \in RV(LV(H)), h_1 \sim_O h_2, h_1 \neq h_2$. Sei $h_1 = (V_{vor,1}, v, V_{nach,1})$ und $h_2 = (V_{vor,2}, v, V_{nach,2})$. Dann ist $V_{vor,1} \neq V_{vor,2}$.

Beweis: Für $V_{vor,1} \neq \emptyset \vee V_{vor,2} \neq \emptyset$ folgt dies unmittelbar aus $V_{vor,1} \cap V_{vor,2} = \emptyset$. Sowohl $V_{vor,1}$ als auch $V_{vor,2}$ können aber nicht leer sein, weil die beiden Abbiegehaltepunkte dann schon bei der Linksverschmelzung verschmolzen worden wären, und bei der Rechtsverschmelzung keine neuen leeren Vorbereiche durch Vereinigung erzeugbar sind.

q.e.d.

Bemerkung: Es reicht daher aus, einfache Mengen von Abbiegehaltepunkten je einmal einer Links- und einer Rechtsverschmelzung zu unterziehen, um die reduzierte Menge von Abbiegehaltepunkten zu erhalten.

Jetzt bleibt nur noch eines zu zeigen. Da ich die Menge der Haltepunktbeziehungen H zu einem Routennetz reduzieren will, muß ich zeigen, daß Links- und Rechtsverschmelzung keine der Eigenschaften von H tangieren, die Voraussetzung dafür sind, daß der assoziierte Verkehrsgraph getrennter Verkehrsgraph zum Routennetz ist. Dies ist im folgenden Satz formuliert.

Satz 4.7

Sei (V, \mathcal{R}) ein Routennetz. Sei (H, f, V) eine Verfeinerung von Orten, wobei $H = H(V, \mathcal{R})$ die Menge von Routenbeziehungen und $f : (V_{vor}, v, V_{nach}) \mapsto v$ ist. Weiterhin sei H schleifen- und wendefrei, der zu H assoziierte Verkehrsgraph $(H, \bar{E}) = VG(H)$ passe zu (V, \mathcal{R}) . Dann ist auch $RV(LV(H))$ schleifen- und wendefrei und der assoziierte Verkehrsgraph $(RV(LV(H)), \bar{E}) = VG(RV(LV(H)))$ paßt zu (V, \mathcal{R}) .

Beweis :

1. $RV(LV(H))$ schleifenfrei
ergibt sich direkt aus Definition 4.18 (Verschmelzung).
2. $RV(LV(H))$ wendefrei
ergibt sich direkt aus Definition 4.18 (Verschmelzung).
3. $(RV(LV(H)), \tilde{E}) = VG(RV(LV(H)))$ paßt zu (V, \mathcal{R})
Beweis durch Konstruktion. Sei $R = [v_1, \dots, v_n] \in \mathcal{R}$ beliebig. Dann ist $\bar{R} = [\bar{v}_1, \dots, \bar{v}_n] \in \mathfrak{R}(RV(LV(H)))$ mit
 - Für alle $1 < i < n$: $\exists_{(V_{vor}, v_i, V_{nach}) \in RV(LV(H))} : v_{i-1} \in V_{vor} \wedge v_{i+1} \in V_{nach}$. Wähle $\bar{v}_i = (V_{vor}, v_i, V_{nach})$ entsprechend.
(vgl. Definition 4.13 (Abbiegebeziehungen) und Definition 4.18 (Verschmelzung))
 - Für $i = 1$: $\exists_{(V_{vor}, v_1, V_{nach}) \in RV(LV(H))} : v_2 \in V_{nach}$. Wähle $\bar{v}_1 = (V_{vor}, v_1, V_{nach})$ entsprechend.
(vgl. Definition 4.13 (Quellrichtungen) und Definition 4.18 (Verschmelzung))
 - Für $i = n$: $\exists_{(V_{vor}, v_n, V_{nach}) \in RV(LV(H))} : v_{n-1} \in V_{vor}$. Wähle $\bar{v}_n = (V_{vor}, v_n, V_{nach})$ entsprechend.
(vgl. Definition 4.13 (Senkrichtungen) und Definition 4.18 (Verschmelzung))

Wegen $f : (V_{vor}, v, V_{nach}) \mapsto v$ gilt offenbar: $\bar{R} \sim_f R$. Wegen Definition 4.11 (Assoziierter Verkehrsgraph) gilt offenbar: \bar{R} liegt in \tilde{E} .

q.e.d.

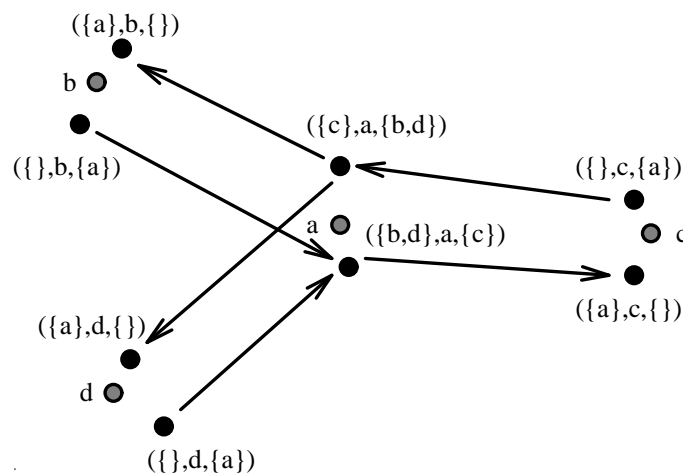


Abbildung 4.9: Assoziierter Verkehrsgraph zur reduzierten Haltebeziehungs- menge H

Beispiel 4.6 : In Abbildung 4.9 ist der assoziierte Verkehrsgraph zu sehen, der sich aus dem Beispiel in Abbildung 4.8 ergibt, wenn die Menge der Haltepunktbeziehungen H durch Links- und Rechtsverschmelzung reduziert wird.

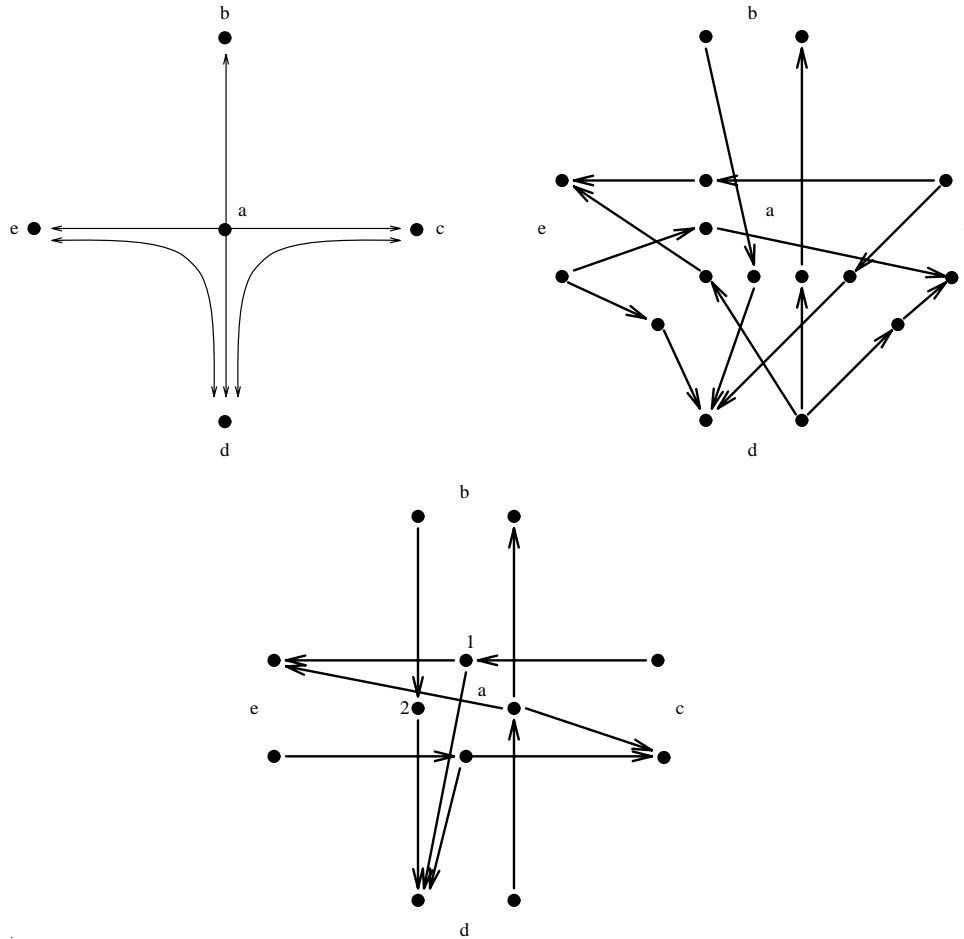


Abbildung 4.10: Vom Liniennetz zum getrennten Verkehrsnetz

Beispiel 4.7 : Ein etwas komplizierteres Beispiel ist in Abbildung 4.10 zu sehen. Links oben ist eine Ortsmenge ($V = \{a, b, c, d, e\}$) zusammen mit den Routenbeschreibungen dargestellt. Senk- und Quellrichtungen treten dabei am mittleren Ort (a) nicht auf. Rechts oben ist dann der assoziierte Verkehrsgraph zur Menge der Haltepunktbeziehungen dargestellt. Dabei wurden die Senk- und Quellrichtungen an den Orten b, c, d und e schon verschmolzen. Unten ist dann das Ergebnis von Links- und Rechtsverschmelzung an Ort a zu sehen. Es fällt auf, daß eigentlich auch noch die mit 1 und 2 markierten Haltepunktbeziehungen verschmolzen werden könnten, ohne die Getrenntheit des Verkehrsgraphen zu zerstören. Ich habe jedoch die Art der Verschmelzung mit Absicht so gewählt, daß keine neuen Abbiegebeziehungen auftreten können. In unserem Beispiel würde die Verschmelzung der beiden markierten Punkte zu einer neuen Bewegungsmöglichkeit von b über a nach e führen. Dieses Beispiel ist dem

U-Bahn-Verkehrsnetz von Stuttgart entnommen und entspricht dem Berliner Platz.

4.2.7 Verkehrsnetz mit Trennung der Fahrtrichtungen

Nachdem in den vergangenen drei Abschnitten die Trennung nur für Verkehrsgraphen durchgeführt wurde, soll nun der Kreis wieder geschlossen werden. Im folgenden ist beschrieben, wie aus dem Liniennetz das Routennetz und das Verkehrsnetz der U-Bahn gewonnen wird.

Definition 4.20

Das Routennetz (V_U, \mathcal{R}_U) heißt **U-Bahn-Routennetz zum Liniennetz** $LN_{et} = (S, L, LCon, name, koord, art, takt, abstand) : \Leftrightarrow$

- $V_U = \{s \in S \mid \exists_{(R, time, dist) \in L} : s \text{ liegt in } R \wedge art(R, time, dist) = \text{U-Bahn}\}$.
- $\mathcal{R}_U = \{R \in \mathfrak{R}(V_U) \mid \exists_{(R', time, dist) \in L} : R = R'\}$.

Aus diesem Routennetz kann nun das getrennte U-Bahn-Verkehrsnetz gewonnen werden.

Definition 4.21

Sei (V_U, \mathcal{R}_U) U-Bahn-Routennetz zum Liniennetz $LN_{et} = (S, L, LCon, name, koord, art, takt, abstand)$. Das Verkehrsnetz $\vartheta_U = (V_U, E_U, T_U, \mu_U, \nu_U, \kappa_U, \delta_U, \tau_U, \lambda_U)$ heißt **getrenntes U-Bahn-Verkehrsnetz zum Liniennetz** $LN_{et} : \Leftrightarrow$

- Der Verkehrsgraph $(V_U, E_U) = VG(H)$ ist der assoziierte Verkehrsgraph zur reduzierten Haltepunktbeziehungsmenge $H = RV(LV(H))$, wobei $H = H(V_U, \mathcal{R}_U)$ die reduzierte Haltepunktbeziehungsmenge zum Routennetz (V_U, \mathcal{R}_U) ist.
- (T_U, λ_U) ist elementare Routentabelle zu (V_U, E_U) .
- $\forall_{(V_{vor}, v, V_{nach}) \in V_U} : \mu_U(V_{vor}, v, V_{nach}) = koord(v)$.
- $\forall_{(V_{vor}, v, V_{nach}) \in V_U} : \nu_U(V_{vor}, v, V_{nach}) = name(v)$.
- $\kappa_U \equiv 1$.
- $\forall_{((V_{vor,1}, v_1, V_{nach,1}), (V_{vor,2}, v_2, V_{nach,2})) \in E_U} \exists_{(R, time, dist) \in L} \exists_{1 \leq j < max(R)} : \pi(R, j) = v_1 \wedge \pi(R, j+1) = v_2 \wedge art(R, time, dist) = \text{U-Bahn} \wedge \delta_U(V_{vor,1}, v_1, V_{nach,1}, V_{vor,2}, v_2, V_{nach,2}) = dist(j) \wedge \tau_U((V_{vor,1}, v_1, V_{nach,1}), (V_{vor,2}, v_2, V_{nach,2})) = time(j)$.

Bemerkungen:

1. Das getrennte U-Bahn-Verkehrsnetz wurde ist mit einer Kapazität von eins versehen. Das bedeutet, daß in diesem Modell immer nur eine U-Bahn auf einem Ort und den einlaufenden Verbindungen sein kann.
2. In der Darstellung werden die Koordinaten so abgeändert, daß man die linke und rechte Fahrtrichtung getrennt wahrnehmen kann.

4.3 Die Mitnahmebeziehung

Weil sowohl die Haltestellen im Fußwegenetz als auch die Haltepunkte im getrennten U-Bahn-Verkehrsnetz direkt aus den Haltestellen des Liniennetzes gewonnen werden, gestaltet sich die Mitnahmebeziehung sehr einfach.

Definition 4.22

Sei $\vartheta_P = (V_P, E_P, T_P, \mu_P, \nu_P, \kappa_P, \delta_P, \tau_P, \lambda_P)$ ein Fußwegenetz und $\vartheta_U = (V_U, E_U, T_U, \mu_U, \nu_U, \kappa_U, \delta_U, \tau_U, \lambda_U)$ ein getrenntes U-Bahn-Verkehrsnetz zum Liniennetz $LNet$. Dann ist $h_U : V_U \rightarrow V_P$ **U-Bahn-Mitnahmebeziehung** mit $h_U : (V_{vor}, v, V_{nach}) \mapsto v$.

Bemerkung: Die Abbildung h_U ist wohldefiniert, weil das Fußwegenetz alle Haltestellen des Liniennetzes als Orte enthält.

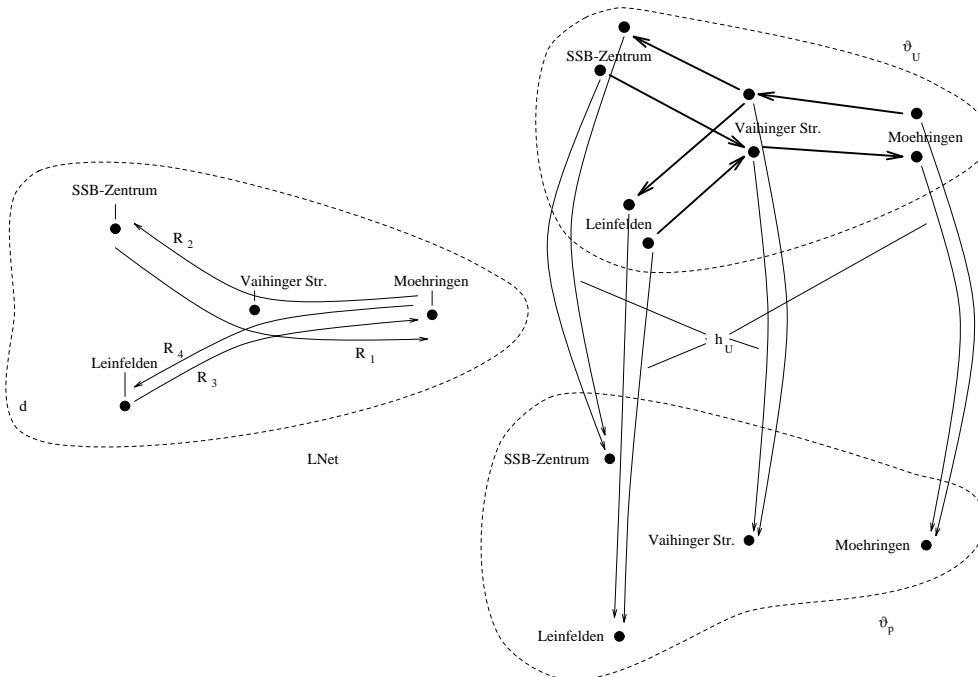


Abbildung 4.11: Modellierung der Mitnahmebeziehung

Beispiel 4.8 : Auf der linken Seite von Abbildung 4.11 ist das Liniennetz $LN_{\text{Net}} = (S, L, LCon, name, koord, art, takt, abstand)$ aus Beispiel 4.1 abgebildet. Rechts oben ist das getrennte U-Bahn-Verkehrsnetz ϑ_U und rechts unten das Fußwegenetz ϑ_P dargestellt. Dazwischen die U-Bahn-Mitnahmebeziehung h_U .

4.4 Die Fahrplanmittel

Mit den Daten des Instituts für Eisenbahn- und Verkehrswesen können die Verkehrsobjekte für die S- und U-Bahn-Verkehrsnetze nur ungenügend modelliert werden. Zwar liegen die Routen vor, aber leider keine genauen Abfahrts- und Ankunftszeiten an den Haltestellen. Auch im Rahmen einer grobkörnigen Modellierung ist dies nicht akzeptabel.

Beim Test der prototypischen Implementierung des S-Bahn-Verkehrsnetzes von Stuttgart habe ich je S-Bahn-Linie zunächst nur zwei Verkehrsobjekte fahren lassen. Dabei kam es am Tunnel zwischen Universität und Schwabstraße zu Verzögerungen von bis zu 30 Minuten, weil alle S-Bahnen gleichzeitig in Vaihingen angekommen sind und die Kapazität des Tunnels zusammen mit der Kapazität des nächsten Haltepunktes auf eins beschränkt ist. Daher werden genaue Abfahrts- und Ankunftszeiten benötigt. Momentan liegen diese jedoch vom VVS nicht in elektronischer Form vor.

Deshalb werde ich die Struktur von Verkehrsobjekten für das U-Bahn-Verkehrsnetz (S-Bahn-Verkehrsnetz analog) nur informal beschreiben. Dabei scheint es mir akzeptabel, je Linie, Fahrtrichtung und Abfahrzeit ein eigenes Verkehrsobjekt zu modellieren. Die Modellierung des Fahrtverlaufs bestimmter Fahrzeuge über einen ganzen Tag erfordert mehr Daten von den jeweiligen Betreibern.

Ein Verkehrsobjekt $o = (R, n, a, m, \gamma, \zeta, \eta)$ zum U-Bahn-Verkehrsnetz $\vartheta_U = (V_U, E_U, T_U, \mu_U, \nu_U, \kappa_U, \delta_U, \tau_U, \lambda_U)$ besteht also aus:

- der Routenbeschreibung $R \in \mathfrak{R}(V_U)$, die aus der Linienbeschreibung gewonnen wird.
- einem Namen $m \in N$, der zumindest aus der Liniennummer und der Abfahrnummer besteht.
- der Art $a = \text{fahrplanmittel}$.
- der Mitnahmekapazität $m \in \mathbb{N}$, über deren Beschaffbarkeit ich bisher keine Informationen besitze.
- der Planankunftszeit $\gamma : I(R) \rightarrow \mathbb{N}$ je Routenpunkt. Leider liegen in Fahrplänen meist nur Abfahrtszeiten vor, so daß die Planankunftszeit zunächst gleich wie die Planabfahrzeit modelliert werden muß.
- der Planabfahrtszeit $\zeta : I(R) \rightarrow \mathbb{N}$ je Routenpunkt. Diese kann dem Fahrplan (außer beim letzten Routenpunkt) entnommen werden.
- der Mindestwartezeit $\eta : I(R) \rightarrow \mathbb{N}$, die vermutlich nicht bei jeder Haltestelle verschieden ist, sondern von der Art des Verkehrsnetzes abhängt.

4.5 Die Personen

Das Institut für Eisenbahn- und Verkehrswesen prognostiziert mit Hilfe des Vierstufenalgorithmus die Verkehrsnachfrage F_{ijmr} (vgl. 2.3.3). Diese Matrix bildet die Grundlage der Modellierung von Personen.

Für jede Fahrtenzahl $f(s, z, \text{öV}, r)$ in der Verkehrsnachfrage F_{ijmr} kann dabei die Routenbeschreibung aus der ÖV-Route $r \in R_{\text{öV}}(s, z)$ gewonnen werden. Leider sind die Abfahrtszeiten jedoch nicht bekannt. Deshalb gehe ich von einer Gleichverteilung der Abfahrtszeiten über den gesamten Untersuchungszeitraum aus. Mit Hilfe der Gleichverteilung werden entsprechend viele Abfahrtszeitpunkte an der Quellverkehrszelle festgelegt und die Personen entsprechend durchnummeriert. Dadurch wird zunächst jede Fahrt durch eine Person modelliert. Mit genauen Daten kann man auch die Bewegung bestimmter Personen auf einer Wegekette modellieren.

Da mir die Verkehrsnachfrage F_{ijmr} bisher nicht elektronisch vorliegt, werde ich die Struktur von Personen für das Fußwegenetz nur informal darstellen. Ein Verkehrsobjekt $o = (R, n, a, m, \gamma, \zeta, \eta)$ zum Personen-Verkehrsnetz $\vartheta_P = (V_P, E_P, T_P, \mu_P, \nu_P, \kappa_P, \delta_P, \tau_P, \lambda_P)$ besteht aus:

- der Routenbeschreibung $R \in \mathfrak{R}(V_P)$, die von der ÖV-Route abgeleitet ist.
- einem Namen $m \in N$, der zumindest aus der Startnummer besteht.
- der Art $a = \text{person}$.
- der Mitnahmekapazität $m = 0$.
- der Planankunftszeit $\gamma : I(R) \rightarrow \mathbf{N}$ je Routenpunkt, über die zunächst keine Informationen vorliegen. Normalerweise ist $\gamma \equiv 0$.
- der Planabfahrtszeit $\zeta : I(R) \rightarrow \mathbf{N}$ je Routenpunkt. Diese wird nur für den ersten Routenpunkt aufgrund einer Gleichverteilung festgelegt. Über die weiteren Routenpunkte liegen zunächst keine Informationen vor, deshalb ist $\zeta(x) = 0$ für alle $x \in I(R) \setminus \{1\}$.
- der Mindestwartezeit $\eta : I(R) \rightarrow \mathbf{N}$, über die auch keine Informationen vorliegen. Normal gilt auch hier, daß $\eta \equiv 0$.

Kapitel 5

Einführung in THOR-Netze

Am Oldenburger Forschungs- und Entwicklungsinstitut für Informatik-Werkzeuge und Systeme (OFFIS) wurden THOR-Netze (**T**imed **H**ierarchical **O**bject-**R**elated **N**ets) als höhere Petrinetze entwickelt, um damit komplexe Systeme ereignisorientiert zu modellieren. Um den Abstraktionsaufwand bei der Modellierung herabzusetzen, wurden dabei sehr viele verschiedene Konzepte berücksichtigt, mit denen man komplexe Systemzusammenhänge darstellen kann.

Im Bericht Nr. 13 der Arbeitsgruppe Informatik-Systeme der Universität Oldenburg [FLSW93] werden THOR-Netze generisch definiert. Diese Definition ist sehr umfangreich und unübersichtlich, da alle Petri-Netz-Klassen gewissermaßen gleichzeitig generisch definiert werden sollten. Außerdem stimmt die definierte Semantik nicht vollständig mit der implementierten Semantik überein, so daß die informale Beschreibung der THOR-Netze im Handbuch zum THORN-Simulator [Wie95a] nachgelesen werden muß.

In Abschnitt 5.1 werden die in THOR-Netzen realisierten Konzepte vorgestellt. Im zweiten Abschnitt (5.2) wird die Reihenfolge der Ereignisse bei der Simulation vorgestellt, die im THORN-Simulator verwirklicht ist. Auf eine formale Beschreibung wurde verzichtet. Es wird ein Grundwissen über Petri-Netze vorausgesetzt, wie es zum Beispiel in Reisig [Rei86] oder Baumgarten [Bau90] dargestellt ist.

Im dritten Teil dieses Kapitels (Abschnitt 5.3) werden zwei Probleme vorgestellt, die bei der Verwendung von THOR-Netzen beachtet werden sollten.

Bei Kenntnis von THOR-Netzen kann dieses Kapitel übersprungen werden.

5.1 Vorstellung der Konzepte

5.1.1 Stellen mit Struktur

Im Gegensatz zu herkömmlichen Petrinetzen können Stellen in THOR-Netzen eine Struktur besitzen, die festlegt, in welcher Reihenfolge die einzelnen Marken von dieser Stelle abgezogen werden können. Es gibt folgende Strukturen:

Multiset: Entspricht den traditionellen Stellen. Die Marken können die Stelle in beliebiger Reihenfolge verlassen.

Stack: Marken können die Stelle nur in der Reihenfolge verlassen, die sich durch die Datenstruktur Stack ergibt.

Queue: Marken können die Stelle nur in der Reihenfolge verlassen, die sich durch die Datenstruktur Queue ergibt.

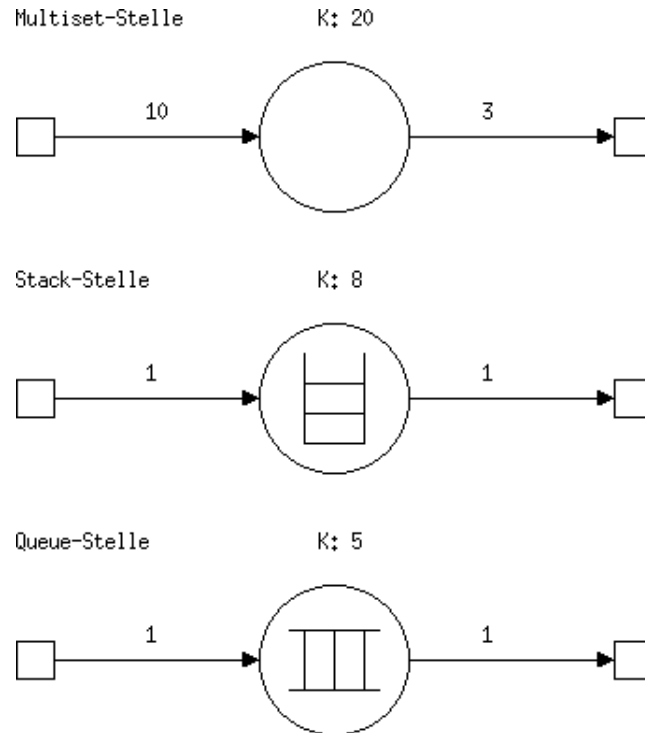


Abbildung 5.1: Multiset-, Stack- und Queue-Stellen

Im THORN-Editor werden diese Stellen durch anschauliche Symbole dargestellt (Abb. 5.1). Um Nichtdeterminismus zu vermeiden, dürfen Kanten, die zu strukturierten Stellen adjazent sind, kein Kantengewicht¹ haben, das größer als eins ist.

Stellen haben eine Kapazität, die angibt, wieviele Marken maximal gleichzeitig auf der Stelle liegen können. Diese Kapazität kann auch unendlich sein.

5.1.2 C++-Objekte als Marken

Im Gegensatz zu einfachen Petri-Netzen werden in THOR-Netzen Objekte statt Marken verwendet. Diese Objekte werden als C++-Klassen spezifiziert und als Instanzen dieser Klassen erzeugt und behandelt.

¹Kantengewichte geben dabei an, wieviele Marken beim Schalten der Transition abgezogen bzw. abgelegt werden.

Die Standardklassen `Bool`, `Int`, `Float`, `Double`, `Char`, `String` und `Token` werden dabei vordefiniert.

Als Beispiel für eine selbstdefinierte Klasse habe ich die Klasse `Person` betrachtet, die hier als Verkehrsteilnehmer modelliert ist. Attribute sind Start- und Zielort, sowie Start- und Ankunftszeit. Die Klassendeklaration in C++ sieht wie folgt aus und kann natürlich weitere Methoden enthalten:

```

/* -----
 * Klasse fuer Personen
 * ----- */

class Person
{
public:
    char        Startort[20];    // Name des Startorts
    char        Zielort[20];    // Name des Zielorts
    Zeit        Startzeit;      // Startzeit in hh:mm:ss
    Zeit        Ankunftszeit;   // Ankunftszeit in hh:mm:ss

    Person ();
};

```

Für eine Simulation muß eine Anfangsbelegung eingegeben werden. Um aus der ASCII-Beschreibung dieser Belegung Objekte zu erzeugen, benötigen sie außer dem Konstruktor noch eine Methode zum Lesen der Objektdaten. Genauso macht die Ausgabe der Endbelegung eine Methode zum Schreiben der Objektdaten erforderlich. Weitere Einzelheiten zu selbstdefinierten Objekten können dem Handbuch zum THORN-Simulator entnommen werden.

Stellen werden mit einem Typ beschriftet (Abbildung 5.2), um festzulegen, welcher Klassenspezifikation die Objekte genügen müssen, die auf dieser Stelle als Marken liegen können.

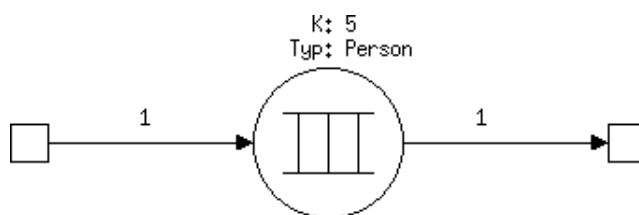


Abbildung 5.2: Stelle mit Objekttyp

Es ist geplant, für Stellen eine zusätzliche Struktur „PriorityQueue“ einzuführen², die die Reihenfolge des Zugriffs auf Objekte nicht aufgrund der Reihenfolge des Eintreffens auf der Stelle, sondern durch eine für diesen Objekttyp definierte Prioritätsfunktion festlegt.

²Diese Struktur ist in [Wie95a] sogar schon dokumentiert, obwohl sie in meiner Version des THORN-Simulators nicht implementiert ist.

5.1.3 Kantentypen

In THOR-Netzen gibt es folgende Kantentypen.

Standard: Entspricht den traditionellen Kanten. Diese Kante muß mit einem Gewicht beschriftet sein, das angibt, wieviele Objekte beim Schalten der adjazenten Transition von der adjazenten Stelle abgezogen bzw. auf ihr abgelegt werden.

Enabling: Auch dieser Kantentyp muß mit einem Gewicht beschriftet sein. Auf der adjazenten Stelle müssen mindestens soviele Objekte liegen, wie das Gewicht der Kante angibt, damit die Transition schalten kann. Allerdings werden die Objekte über Enabling-Kanten nicht abgezogen, sondern bleiben auf der Stelle liegen.

Inhibitor: Dieser Kantentyp hat kein Gewicht. Wenn auf der adjazenten Stelle mindestens ein Objekt liegt, kann die adjazente Transition nicht schalten.

Consuming: Auch dieser Kantentyp hat kein Gewicht. Die adjazente Transition schaltet unabhängig. Wenn sie schaltet, wird die adjazente Stelle vollständig geleert.

Alle Kantentypen außer den Standard-Kanten treten nur zwischen Stellen und Transitionen auf. Die Darstellung der einzelnen Kantentypen im THORN-Editor kann Abbildung 5.3 entnommen werden.

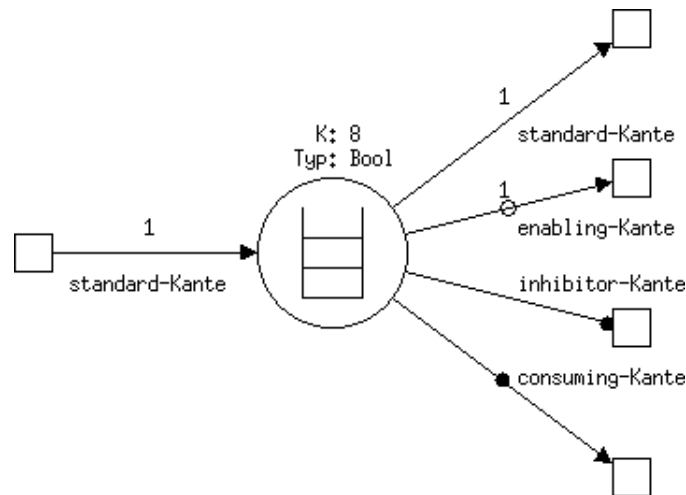


Abbildung 5.3: Die Kantentypen bei THOR-Netzen

5.1.4 Transitionen mit Programmierung

Transitionen in THOR-Netzen besitzen eine Schaltbedingung (SB), eine Schalt-dauer (SD), eine Verzögerungszeit (VZ), eine Schaltaktion (SA) sowie eine Kapazität (K). Die Kapazität gibt an, wie oft die Transition zu sich selbst parallel

schalten kann. Ist die Kapazität Null, so kann die Transition immer erst zu schalten beginnen, nachdem die Schaltdauer des letzten Schaltvorgangs verstrichen ist. Die Differenz zwischen der tatsächlichen Anzahl paralleler Schaltvorgänge und der Kapazität heißt Restkapazität.

Objekte, die am Schalten der Transition beteiligt sind, also

- Vorbereichsobjekte auf Stellen, die über Standard- oder über Enabling-Kanten mit der Transition verbunden sind, und
- Objekte, die beim Schalten erzeugt werden und im Nachbereich der Transition abgelegt werden sollen,

sind der Transition bekannt. Ihre Methoden können aufgerufen und ihre öffentlichen Attribute abgelesen werden. Schaltdauer, Verzögerungszeit, Schaltbedingung und Schaltaktion werden in C++ formuliert. Über Enabling-Kanten kann dabei nur lesend zugegriffen werden.

Die Kanten sind mit Bezeichnern für die Objekte beschriftet. Ist das Kantengewicht größer als eins, so werden die Bezeichner automatisch expandiert (Bsp.: x mit Kantengewicht 3 wird zu $x[1]$, $x[2]$ und $x[3]$).

Die C++-Ausdrücke für Schaltdauer und Verzögerungszeit müssen als Ergebnis einen positiven Integerwert haben. Die Schaltbedingung muß ein boolescher Ausdruck sein. Alle drei Ausdrücke dürfen auf die Nachbereichsobjekte nicht zugreifen³.

Die Schaltaktion kann auch Methoden der erzeugten Nachbereichsobjekte aufrufen, um sie z.B. zu initialisieren.

In Abbildung 5.4 ist eine programmierte Transition abgebildet. Diese Transition zieht von einer festen Zahl Zahlen ab, die auf der Subtrahenden-Stelle ankommen. Die Schaltbedingung stellt dabei sicher, daß nur solche Zahlen abgezogen werden, die tatsächlich kleiner sind als die feste Zahl.

Durch die Verwendung von Methoden der Objekte können auch die Objekte selbst verändert werden. Dadurch ist eine weitreichende Modellierung innerhalb der Objekte möglich, es macht jedoch die Analyse von THOR-Netzen extrem schwierig.

5.1.5 Die beiden Zeitkonzepte

In THOR-Netzen sind zwei Zeitkonzepte realisiert.

Schaltdauer: Zerlegt das Schalten einer Transition in Schaltbeginn (Objekte aus Vorbereich abziehen) und Schaltende (Nachbereichsobjekte erzeugen, Schaltaktion durchführen, Objekte im Nachbereich ablegen). Während der Schaltdauer wird Platz im Nachbereich der Transition reserviert, damit dieser nicht anderweitig beansprucht werden kann. Die Kapazität der

³Bei der Berechnung dieser Ausdrücke ist das Schalten der Transition noch nicht sicher.

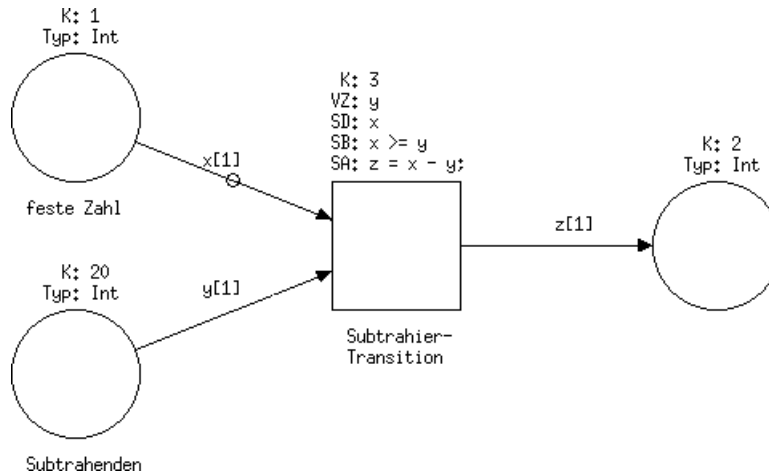


Abbildung 5.4: Transition als Subtrahierer

Transition gibt an, wie oft die Transition parallel zu sich selber schalten kann. Die Schaltdauer ist eine nicht-negative ganze Zahl⁴.

Verzögerungszeit: Eine Transition, die schalten kann, schaltet erst nach Ablauf der Verzögerungszeit. Dabei kann es vorkommen, daß Vorbereichsobjekte durch andere Transitionen abgezogen werden und die Verzögerung deshalb abgebrochen werden muß. Wenn die Verzögerungszeit abgelaufen ist, schaltet die Transition u.U. nicht sofort (wenn z.B. nicht genügend Platz im Nachbereich vorhanden ist). Die Verzögerungszeit ist deshalb eine minimale Verzögerungszeit. Die Verzögerungszeit ist eine nicht-negative ganze Zahl.

Das Zusammenspiel der Zeitkonzepte mit der Schaltreihenfolge bei der Simulation von THOR-Netzen wird in Abschnitt 5.2.3 genauer erklärt.

5.1.6 Das Hierarchiekonzept

Da Petrinetze sehr leicht sehr unüberschaubar werden, gibt es eine Vielzahl von Konzepten, größere Netze hierarchisch zu strukturieren. In THOR-Netzen sind zwei dieser Konzepte realisiert:

Aufrufkonzept: Das Schalten einer **Aufruftransition** ruft ein THOR-Netz als Unternetz auf. Aus dem Vorbereich dieser Transition können Marken auf Stellen des neuen Unternetzes kopiert werden. Eine ausgezeichnete Transition im Unternetz beendet durch ihr Schalten den Unternetzaufruf. Von bestimmten Stellen werden dann Marken in den Nachbereich der aufrufenden Transition kopiert. Durch dieses Konzept kann sich die Größe

⁴Die Schaltdauer Null wurde in der Entwicklungsphase von THOR-Netzen erst sehr spät eingeführt. Durch Fallstudien wurde sichtbar, daß kausale Ursache-Wirkung-Zusammenhänge oft ohne Schaltdauer modelliert werden müssen.

von THOR-Netzen während der Simulation ändern. Der Unternetz-Aufruf ist auch rekursiv möglich.

Transitionsverfeinerung: Eine **Transitionsverfeinerung** steht in Wirklichkeit für ein ganzes THOR-Unternetz. Dies ist nur ein syntaktisches Konzept. Durch Einsetzen des Unternetzes an die Stelle der Transition ändert sich das Netz nicht, es wird nur unübersichtlicher.

Bei THOR-Netzen werden beide Konzepte durch die Aufruftransition kombiniert. Außer mit normalen Kanten können Stellen auch durch sogenannte Hyper-Kanten mit einer Aufruftransition verbunden sein. Für Stellen, die über Standard- oder Enabling-Kanten bzw. über ein Hyper-Kante mit der Aufruftransition verbunden sind, gibt es korrespondierende Stellen im Unternetz. In Abbildung 5.5 sind eine Aufruftransition und das zugehörige Unternetz abgebildet.

Hyperkanten sind gestrichelt dargestellt, die zugehörigen Stellen heißen Share-Stellen. Stellen, die über Standard- oder Enabling-Kanten an die Aufruftransition angebunden sind, heißen Ein-/Ausgabestellen.

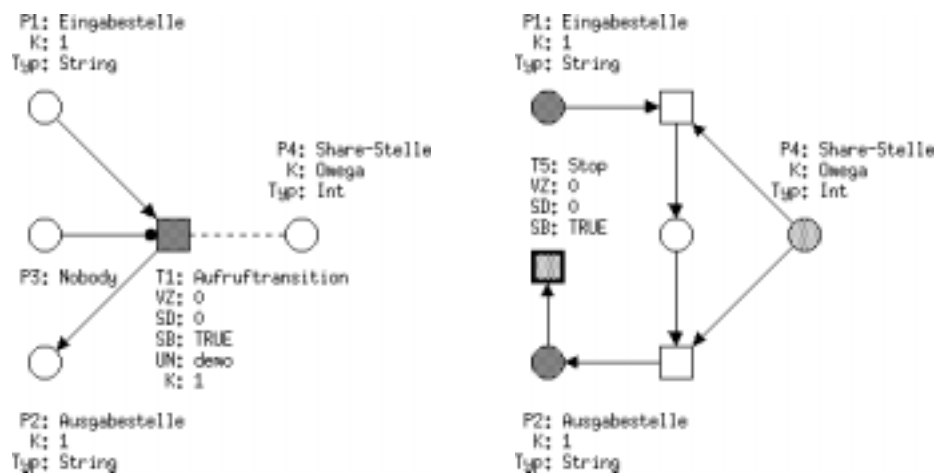


Abbildung 5.5: Eine Aufruftransition (links) und das Unternetz (rechts)

Kann die Transition T1 schalten (weil auf P1 eine Marke und auf P3 keine Marke liegt, vgl. 5.2.3), so wird das Unternetz rechts in der Abbildung erzeugt. Die Share-Stelle P4 wird dabei nicht erzeugt, sondern tatsächlich mit der entsprechenden Stelle im aufrufenden Netz identifiziert. Die Objekte der Eingabestelle P1 werden vom aufrufenden Netz in das Unternetz kopiert. Das Schalten der Stop-Transition T5 im Unternetz bewirkt das Kopieren der Objekte auf der Ausgabestelle P2 im Unternetz auf die entsprechende Stelle P2 im aufrufenden Netz. Anschließend wird das Unternetz gelöscht.

Aus- und Eingabe-Stellen bestehen je einmal für das aufrufende Netz und für das Unternetz. Während das Unternetz besteht, arbeiten sie vollkommen unabhängig voneinander.

Ist eine Aufruftransition nur über Hyperkanten mit Stellen verbunden und besitzt das Unternetz keine Stop-Transition, dann schaltet die Transition einmal am Anfang der Simulation und spannt das Unternetz auf. Dieser Fall entspricht einer Transitionsverfeinerung.

5.2 Simulation von THOR-Netzen

Die Simulation von THOR-Netzen ist wegen der Vielzahl der Konzepte nicht einfach zu beschreiben. Insbesondere die Bedingungen für das Schalten von Transitionen sind sehr kompliziert, da die Ausdrücke für Verzögerungszeit und Schaltbedingung vor dem eigentlichen Schalten ausgewertet werden müssen.

Weil man auch vor dem Schaltbeginn schon von einem Ereignis sprechen kann, spricht man daher statt von schaltenden Transitionen von Ereignissen. In diesem Abschnitt wird der Ereignisbegriff zunächst eingeführt (Teil 5.2.1). Anschließend werden die Zustandsübergänge bei Schaltbeginn und -ende informal dargelegt (Teil 5.2.2). In Teil 5.2.3 wird die Schaltregel und damit die Schaltreihenfolge beschrieben.

5.2.1 Ereignisse

Die folgende Liste faßt den Ereignisbegriff informal.

- Eine Transition und eine Menge von Objekten auf den Vorbereichsstellen der Transition heißt **Ereignis**, wenn die Objekte gemäß der einführenden Standard- und Enabling-Kantengewichte verteilt sind und auf den Vorbereichsstellen mit Inhibitor-Kanten keine Objekte liegen.
- Ein Ereignis **erfüllt die Schaltbedingung** genau dann, wenn die Objekte die Schaltbedingung der Transition erfüllen. Solange ein Ereignis seine Schaltbedingung nicht erfüllt, heißt es **undefined**.
- Sobald ein Ereignis die Schaltbedingung erfüllt, beginnt es zu **verzögern**. Zwischen dem **Beginn** und dem **Ende** der Verzögerung heißt ein Ereignis **verzögernd** und besitzt eine **Restverzögerungszeit**. Beim Beginn der Verzögerung eines Ereignisses wird die Verzögerungszeit berechnet und als Restverzögerungszeit gesetzt.
- Ein Ereignis ist **vollständig verzögert**, wenn seine Restverzögerungszeit Null ist.
- Eine Ereignis heißt **aktiviert**, wenn
 - es vollständig verzögert ist,
 - die Transition eine Restkapazität größer oder gleich Eins besitzt,
 - die Objekte abgezogen werden können⁵ und

⁵bei strukturierten Stellen ist das Entfernen von Objekten nur möglich, wenn die Datenstruktur dies zuläßt.

- genügend Platz zur Ablage der zu erzeugenden Objekte im Nachbarbereich der Transition vorhanden ist.
- Sobald ein Ereignis aktiviert ist, beginnt es zu **schalten**. Zwischen **Schaltbeginn** und **Schaltende** heißt ein Ereignis **schaltend** und besitzt eine **Restschaltdauer**. Beim Schaltbeginn wird die Schaltdauer berechnet und als Restschaltdauer gesetzt.

Ein Ereignis kann vor dem Schaltbeginn verschwinden, wenn die Voraussetzungen für seine Existenz verletzt sind.

Sind mehrere Ereignisse gleichzeitig aktiviert, so kann der Schaltbeginn eines dieser Ereignisse ein anderes auch wieder deaktivieren.

Der Lebenszyklus von Ereignissen ist in Abbildung 5.6 in Form eines Petri-Netzes dargestellt.

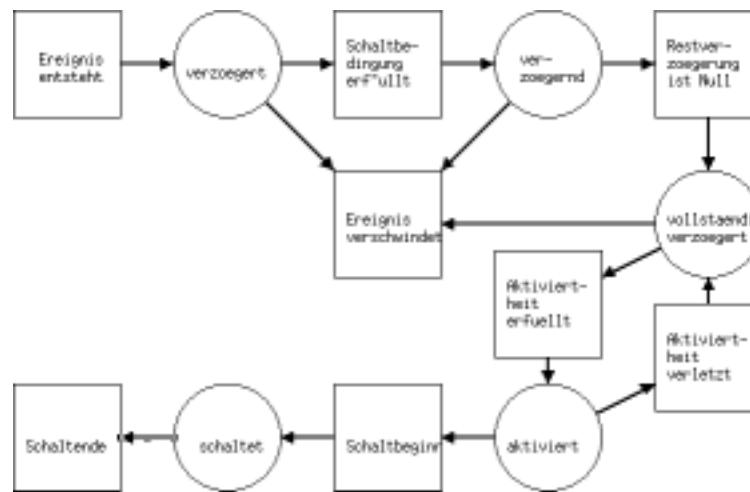


Abbildung 5.6: Lebenszyklus von Ereignissen

Ein Ereignis kann auch nur aus einer Transition bestehen, wenn die Transition keine einführenden Standard- oder Enabling-Kanten besitzt.

5.2.2 Schaltbeginn und Schaltende

Schaltende und Schaltbeginn eines Ereignisses können in den Lebenszyklus anderer Ereignisse eingreifen. Deshalb werden an dieser Stelle alle Aktivitäten bei Schaltbeginn und Schaltende aufgezählt sowie die möglichen Konsequenzen für andere Ereignisse genannt.

Beim Schaltbeginn eines Ereignisses

- werden die Objekte im Vorbereich der Transition über
 - Standard-Kanten abgezogen und über
 - Enabling-Kanten zur Verwendung in der Schaltaktion kopiert,
- alle Objekte über Consuming-Kanten abgezogen und gelöscht,
- Platz entsprechend der Kantengewichte im Nachbereich der Transition reserviert und
- die Restschaltkapazität der Transition dekrementiert.

Dabei können andere Ereignisse

- verschwinden (weil ihre Vorbereichsobjekte abgezogen wurden),
- deaktiviert werden (weil entweder die Kapazität der Transition oder die Kapazität der Nachbereichsstellen beansprucht wurde) oder
- aktiviert werden (weil Objekte aus ihrem Nachbereich abgezogen wurden und nun genügend Platz zur Verfügung steht).

Beim Schaltende eines Ereignisses

- werden Objekte für den Nachbereich der Transition erzeugt,
- wird die Schaltaktion durchgeführt,
- werden die Objekte im Nachbereich der Transition abgelegt,
- die reservierten Plätze freigegeben und
- die Restschaltkapazität der Transition inkrementiert.

Dabei können andere Ereignisse

- verschwinden (weil ein Objekt auf eine Stelle gelegt wurde, die mit einer Inhibitor-Kante verbunden ist),
- erzeugt werden (weil Objekte abgelegt wurden) oder
- aktiviert werden (weil die Kapazität der Transition freigegeben wurde).

5.2.3 Die Schaltregel

Da sich die Ereignisse gegenseitig beeinflussen, muß die Schaltreihenfolge für

- das Schaltende von Ereignissen,
- den Schaltbeginn von Ereignissen und
- das Inkrementieren aller Restschaltdauern und Restverzögerungszeiten größer als Null

in einer Schaltregel festgelegt werden.

Die aktuelle Schaltregel für THOR-Netze lautet:

1. Die Menge der Ereignisse, deren Restschaltdauer Null ist, wird gleichzeitig beendet. Neue Ereignisse, die die Schaltbedingung erfüllen, bekommen ihre Restverzögerungszeit gemäß der Verzögerungszeit der Transition.
2. Aus der Menge der aktivierten Ereignisse wird eines zufällig ausgewählt und beginnt zu schalten. Hat die Transition eine Schaltdauer größer Null, so bekommt das Ereignis seine Restschaltdauer gemäß der Schaltdauer der Transition. Hat die Transition die Schaltdauer Null, so beendet das Ereignis das Schalten unmittelbar danach. Neue Ereignisse, die die Schaltbedingung erfüllen, bekommen ihre Restverzögerungszeit gemäß der Verzögerungszeit der Transition.
3. Wenn die Menge der aktivierten Ereignisse nicht leer ist, dann wiederhole Schritt 2.
4. Dekrementiere gleichzeitig alle Restverzögerungszeiten und Restschaltdauern, die größer als Null sind.

Im Gegensatz zu einer maximalen Schaltregel wird immer nur ein Ereignis ausgewählt und geschaltet.

5.3 Schwierigkeiten

5.3.1 Maximale Schaltregel

Bei der Simulation von Petri-Netzen ohne Zeitbegriff wird häufig versucht, bei jedem Simulationsschritt eine maximale Menge von Transitionen gleichzeitig zu schalten. So entsteht auch ohne explizite Zeitmodellierung der Begriff der Gleichzeitigkeit. In Abbildung 5.7(a) ist ein Beispiel ohne Zeitbeschriftung gegeben. Im ersten Simulationsschritt schalten T1 und T3 gleichzeitig. T4 kann nicht gleichzeitig schalten, weil die Nachbereichsstelle S4 erst geräumt werden muß. Im zweiten Simulationsschritt schaltet T4.

Diese Zusammenfassung von Simulationsschritten ist auch bei Einführung einer Schaltdauer noch sinnvoll. Man kann dann Restschaltdauern für Schaltvorgänge

einführen. Das Schalten zerfällt in Schaltbeginn und Schaltende. Das Dekrementieren der Restschaltdauern wird an einen Simulationsschritt gekoppelt.

Schaltregel:

1. Maximale Menge von Schaltendeereignissen gleichzeitig ausführen.
2. Maximale Menge von Schaltbeginneereignissen gleichzeitig ausführen.
3. Alle Restschaltdauern um eins verringern.

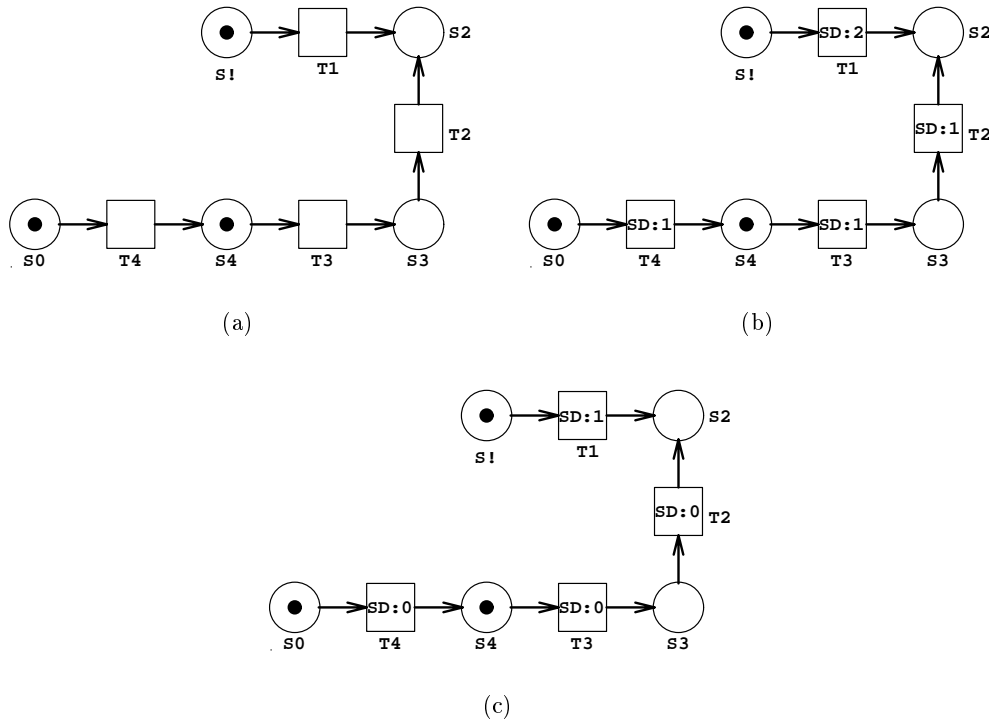


Abbildung 5.7: Die Auswahl von Schaltereignissen

Betrachtet man diese Regel zunächst für das Beispiel in Abbildung 5.7(b) (also ohne Schaltdauern gleich Null), so ergibt sich folgende Simulationsreihenfolge:

- 1. Iteration:** Keine Schaltendeaktionen, Schaltbeginn für T1 und T3, nach dem Verringern der Restschaltdauern ist die Restschaltdauer von T1 Eins und die Restschaltdauer von T3 Null.
- 2. Iteration:** Schaltende für T3, Schaltbeginn für T4, nach dem Verringern der Restschaltdauern sind die Restschaltdauern für T1 und T4 jeweils Null.
- 3. Iteration:** Schaltende für T4 und T1.

Hier wird also der Schaltbeginn von T4 eine Zeiteinheit verzögert, nachdem Stelle S4 geräumt ist. Dies wird zum Problem, wenn Transitionen mit Schaltdauern gleich Null zugelassen sind.

Betrachtet man nämlich das Beispiel in Abbildung 5.7(c), so ergibt sich folgende Simulationsreihenfolge:

- 1. Iteration:** Keine Schaltendeaktionen, Schaltbeginn für T1 und T3, nach dem Verringern der Restschaltdauern sind beide Restschaltdauern gleich, nämlich Null.
- 2. Iteration:** Schaltende für T1 und T3, Schaltbeginn für T4, nach dem Verringern der Restschaltdauer ist die Restschaltdauer für T4 Null.
- 3. Iteration:** Schaltende für T4.

Damit ist aber der Grund für die Einführung der Schaltdauer Null hinfällig. Es sollten zeitlose Zusammenhänge modelliert werden. Obwohl Transition T3 die Schaltdauer Null hat, wird T4 eine Zeiteinheit verzögert, bevor sie zu schalten beginnt.

Vermutlich wurde die Simulation von THOR-Netzen deswegen nicht mit einer maximalen Schaltregel realisiert, sondern die in Teil 5.2.3 vorgestellte Schaltregel verwirklicht. Für das letzte Beispiel ergibt sich mit der realisierten Schaltregel (vgl. Teil 5.2.3) viele mögliche Schaltreihenfolgen. Davon habe ich drei ausgewählt.

- Schaltbeginn T1, Schaltbeginn T3, Schaltende T3, Schaltbeginn T4, Schaltende T4 ...
- Schaltbeginn T3, Schaltende T3, Schaltbeginn T1, ...
- Schaltbeginn T3, Schaltende T3, Schaltbeginn T2, Schaltende T2 ...

Wie man sieht, hat die implementierte Regel den Nachteil, hochgradig nichtdeterministisch zu sein. Trotzdem finde ich es gut, daß die Marke, die zu Anfang auf Stelle S4 liegt, eine Chance hat, den Konflikt zwischen T2 und T1 für sich zu entscheiden, da sie für das Schalten von T3 keine Zeit benötigt. Außerdem schaltet T4 mit dieser Schaltregel immer vor der Verringerung aller Restschaltdauern.

5.3.2 C++-Objekte und Verzögerungszeit

Ein wesentliches Problem bei der Simulation von THOR-Netzen besteht darin, zu erkennen, welche Ereignisse verzögert werden können. Das Entstehen von Ereignissen ist wie bei herkömmlichen Petrinetzen an den Schaltbeginn und das Schaltende anderer Transitionen geknüpft (vgl. Teil 5.2.2).

In Abbildung 5.8 gibt es eine Transition die aus einem Ampelgehäuse (links unten) und drei verschiedenfarbigen Farblichtern (links oben) eine Ampel (rechts) herstellt. Alle Stellen sind zunächst leer. Nun werden gleichzeitig 20 Ampelgehäuse und 50 Farblichter auf den entsprechenden Stellen abgelegt. Jede Auswahl von drei Farblichtern und einem Ampelgehäuse ist ein Ereignis.

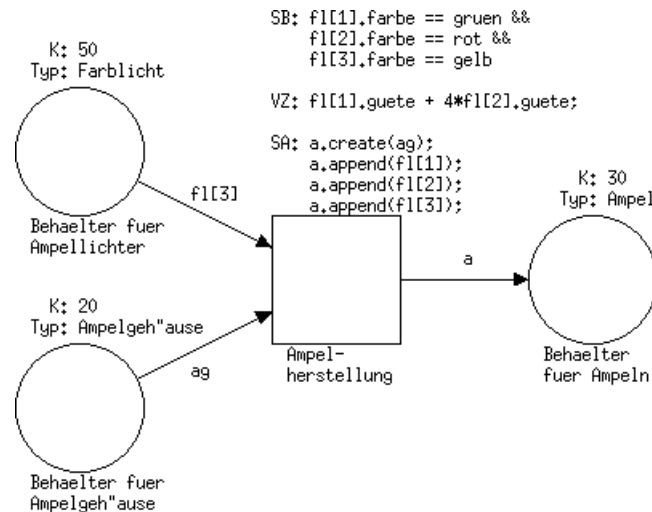


Abbildung 5.8: Herstellung von Ampeln

Tatsächlich muß jedoch auch jede Permutation der drei Farblichter berücksichtigt werden, da es nicht entscheidbar ist, ob die Reihenfolge eine Bedeutung hat. Im Beispiel ist es so, daß die Verzögerungszeit stark von der Güte roter Farblichter und schwach von der Güte grüner Farblichter abhängt. Damit ergeben sich $\binom{50}{3} \times 3! \times 20 = 392000$ verschiedene Ereignisse, deren Schaltbedingung überprüft werden muß.

Für dieses Laufzeitproblem gibt es bisher keine zufriedenstellende Lösung⁶.

Aus dem Beispiel wird klar, daß man auf eine Verwendung von Kantengewichten ≥ 1 soweit als möglich verzichten sollte.

⁶Im Februar 1996 lief in diesem Zusammenhang bei OFFIS eine Diplomarbeit mit dem Ziel, die Mächtigkeit von C++ für Schaltbedingung und Verzögerungszeit einzuschränken und beide Ausdrücke vom THOR-Compiler parsen und optimieren zu lassen. Über die Ergebnisse dieser Arbeit liegen mir allerdings keine Informationen vor.

Kapitel 6

Prototyp mit THOR-Netzen

Aus dem Simulationsmodell für den ÖPNV, das in Kapitel 3 entwickelt wurde, ist mit Hilfe von THOR-Netzen ein Prototyp entstanden. In diesem Kapitel wird beschrieben, wie die einzelnen Konzepte des Simulationsmodells mit THOR-Netzen realisiert wurden.

Bei der Umsetzung des Simulationsmodells in ein THOR-Netz liegt es nahe, Personen und Fahrplanmittel als C++-Objekte zu modellieren. Diese können sich als Marken durch ihre jeweiligen Verkehrsnetze bewegen. Leider läßt sich die Mitnahme einer Klasse von Marken durch eine andere Klasse von Marken nicht direkt mit den Konzepten höherer Petri-Netze beschreiben. Deshalb muß die Mitnahme von Personen durch Fahrplanmittel im C++-Code realisiert werden.

Abweichend von der Spezifikation gibt es nicht nur eine Quelle. An jeder Stelle im THOR-Netz, die Eintrittsstelle von Verkehrsobjekten ist, muß eine Quelle modelliert werden.

Der Fahrtverlauf wird nicht in den C++-Objekten verwaltet. Stattdessen werden die Ankunfts- und Abfahrtszeiten von Verkehrsobjekten an bestimmten Orten (Quellen, Senken und Haltestellen/-punkten) direkt in eine Ausgabedatei geschrieben. Deshalb können die Verkehrsobjekte nach dem Austritt aus dem Verkehrsnetz gelöscht werden.

In Abschnitt 6.1 wird der Klassenentwurf für die Realisierung der Verkehrsobjekte und in Abschnitt 6.2 die Realisierung des Simulationsmodells in Form von THOR-Netzen dargestellt. Zum besseren Verständnis der programmierten Transitionen sind in Anhang A Teile der Klassendeklaration für die benutzerdefinierten Objekte abgedruckt.

Bei der Modellierung von Verkehr mit THOR-Netzen habe ich einige Anregungen aus den Arbeiten von Ralf Wieting [Wie95b] und Stefan Preuß [Pre93] entnommen. Höhere Petri-Netze wurden auch von Frank DiCesare [DKGL94] und Huaitien Wang [WLD93] zur Modellierung von Verkehr verwendet. In [FHKP93] werden Petri-Netze mit Smalltalk-Objekten als Marken verwendet, um den Verkehrsfluß über eine Strassenkreuzung zu optimieren.

In Abschnitt 6.3 werden die Abweichungen des Prototyps von der formalen Spezifikation des Simulationsmodells genannt. In Abschnitt 6.4 werden einige Möglichkeiten zur Beschleunigung der Simulation beschrieben.

6.1 Der Klassentwurf

Implementiert man die Aufnahme von Personen in Fahrplanmitteln, so bietet sich eine Verwaltung mit Listen an. Da auch die Routenbeschreibung als Liste verwaltet wird, ist es sinnvoll, die mitgenommenen Personen an dem Routenpunkt zu verwalten, an dem sie aussteigen werden.

In Abbildung 6.1 ist der grobe Klassentwurf abgebildet. Sowohl Personen als auch Fahrplanmittel sind als Klassen vom Verkehrsobjekt abgeleitet. Die Routenbeschreibung ist einmal als Route und einmal als Abfahrplan realisiert. Der Abfahrplan verweist für jeden Halt nicht nur auf den Ort, sondern auch auf eine Liste von Personen (Fahrgastliste) die an diesem Ort aussteigen. Halt besitzt zwei Verweise auf Orte, weil neben dem Ort im Schienennetz (Haltepunkt) auch die zugeordnete Haltestelle im Fußwegenetz bekannt sein muß.

Die Klassen, deren Objekte sich durch das THOR-Netz bewegen sollen, sind in der Darstellung grau unterlegt.

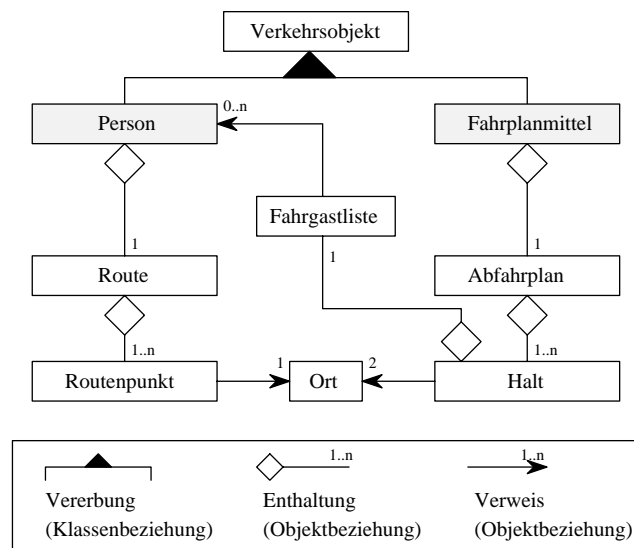


Abbildung 6.1: Klassentwurf für den Prototyp

Beschreibung der Klassen:

class Verkehrsobjekt

Beschreibung: Superklasse für Person und Fahrplanmittel.

class Person **superclass** Verkehrsobjekt

Attribute: Route

Position

Naechster_ort

Am_ziel

Startzeit

Zielzeit

Methoden: `setze_route(Route)`
`position(Ort)`

Beschreibung: *Position* ist der zuletzt angefahrene Routenpunkt und *Naechster_ort* der nächste Routenpunkt. *Am_ziel* ist wahr, wenn der zuletzt angefahrene Routenpunkt bereits erreicht ist. *Startzeit* und *Zielzeit* werden intern benötigt, weil diese Informationen erst am Ende der Fahrt in das Ausgabe-File geschrieben werden sollen. Die Methode *position* informiert das Verkehrsobjekt über den aktuellen Ort.

class Route

Attribute: LIST[Routenpunkt]

Methoden: `append(Routenpunkt)`

Beschreibung: Eine Liste von Routenpunkten. Neben der angegebenen Methode muß der Zugriff implementiert werden.

class Routenpunkt

Attribute: Ort

Beschreibung: Ein Routenpunkt von Personen verweist auf einen *Ort* im Fußwegenetz.

class Fahrplanmittel **superclass** Verkehrsobjekt

Attribute: Abfahrplan

`Position`

`Naechster_halt`

`Anzahl`

`Hier_halten`

`Am_ziel`

Methoden: `setze_fahrplan(Abfahrplan)`

`position(Ort)`

`einstieg(Person)`

`ausstieg`

`ausstieg_fertig`

`geht_nach(Ort)`

Beschreibung: *Position* ist der zuletzt angefahrene Routenpunkt und *Naechster_halt* der nächste Routenpunkt. *Am_ziel* ist wahr, wenn der letzte Routenpunkt bereits erreicht ist. *Anzahl* gibt an, wieviele Personen befördert werden können. *Hier_halten* ist wahr, wenn der aktuelle Ort im Abfahrplan vorkommt. Die Methode *position* informiert das Verkehrsobjekt über den aktuellen Ort. Die Methoden *einstieg*, *ausstieg* und *ausstieg_fertig* regeln den Ein- und Ausstieg von Personen. Die Methode *geht_nach* testet, ob ein bestimmter Ort im Fußwegenetz angesteuert wird.

class Abfahrplan

Attribute: LIST[Halt]

Abfahrnummer

Linie

Max_anzahl

Methoden: append(Halt)

Beschreibung: Neben einer Liste von Haltepunkten werden im Abfahrplan auch die *Abfahrnummer*, die *Linie* und die *Max_anzahl* der Fahrgäste verwaltet. So ist es möglich, alle relevanten Informationen für einen Fahrtverlauf in den Abfahrplan zu stecken. Beim Start der Simulation existieren dann nur die Abfahrpläne als Marken.

class Halt

Attribute: Fahrgastliste

Haltepunkt

Haltestelle

Ankunftszeit

Abfahrtszeit

Beschreibung: *Haltepunkt* und *Haltestelle* sind Orte im öffentlichen Netz bzw. im Fußwegenetz. *Ankunftszeit* und *Abfahrtszeit* sind Planankunftszeit bzw. Planabfahrtszeit am entsprechenden Routenpunkt.

class Fahrgastliste

Attribute: LIST[Person]

Beschreibung: Eine Fahrgastliste ist eine Liste von Personen. Es sind die üblichen Operationen notwendig.

Die Methoden für Initialisierung, Kopie und Speicherfreigabe sind hier nicht genannt. Neben den vorgestellten Klassen gibt es noch weitere, die z.B. bei der Realisierung von Quellen und Senken mit THOR-Netzen eine Rolle spielen. Hier soll nur ein grober Überblick gegeben werden, damit die Programmierung der Transitionen in den nächsten Abschnitten verständlich ist. Genauere Informationen können den Klassenbeschreibungen in Anhang A entnommen werden.

Statt Personen als Klasse zu implementieren, wurden Personengruppen implementiert, so daß ein Objekt mehrere Personen repräsentieren kann. Dies war notwendig, um eine ausreichende Geschwindigkeit der Simulation zu erreichen.

6.2 Die Umsetzung in THOR-Netze

6.2.1 Modellierung der Zeit

Öffentliche Verkehrsmittel dürfen nicht zu früh von einer Haltestelle abfahren. Auf den ersten Blick sieht es so aus, als ob man das Fortschreiten der Zeit

innerhalb der Verkehrsobjekte modellieren kann. Dies ist jedoch nicht der Fall, da es nie sicher ist, ob eine Marke im THOR-Netz eine unfreiwillige Verzögerung in Form eines Konflikts erfahren hat. Da es andererseits nicht möglich ist, auf die interne Uhr des THORN-Simulators zuzugreifen, muß das Fortschreiten der Zeit explizit in einem konfliktfreien Netz simuliert werden.

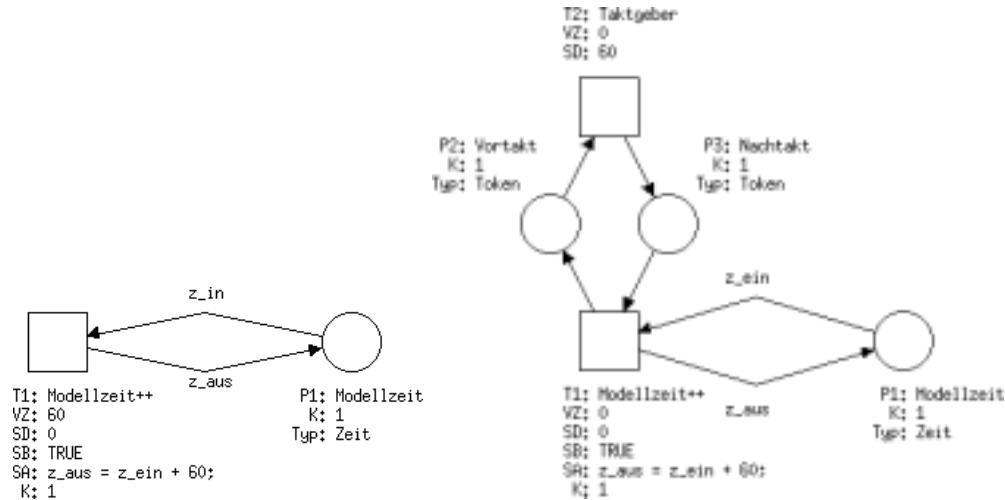


Abbildung 6.2: Modellzeit als THOR-Netz

Die einfachste Lösung dazu ist in Abbildung 6.2 (links) gezeigt. Am Anfang liegt auf Stelle P1 eine Zeitmarke mit der Anfangszeit der Simulation (z.B. 00:00:00). Die Transition T1 erhöht diese Zeit alle 60 Zeiteinheiten. Diese Lösung habe ich nicht verwendet, weil die Verzögerungszeit in meiner Version des THORN-Compilers fehlerhaft implementiert war¹.

Um die Modellzeit nur mit Schaltdauer genauso zu implementieren, benötigt man das THOR-Netz in Abbildung 6.2 (rechts). Die Anfangsmarkierung enthält zusätzlich zur Zeit noch ein Token auf Stelle P2. Es dauert dann jeweils 60 Zeiteinheiten, bis das Token auf Stelle P3 liegt. Dann kann Transition T1 schalten und die Zeit erhöhen.

Beide Lösungen sind nur möglich, weil für Transitionen mit Schaltdauer Null durch die Schaltregel (Abschnitt 5.2.3) garantiert ist, daß das Schaltende unmittelbar auf den Schaltbeginn folgt. Die Marke für die Modellzeit liegt daher immer auf Stelle P1 und kann jederzeit gelesen werden (z.B. um ein Fahrplanmittel gemäß Planabfahrtszeit abfahren zu lassen).

6.2.2 Die Verkehrsnetze

In Abbildung 6.3 ist die Umsetzung einer Verbindung zwischen zwei Orten für Fahrplanmittel abgebildet. Auf der Stelle P5 liegen die Wegparameter als Marke. Mit `par.wegzeit()` kann die Fahrtdauer von Haltestelle A nach Haltestelle B

¹Dieser Fehler ist inzwischen durch OFFIS behoben

abgefragt werden. Die Methode `par.zwischenziel()` gibt den Ort zurück, der durch Stelle P3 modelliert wird.

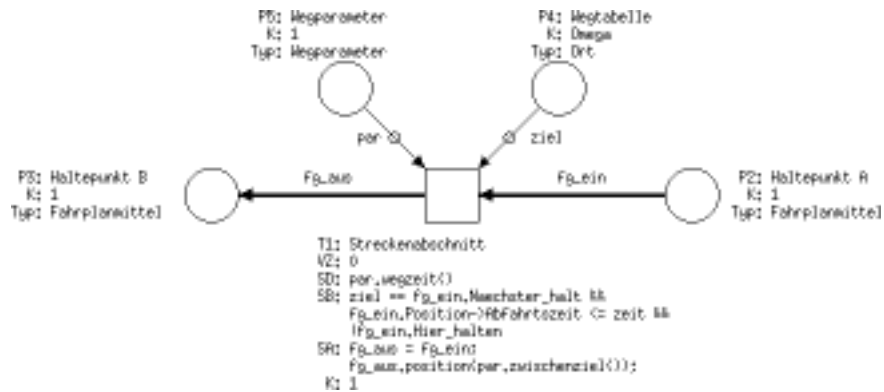


Abbildung 6.3: Eine Verbindung für Fahrplanmittel als THOR-Netz

Die Routentabelle wird durch die Stelle P4 realisiert. Dort liegen alle Orte als Marken, die über die Verbindung erreicht werden können. Durch die Schaltbedingung der Transition können sich Verkehrsobjekte nur dann auf dieser Verbindung bewegen, wenn ihr nächster Routenpunkt auch auf Stelle P4 enthalten ist. Die Schaltbedingung stellt weiterhin sicher, daß die Fahrplanmittel nicht abfahren können, falls an Stelle P2 noch Leute aus- oder einsteigen müssen (`Hier_halten`).

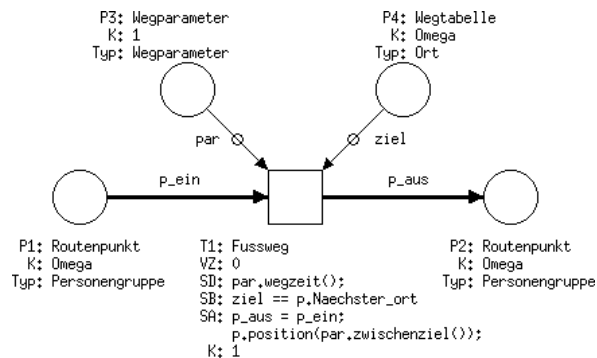


Abbildung 6.4: Eine Verbindung für Personen als THOR-Netz

Im Fußwegenetz (Abb. 6.4) muß die Schaltbedingung der Transition keinen Test daraufhin enthalten, ob das Verkehrsobjekt noch halten soll.

6.2.3 Die Mitnahmebeziehung

Die Mitnahmebeziehung zwischen Haltepunkten und Haltestellen ist im Prototyp als Aufruftransition gelöst (Abb. 6.5). Die Ankunft eines Fahrplanmittels auf Stelle P1 erzeugt ein Unternetz für den Einstieg von Personen. Dies geschieht jedoch nur, falls das Fahrplanmittel `Hier_halten` muß. Die Stelle P2,

die die Haltestelle für Personen modelliert, ist als Share-Stelle über eine Hyperkante an die Aufruftransition angeschloßen. Dadurch ist es möglich, beliebig viele Personen aus- oder einsteigen zu lassen. Auch die Modellzeit ist als Share-Stelle angeschloßen, damit das Fahrplanmittel zum richtigen Zeitpunkt abfahren kann.

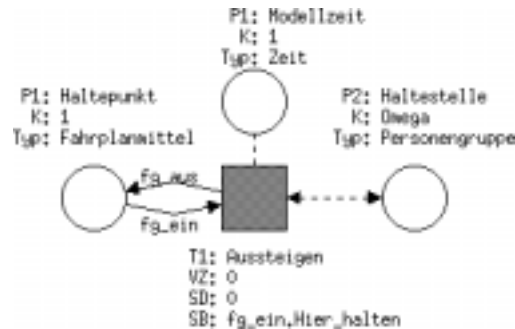


Abbildung 6.5: Aufruftransition für Aus- und Einstieg

Der Einstieg selbst (Abb. 6.6) ist wie folgt gelöst. Über die Transition T1 können Personen aussteigen, solange es noch Personen im Fahrplanmittel gibt, die aussteigen möchten. Dabei werden die Personen über den aktuellen Ort informiert (`p_aus.position(fg_ein.Position->Haltestelle)`).

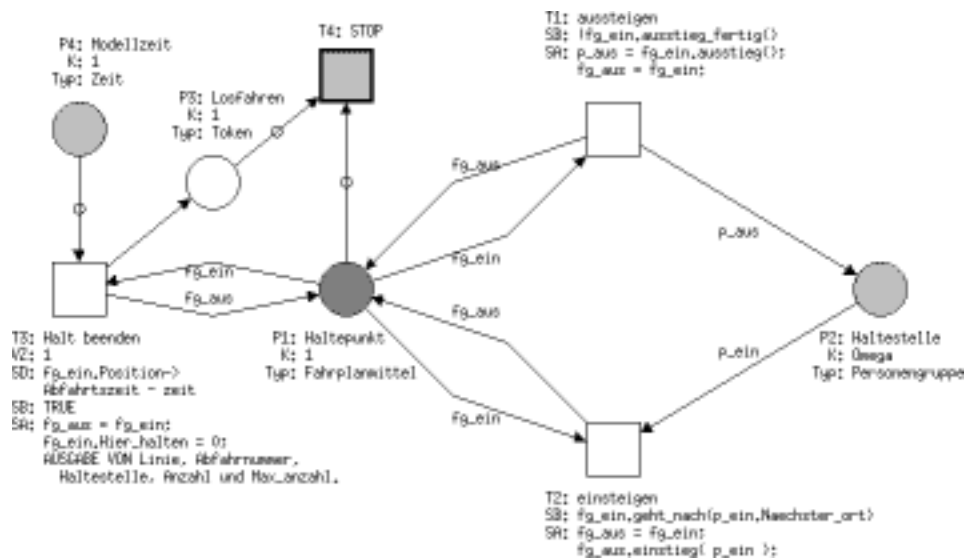


Abbildung 6.6: Unternetz für Aufruftransition in Abbildung 6.5

Über die Transition T2 können Personen einsteigen, falls das Fahrplanmittel zu ihrem nächsten Routenpunkt fährt. Wenn es keine Ein- und Aussteigevorgänge mehr gibt, schaltet die Transition T3 mit einer Verzögerung von einem Zeittakt und legt eine Marke auf Stelle P3. Dabei ist die Schaltdauer so formuliert,

daß das Schaltende nicht vor der Abfahrtszeit des Fahrplanmittels liegt². Mit der abgelegten Marke kann die Stoptransition T4 schalten, durch die das Unternetz beendet wird. Außerdem wird das Halten des Fahrplanmittels in die Ausgabedatei geschrieben.

Im Unterschied zur formalen Beschreibung des Simulationsmodells geschehen also alle Ein- und Aussteigevorgänge unmittelbar nach der Ankunft des Fahrplanmittels. Einen Zeittakt später können selbst dann keine Einstiege mehr stattfinden, falls neue Personen ankommen, da dann bereits T3 schaltet.

6.2.4 Quellen

Die Quellen für Personen (Abb. 6.7 links) sind so gestaltet, daß in den Quellenmarken auf Stelle P2 die Verkehrsnachfrage verwaltet wird. Die genaue Struktur der Klasse `Quelle` ist hier nicht dargestellt. Die Methoden `fahrtanzahl_jetzt` und `get_gruppe` sind so gestaltet, daß die Transition T1 bei jedem Schalten der Modellzeit die Möglichkeit hat, eine Personengruppe zu erzeugen und auf Stelle P3 abzulegen. Die Umsetzung von Verkehrsnachfrage auf erzeugte Personengruppen entspricht dabei einer Gleichverteilung über die Zeit (vgl. Abschnitt 4.5).

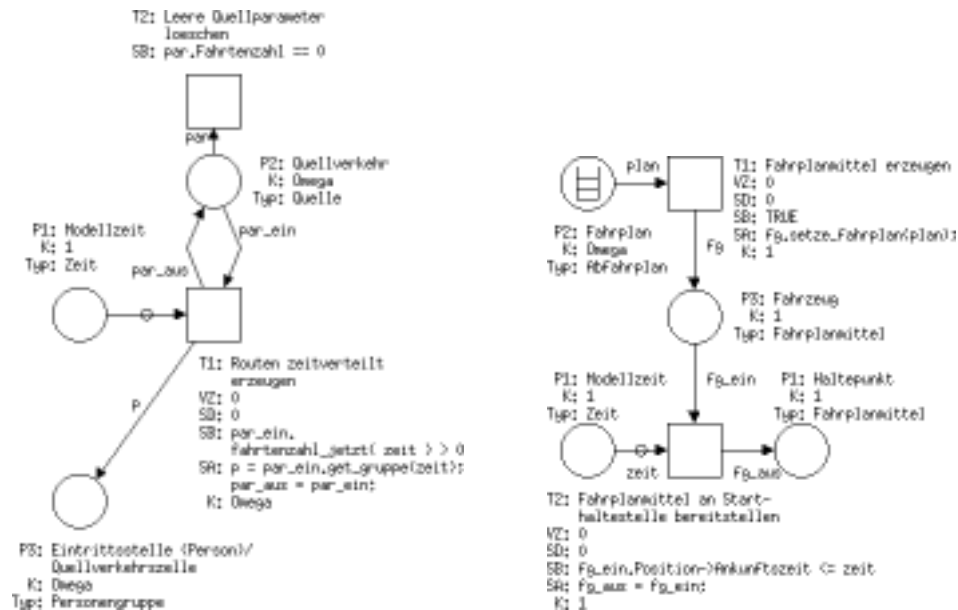


Abbildung 6.7: Quellen als THOR-Netze

Die Quellen von Fahrplanmitteln (Abb. 6.7 rechts) sind einfacher gestaltet. Auf Stelle P2 liegt für jeden Fahrtverlauf ein Abfahrplan (ohne Fahrgäste) als Marke. Transition T1 erzeugt daraus immer ein Fahrplanmittel. Von Stelle P3

²Der Differenzoperator für die Klasse `Zeit` wurde so implementiert, daß er eine nicht-negative ganze Zahl zum Ergebnis hat.

tritt das Fahrplanmittel über Transition T2 in das Verkehrsnetz ein, sobald es seine Ankunftszeit am ersten Routenpunkt erreicht hat.

6.2.5 Senken

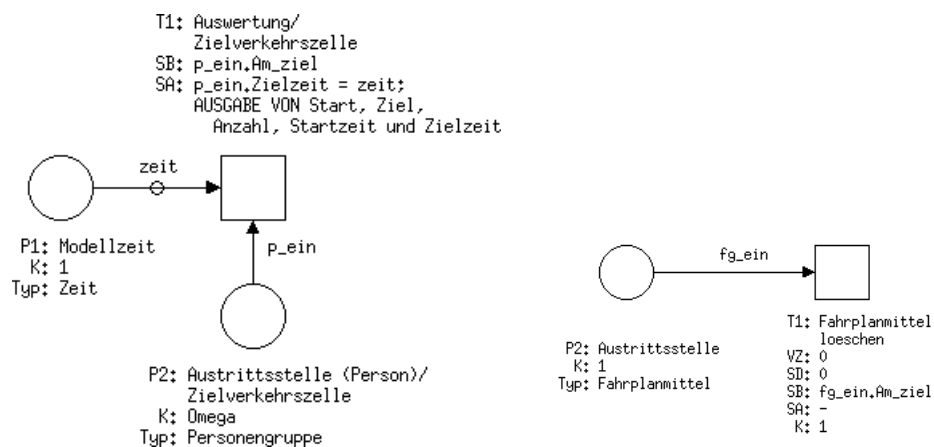


Abbildung 6.8: Senken als THOR-Netze

In Abbildung 6.8 werden die Senken dargestellt. An allen Stellen, die letzter Routenpunkt eines Fahrtverlaufs sind, muß eine Transition existieren, die die entsprechenden Verkehrsobjekte aus dem Verkehrsnetz nimmt. Für Personengruppen wird dabei zusätzlich der Fahrtverlauf in die Ausgabedatei geschrieben.

6.3 Abweichungen vom formalen Simulationsmodell

Die vorgestellte prototypische Implementierung meines Simulationsmodells weicht in folgenden Punkten von der formalen Spezifikation (Kapitel 3.2) entscheidend ab:

- Die Auswahl des schaltenden Ereignisses wird durch den THORN-Simulator zufällig ausgewählt.
- Die Modellzeit wird nur alle 60 Zeiteinheiten (Sekunden) erhöht. Deshalb kann die Einhaltung der Abfahrtszeit nicht exakt modelliert werden. Für die Fahrdauern spielt dies keine Rolle, da sie durch Schaltdauern realisiert sind.
- Die Mindestwartezeit ist nicht implementiert. Die Fahrplanmittel haben jedoch an allen Haltepunkten, die einer Haltestelle zugeordnet sind, eine Verzögerung von einer Zeiteinheit.
- Nur ein Teil der Informationen im Zustand wird tatsächlich in die Ausgabedatei ausgegeben.

6.4 Beschleunigung der Simulation

Der THORN-Simulator besitzt eine interne globale Uhr, um die Ereignisse zum richtigen Zeitpunkt stattfinden zu lassen. Für eine Verkehrssimulation des ÖPNV, bei dem Abfahrtszeitpunkte eine Rolle spielen, wäre es hilfreich, wenn diese Uhr zur Berechnung des Schaltzeitpunkts abgelesen werden könnte. Damit entfielen ein beachtlicher Simulationsaufwand für die Aktualisierung der Modellzeit. Nach der Erhöhung der Modellzeit müssen nämlich sämtliche Transitionen im Nachbereich auf das Auftreten neuer Ereignisse hin untersucht werden. Eine Stärke der Simulation von Petri-Netzen liegt aber eigentlich gerade in der lokalen Wirkung von schaltenden Ereignissen auf das Entstehen und Verschwinden anderer Ereignisse.

Man könnte sich sogar vorstellen, THOR-Netze auf einer parallelen Architektur mit gemeinsamem Adressraum zu simulieren. Jeder Prozessor simuliert dann nur ein paar Transitionen. Im Extremfall könnte jede Transition einen eigenen Prozessor bekommen. Es wird festgelegt, welche Zahl von Maschinentakten einer Zeiteinheit entspricht. Diese Zahl muß groß genug sein, daß die Durchführung der Schaltaktion normalerweise rechtzeitig abgewickelt ist und nur selten ein Interruptsignal zum Aussetzen der nächsten Zeiteinheit gesendet werden muß.

Bei OFFIS (Oldenburger Forschungs- und Entwicklungsinstitut für Informatik-Werkzeuge und Systeme) wird z.Zt. an einem verteilten Simulator von THOR-Netzen gearbeitet. Es ist jedoch so, daß die meisten Verkehrsnetze so stark zusammenhängen, daß der Kommunikationsaufwand für die Übertragung von Objekten den Vorteil der verteilten Simulation mehr als aufwiegen wird.

Kapitel 7

Simulationsergebnisse

Für das Beispiel Stuttgart wurde die in Kapitel 4 vorgestellte Modellierung durchgeführt. Die resultierenden getrennten U- und S-Bahn-Verkehrsnetze sowie das Fußwegenetz wurden dann in THOR-Netze umgesetzt. Diese Umsetzung ist in Kapitel 6 beschrieben. In Anhang B sind die getrennten Verkehrsnetze sowie die resultierenden THOR-Netze dargestellt. Dabei wurden die Verkehrsnetze für U- und S-Bahn im Fußwegenetz durch eine verfeinerte Transition realisiert, um die Übersichtlichkeit zu erhöhen. In den Netzen sind die Enabling-Kanten, in den Unternetzen für U- und S-Bahn-Netz sind auch die Hyperkanten, die von der Modellzeit ausgehen, nicht dargestellt, um die Übersichtlichkeit zu erhöhen.

Die Größe der betrachteten Liniennetze ist in folgender Tabelle dargestellt. Die Gesamtzahl der Haltestellen ist kleiner als die einfache Summe, weil es eine Anzahl gemeinsamer Haltestellen gibt.

Liniennetz	nur U-Bahn	nur S-Bahn	Gesamt
Haltestellen	97	70	158
Verbindungen	212	138	350

Unten ist die Größe der resultierenden THOR-Netze dargestellt. Das Unternetz „Mitnahme“ ist dabei das bekannte Unternetz aus Kapitel 6.2.3 zur Abwicklung von Aus- und Einstieg. Die Netze können während der Simulation auch erheblich größer werden, da jedes Fahrplanmittel im U- und im S-Bahn-Netz bei der Ankunft an einem Haltepunkt das jeweilige Mitnahme-Unternetz aufspannt.

THOR-Netz	U-Bahn	S-Bahn	Fußwege	Mitnahme	Gesamt
Transitionen	297	210	212	3	722
Stellen	661	535	231	4	1531
(N/Sh/St/IO)	(635/98/28/-)	(440/71/24/-)	(231/-/-/-)	(1/2/-/1)	
Enabling-Kanten	452	300	140	0	892
Hyper-Kanten	382	280	169	0	831
Standard-Kanten	947	676	356	0	1979
Aufruftransitionen	191	140	2	0	333
Stoptransitionen	0	0	0	1	1

N=Normal Sh=Share St=Stack IO=Ein-/Ausgabe

In den folgenden Abschnitten werden die Ergebnisse aus den ersten Simulationsläufen dargestellt. Die eingesetzten Fahrplanmittel sind bei allen Versuchen die gleichen. Der Fahrplan¹ wurde halbautomatisch, z.T. mit Hilfe von Shell-Scripten, in Abfahrpläne für jedes Verkehrsobjekt umgesetzt. Dabei entstanden

- 2210 Abfahrten im U-Bahn-Netz (mit Straßenbahn) und
- 587 Abfahrten im S-Bahn-Netz.

Wenn auch Personen modelliert wurden, ist dies dem jeweiligen Abschnitt zu entnehmen.

7.1 Kapazitätsprobleme

Bei der ursprünglichen Modellierung sollte jeder Haltepunkt die Kapazität eins besitzen, um ein Überholen auszuschließen. Dabei wird vorausgesetzt, daß der zeitliche Abstand zwischen zwei Fahrplanmitteln immer mindestens so groß wie die Fahrtdauer zwischen den Haltestellen ist.

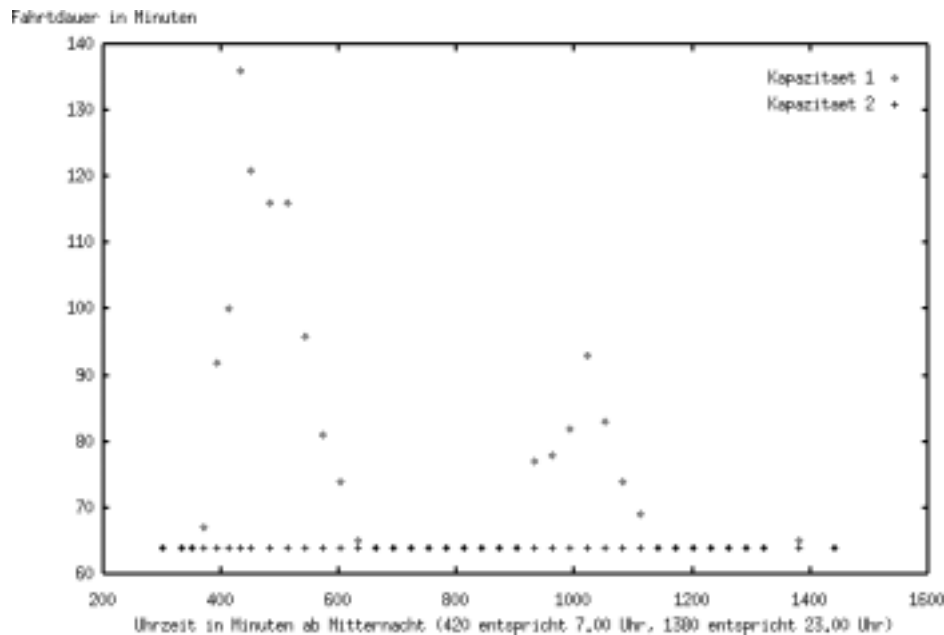


Abbildung 7.1: Verspätung der Linie S2 (Flughafen – Schorndorf)

Diese Voraussetzung erwies sich bereits im ersten Simulationslauf mit allen Fahrplanmitteln als falsch. Für die S-Bahn-Linie S2 ist in Abbildung 7.1 die Fahrtdauer (in Minuten) über der Abfahrtszeit (in Minuten ab 0.00 Uhr/ 480 $\hat{=}$ 8.00 Uhr) aufgetragen (mit Rauten). Man erkennt, daß insbesondere morgens, wenn alle Linien ihre kürzeste Taktzeit haben, erhebliche Verspätungen

¹Es wurde ein normaler Werktag ohne Schulferien zugrundegelegt.

eintreten. Der simulierte U-Bahn-Verkehr kam sogar vollständig zum Erliegen, weil die U-Bahnen in der Stadtmittle zyklisch aufeinander warteten, sobald auf jedem Haltepunkt des Zyklus eine U-Bahn stand.

Die Modellierung stimmt nicht mit der Wirklichkeit überein. Die S-Bahnen haben in den Stoßzeiten wohl weniger als eine Haltestelle Abstand, da die Verspätungen aus der Simulation im realen Fahrplan nicht eintreten.

Um trotzdem einige sinnvolle Simulationsläufe zu erhalten, habe ich die Kapazitäten aller Haltepunkte auf zwei erhöht. Das Resultat ist in obiger Abbildung durch kleine Kreuze dargestellt. Nach einer Verdoppelung der Kapazität traten keine Verspätungen mehr auf.

Durch die Verdoppelung der Kapazität können sich nun immer zwei Fahrplanmittel gleichzeitig auf einem Haltepunkt, einschließlich einführender Verbindungen, aufhalten. Für die Demonstration der Durchführbarkeit der Simulation genügt diese Modellierung. Für reale Anwendungen sollte jedoch in Zusammenarbeit mit den öffentlichen Verkehrsbetrieben versucht werden, die Kapazitäten so zu modellieren, daß sie die Wirklichkeit besser abbilden können.

7.2 Simulationsgeschwindigkeit

Es wurden noch keine Simulationsläufe mit einer großen Anzahl von Personen durchgeführt. In folgender Darstellung finden sich daher nur die Zahlen für die Simulation des gesamten U- und S-Bahn-Verkehrs für einen ganzen Werktag (0.00 Uhr bis ca. 1.30 Uhr am nächsten Morgen). Die Simulation wurde auf einer Workstation unter dem Betriebssystem SunOS 4.1.3 durchgeführt.

geschaltete Transitionen	davon Aufruf- transitionen	Zeitverbrauch		
		real	user	sys
197592	47071	6236.1 s	2101.7 s	747.8 s

Werden außer den Fahrplanmitteln noch Personen modelliert, so entstehen keine neuen zu schaltenden Aufruftransitionen. Die 47071 Mitnahmenetze, die als Unternetze aufgespannt und wieder beendet wurden, haben sicher entscheidenden Anteil an der benötigten Zeit. Ich gehe daher davon aus, daß auch eine realistische Anzahl von Personen simuliert werden kann.

Um diese Simulation durchführen zu können, wurden alle Kapazitäten der Haltepunkte auf zwei erhöht (vgl. Abschnitt 7.1).

7.3 Auswertung der Bewegung von Personen

Um die Mitnahme von Personen durch die Fahrplanmittel zu testen, habe ich als Demonstrationsbeispiel an der Haltestelle „Bietigheim-Bissingen“ eine Quelle für Personengruppen bereitgestellt, die zwischen 6.00 Uhr (360 Minuten nach Mitternacht) und 24.00 Uhr (1780 Minuten nach Mitternacht) 1000 Personen gleichmäßig auf folgende Rundfahrt schickt:

1. Stadtmitte (S5)
2. Untertürkheim (S1 oder U5)
3. Wilhelmsplatz (Straßenbahn 13)
4. Feuerbach Bf (weiter mit der Straßenbahn 13)
5. Bietigheim-Bissingen (S5)

In Abbildung 7.2 ist die Fahrtdauer (in Minuten) in Abhängigkeit von der Abfahrtszeit (in Minuten ab Mitternacht) aufgetragen. Man sieht, daß die Fahrtdauer einerseits vom Takt der S5 in Bietigheim-Bissingen (kleine Zacken), aber auch von der Qualität der Verbindungen abhängt (große Stufen). Am kürzesten (nämlich 105 Minuten) ist die Fahrt um 8.30 (510), 9.00 (540), 12.00 (720), 18.00 (1080) und 18.30 (1110). Die letzte Verbindung startet um 22.00 (1320) und braucht 165 Minuten.

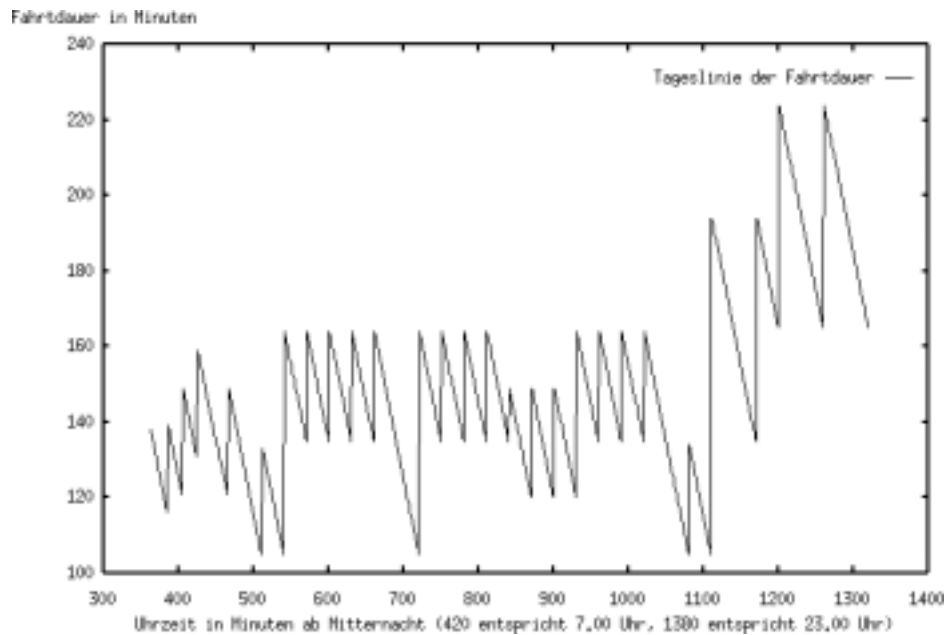


Abbildung 7.2: Tagesverlauf der Fahrtdauer für einen festgelegten Fahrtverlauf

Um diese Simulation durchführen zu können, wurden wiederum alle Kapazitäten der Haltepunkte auf zwei erhöht (vgl. Abschnitt 7.1). Daher fahren die Fahrplanmittel nach Fahrplan und haben keine betrieblich bedingten Verzögerungen. In realen Anwendungen können diese jedoch sehr wohl simuliert werden. Die Auswirkungen schlagen sich direkt in den Fahrtverläufen der Personen nieder.

7.4 Simulative Bestwegsuche

Liegt einmal ein Simulationsmodell für den öffentlichen Nahverkehr vor, so kann man die besten Wege sowie die Fahrtauern für diese Wege simulativ ermitteln.

Dazu sind folgende Änderungen am Simulationsmodell notwendig:

1. Es wird genau eine Person modelliert. Diese Person hat eine Starthaltestelle (1. Routenpunkt), an der sie zur Startzeit (Ankunftszeit am ersten Routenpunkt) in das Fußwegenetz eintritt.
2. Kommt ein Fahrplanmittel an einer Haltestelle vorbei, an der sich eine Person befindet, so wird eine Kopie dieser Person erzeugt, die in das Fahrplanmittel einsteigt. An der nächsten Haltestelle steigt die Person wieder aus.
3. Beim Ausstieg von Personen wird geprüft, ob sich schon eine Person auf der entsprechenden Haltestelle befindet. Ist das der Fall, so wird die aussteigende Person gelöscht. Andernfalls wird die Routenbeschreibung der aussteigenden Person um die aktuelle Haltestelle erweitert und die Ankunftszeit an dem neuen Routenpunkt gesetzt. Wegen Änderung Nr. 2 steigt natürlich unmittelbar danach eine Kopie der Person wieder in das gleiche Fahrplanmittel ein.

Am Schluß der Simulation befindet sich auf jeder Haltestelle eine Kopie der anfänglichen Person, die diese Haltestelle in minimaler Zeit erreichen konnte. Da jede Person, insbesondere auch Kopien, die Information über den Fahrtverlauf im Zustand gespeichert hat, erhält man alle kürzesten Wege einschließlich des zeitlichen Fahrtverlaufs ab der Starthaltestelle. Die simulative Bestwegsuche ähnelt der Berechnung kürzester Wege in einem Graphen nach Dijkstra.

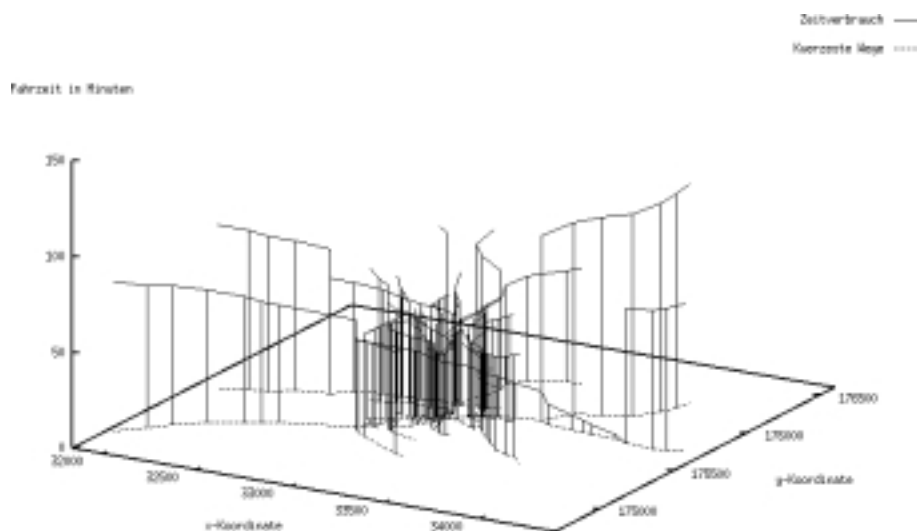


Abbildung 7.3: Darstellung der Bestwegberechnung ab Geradstetten

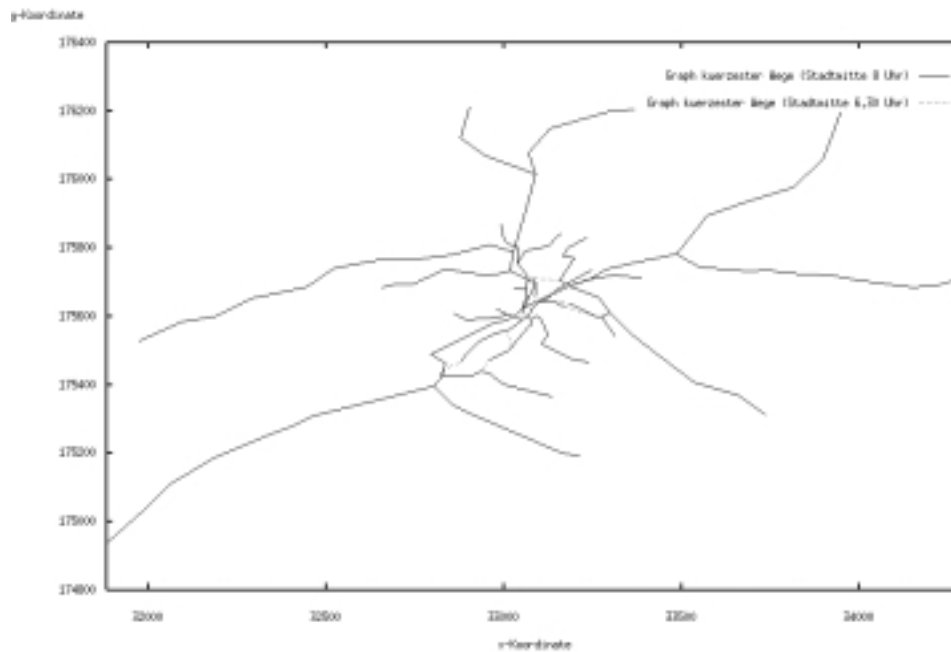


Abbildung 7.4: Unterschiedliche Bestweggraphen für verschiedene Uhrzeiten

In Abbildung 7.3 ist das Ergebnis einer simulativen Bestwegsuche dargestellt. Die z-Achse stellt die Zeitachse dar. Der Graph kürzester Wege ist einmal ohne (gestrichelt) und einmal mit der Fahrdauer (durchgezogen) dargestellt. Die langen Ausläufer der S-Bahn sind noch deutlich erkennbar. Der linke Ausläufer gehört zur Linie S2. Starthaltestelle für dieses Beispiel ist die Haltestelle „Geradstetten“ an der S2. Die simulierte Person tritt zum Zeitpunkt 4.48 (0) in das Fußwegenetz ein. In diesem Moment fährt die erste S-Bahn in Richtung Stuttgart ab. In der Abbildung erkennt man dies am sanften Ansteigen der zeitlichen Verlaufslinie aus der Startebene. Die erste S-Bahn in die Gegenrichtung fährt erst sehr viel später in Geradstetten ab. Dies wird an dem starken Zeitsprung sichtbar, der sich in der Gegenrichtung zeigt. Auch in Richtung Stuttgart muß unsere Person an der nächsten Verzweigung (Waiblingen) sehr lange auf die Linie S3 von Stuttgart nach Backnang (im Bild der rechte hintere Ausläufer) warten.

Ich konnte diese Art der Bestwegsuche nicht näher untersuchen, denke aber, daß sie sich qualitativ tatsächlich von herkömmlichen Verfahren unterscheidet. Das Vergleichen von Weglängen entfällt, weil die Kopie mit dem kürzeren Weg einfach zuerst an der jeweiligen Haltestelle ankommt.

In Abbildung 7.4 ist für ein weiteres Beispiel nur der Graph kürzester Wege dargestellt. Diesmal ist „Stadtmitte“ die Starthaltestelle. Die simulative Bestwegsuche wurde für zwei unterschiedliche Zeitpunkte durchgeführt. Man sieht, daß sich die beiden übereinandergelegten Graphen für unterschiedliche Zeitpunkte unterscheiden. Im Gegensatz zur statischen Definition des besten ÖV-Weges (vgl. Kapitel 2.3.2.3) hängt der Bestweg hier vom betrachteten Startzeitpunkt

ab.

Weitere Beispiele für die Auswertung simulativer Bestwegsuche finden sich in Anhang C. Dort erkennt man auch die Nachteile einer zentralistischen Vernetzung. Ist die Starthaltestelle eine Endhaltestelle, z.B. „Herrenberg“, so entsteht selbst zu benachbarten S-Bahn-Ausläufern eine große Reisedauer. Der Nachteil, der sich aus der Randlage für die Mobilität ohnehin ergibt, wird noch verstärkt.

Realisiert worden ist diese Bestwegsuche durch zwei kleine Änderungen am Prototyp. Die Methode `einstieg` wurde dahingehend abgeändert, daß die einsteigenden Personen automatisch am nächsten Haltepunkt der Fahrplanmittel wieder aussteigen. Das Unternetz für die Mitnahme mußte abgeändert werden. Statt der in Abschnitt 6.2.3 vorgestellten Lösung kam das Unternetz in Abbildung 7.5 zum Einsatz. Die aufrufenden großen THOR-Netze für das S- und das U-Bahn-Netz blieben unverändert.

Durch die Inhibitor-Kante von Stelle P2 zur Ausstiegstransition T1 ist sichergestellt, daß kein Ausstieg mehr möglich ist, nachdem die erste Kopie die Haltestelle erreicht hat. Kopien, die dadurch im Fahrplanmittel bleiben, werden am Ende der Fahrt mit dem Fahrplanmittel gelöscht. Wenn eine Kopie aussteigt, so wird die `Zielzeit` gesetzt und der aktuelle Ort, der sich sich aus der Position des Fahrplanmittels ergibt, an die Routenbeschreibung angefügt.

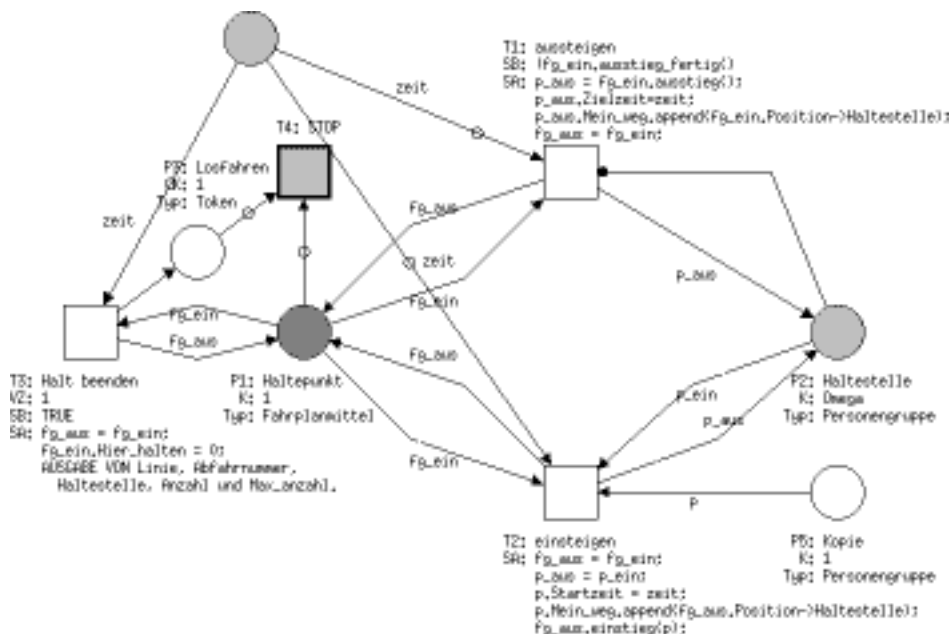


Abbildung 7.5: Das Unternetz für die Bestwegsuche

Die Schaltbedingung für die Einstiegstransition T2 entfällt. Gibt es eine Person auf P2 so wird diese nach dem Schalten der Transition auf P2 zurückgelegt. Stattdessen steigt eine Kopie in das Fahrplanmittel, die bei jedem Unternetzaufruf auf Stelle P5 bereitgestellt wird. Dabei wird die `Startzeit` gesetzt und der aktuelle Ort, der sich aus der Position des Fahrplanmittels ergibt, an die

Routenbeschreibung angefügt. Der Abzug der Kopie ist je Unternetzaufruf nur einmal möglich.

Natürlich könnte man auch die Bestwege von allen Startorten zu verschiedenen Zeiten gleichzeitig berechnen, da man in jeder Person sowohl Startort als auch Startzeit speichert. Dann gestaltet sich das eben beschriebene Unternetz ein wenig komplizierter, da auf jeder Haltestelle für jeden Startort und jede Startzeit eine Kopie ankommen muß. Verglichen mit dem Aufwand, die Bewegung von Hunderttausenden von Fahrgästen zu simulieren, handelt es sich jedoch um eine Kleinigkeit und erhöht die Simulationszeit nur unwesentlich.

7.5 Zusammenfassung der Ergebnisse

THOR-Netze haben sich bei der Simulation von öffentlichem Nahverkehr bewährt. Hier eine Zusammenfassung der Vorteile:

- + Simulator verfügbar
- + hohe Geschwindigkeit
- + Module sind austauschbar
- + Modellierung ist einfach

Dem stehen folgende Nachteile gegenüber:

- Verkehrssimulation ist (nur) Spezialfall (kein Zugriff auf interne Uhr möglich).
- Kanten können in der Darstellung normalerweise nicht ausgeblendet werden (Netze werden schnell unübersichtlich).
- Sehr großer Speicherbedarf bei der Übersetzung (für das U-Bahn-Netz ca. 220 MByte).
- Netzgröße beschränkt (U-Bahn-Netz war maximal groß²).

Ich vermute, daß die meisten Nachteile behoben werden können. Deshalb halte ich es für sinnvoll, die Idee der Verkehrssimulation mit THOR-Netzen weiterzuerfolgen. Dann müßten jedoch die THOR-Netz-Werkzeuge Editor, Compiler und Simulator in weiteren Arbeiten an die besondere Anwendung „Verkehrssimulation“ angepaßt werden.

²Das U-Bahn-Verkehrsnetz konnte als THOR-Netz erst kompiliert werden, nachdem zwei Haltestellen entfernt wurden.

Kapitel 8

Ausblick

Diese Arbeit stellt ein Simulationsmodell für die Abläufe im öffentlichen Verkehr vor. Durch die Entwicklung eines Prototyps mit THOR-Netzen konnten erste Anwendungsmöglichkeiten einer personenbezogenen Simulation demonstriert werden. Die weitere Arbeit besteht nun im wesentlichen darin,

- das Modell und den Prototyp weiterzuentwickeln,
- den bestehenden Prototyp ausgiebig zu testen und weitere Analysen auf den umfangreichen Simulationsergebnissen durchzuführen,
- zusammen mit Verkehrsbetrieben sinnvolle Anwendungsbeispiele zu erarbeiten und
- Metriken für die Qualität des ÖPNV formal zu spezifizieren.

Bei der Weiterentwicklung des Modells muß die Beschränkung der Kapazität von Verkehrsnetzen dichter an den betrieblichen Eigenschaften der realen Netze modelliert werden. Dabei geht es nicht nur um eine feinkörnigerere Abbildung, z.B. durch mehr Orte, sondern auch um neue Modellierungskonzepte wie z.B. Signalstrecken.

Ein verbesserter Prototyp sollte die Einhaltung der Abfahrtszeiten mit Hilfe einer internen Uhr regeln, damit die explizite Modellierung der Zeit entfallen kann. Außerdem sollte ein Editor entwickelt werden, der speziell auf Verkehrssimulation zugeschnitten ist.

Durch weitere Simulationen muß geprüft werden, wie sich die Laufzeiten ändern, wenn zusätzlich zu S- und U-Bahn-Verkehrsnetz noch einige 100.000 Personen simuliert werden. Dazu sollte auch das Bus-Verkehrsnetz noch integriert werden, das am IEUV in gleicher Form wie die anderen Netze vorliegt.

Bei der Entwicklung von Qualitätsmetriken denke ich insbesondere an die Bewertung des Vernetzungsgrades. Für alle Start-Ziel-Kombinationen in einem Untersuchungsraum könnte man z.B. den Quotienten aus kürzester Fahrtdauer und Luftlinienabstand bilden. Der Mittelwert ist dann ein Maß für die Vernetzung. Durch simulative Bestwegsuche kann der Tagesverlauf des Vernetzungsgrades mit nur einem Simulationslauf ermittelt werden.

Anhang A

Selbstdefinierte Marken – Deklaration

Um die prototypische Umsetzung des Simulationsmodells in THOR-Netze (Kapitel 6) vollständig verstehen zu können, sind hier Teile der Klassendeklarationen für die selbstdefinierten Marken abgedruckt. Insbesondere fehlen die geschützten Methoden `Load` und `Store`, die in jeder Klasse für THOR-Netz-Objekte definiert sein müssen.

```
/* -----  
    class Zeit (bzw. Time) - Header-Datei  
  
    Comment:      vom Oldenburger Verkehrsbeispiel "ubernommen  
  
    Author:       ??;  
    Date:         ??;  
    Description:  Uhrzeit (00:00:00-23:59:59) als Datentyp  
  
    Changelog:    04.04.96 double &operator/ eingefuehrt,  
                  stattdessen Time &operator/ geloescht.  
                  13.05.96 &operator- geaendert (von Time auf int).  
* ----- */  
  
class Time : public IOABLE  
{  
private:  
    long seconds;  
  
public:  
    Time();  
    Time(long ss);  
    Time(int hh,int mm,int ss);  
  
    int hour() const;  
    int minute() const;  
    int second() const;  
  
    int Stunde() const { return hour(); }  
    int Minute() const { return minute(); }  
    int Sekunde() const { return second(); }  
  
    String Ausgabe() const;  
  
    friend Time operator+( const Time &x, const long &i );  
    friend int operator-( const Time &x, const Time &y ); \\ Ergebnis immer > 0  
};  
typedef Time Zeit;
```

```

/* -----
   class Ort - Header-Datei

   Comment:

   Author:      Markus Schaal
   Date:        28.03.1996
   Description:  Eigentlich nur ein String, aber ...

   Changelog:   10.6.1996 F"ur eine neue Version des
                  THORN-Compilers konnte der
                  Code durch typedef ersetzt werden
* ----- */

typedef String Ort;

/* -----
   class Route - Header-Datei

   Comment:

   Author:      Markus Schaal
   Date:        03.04.1996
   Description:  kapselt die Routeninformation f"ur die
                  Erzeugung von Personen.

   Changelog:
* ----- */

class Routenpunkt
{
public:
    Ort      Name;      // Ortsname des Routenpunkts
    Routenpunkt *next; // n"achster Routenpunkt (Default: NIL)

    Routenpunkt()
    { next = NULL; }

    Routenpunkt( Ort o )
    { Name = o; next = NULL; }
};

class Route : public IOABLE
{
public:
    Routenpunkt *first; // pointer to the list of entries
    Routenpunkt *last;  // pointer to the list of entries

    Route() // inline ok
    { first = NULL; last = NULL; }
    Route(const Route &r); // Shallow Copy wg. Effizienz
    ~Route();

    void append( Routenpunkt *re );
};

```



```

/* -----
   class Abfahrplan - Header-Datei

   Comment:

   Author:      Markus Schaal
   Date:        28.03.1996
   Description: modelliert den Abfahrplan fuer "offentliche
                 Verkehrsmittel. Listen werden dabei explizit
                 programmiert und machen die Klassen
                 Halt und Fahrgast als Listenelemente
                 notwendig.
   * ----- */

class Fahrgast
{
public:
    Personengruppe    Gast;    // Personengruppe als Element der Fahrgastliste
    Fahrgast*         next;    // Verweis auf das n"achste Element der
                               // Fahrgastliste

    Fahrgast() { next = NULL; } // inline ok
};

class Halt
{
public:
    Ort                Haltepunkt;    // name des Haltepunkts
    Ort                Haltestelle;   // Name der Haltestelle
    Halt               *next;         // Zeiger auf das n"achste Element in der
                                     // Haltliste

    Zeit               Ankunftszeit;  // fr"uehster Ankunftszeitpunkt am Halt
    Zeit               Abfahrtszeit;  // sp"atester Abfahrtszeitpunkt am Halt

    Fahrgast*         Fahrgastliste;  // Zeiger auf den ersten Fahrgast mit dieser
                                     // Zielhaltestelle

    Halt()
    { Ankunftszeit = 0; Abfahrtszeit= 0; next = NULL;
      Fahrgastliste = NULL; }

    Halt( Ort punkt, Ort stelle, Zeit ab, Zeit an )
    { Haltepunkt = punkt; Haltestelle = stelle;
      Ankunftszeit = an; Abfahrtszeit = ab; next = NULL;
      Fahrgastliste = NULL; }
};

class Abfahrplan : public IOABLE
{
public:
    int                Max_anzahl;    // Maximale Zahl der Fahrgaeste im
                                     // eingesetzten Fahrplanmittel
    String             Linie;         // Linienbezeichnung
    int                Abfahrnummer;  // Abfahrnummer der Linie
    Halt               *first;        // Zeiger auf den ersten Halt
    Halt               *last;         // Zeiger auf den letzten Halt

    Abfahrplan()
    { first = NULL; last = NULL; Max_anzahl = 50;
      Linie = "S1"; Abfahrnummer = 1; }
    Abfahrplan(const Abfahrplan &af); // Shallow-Copy
    ~Abfahrplan();

    void append( Halt *re );
};

```

```

/* -----
class Zeit Fahrplanmittel - Header-Datei

    Author:      Markus Schaal
    Date:        28.03.1996
    Description: modelliert "offentliche Verkehrsmittel, die nach
                  Fahrplan fahren.
* ----- */

class Fahrplanmittel : public IOABLE
{
public:
    Abfahrplan    Mein_fahrplan; //
    Halt*         Position;      // Zeiger auf aktuellen Halt
    int           Anzahl;        // Anzahl der Fahrgaeste
    bool          Am_ziel;       // TRUE, wenn das Ziel erreicht ist
    bool          Hier_halten;    // TRUE, wenn der naechste Haltepunkt
                                // im Verkehrsnetz ein Haltepunkt des
                                // Fahrplanmittels ist.
    Ort           Naechster_halt; // Name des n"achsten Haltepunkts

    Fahrplanmittel()             // inline ok
    { Position = NULL; Am_ziel = 0; Hier_halten = 1;
      Anzahl = 0; }

    Fahrplanmittel(const Fahrplanmittel &v); // Copy-Konstruktor
                                              // (Achtung: shallow-copy)

    void setze_fahrplan(Abfahrplan ap);
    void position(Ort o);                // informiert Fahrplanmittel
                                          // ueber aktuelle Position

    void einstieg(Personengruppe p);
    Personengruppe ausstieg();
    bool ausstieg_fertig() const;
    bool geht_nach(Ort o) const;         // liefert TRUE zurueck, falls
                                          // o der Name einer Halte-
                                          // stelle in Fahrtrichtung ist.
};

/* -----
class Streckenabschnitt - Header-Datei

    Author:      Markus Schaal
    Date:        28.03.1996
    Description: kapselt die notwendigen Parameter eines
                  Streckenabschnitts.
* ----- */

class Wegparameter : public IOABLE
{
public:
    Wegparameter()                 // inline ok
    { Zeitverbrauch_normal = 300; Standardabweichung = 0; }

    int wegzeit() const;           // Wegzeitberechnung
    Ort zwischenziel() const      // inline ok
    { return Nach_ort; }

private:
    int    Zeitverbrauch_normal;
    int    Standardabweichung;
    Ort    Nach_ort;
    Ort    Von_ort;
};

```

Anhang B

Verkehrsgraphen und THOR-Netze

Auf den folgenden Seiten sind die THOR-Netze für das Simulationsbeispiel Stuttgart abgebildet. Dabei sind für das U- und S-Bahn-Verkehrsnetz keine Kanten zur Modellzeit (Stelle P1) dargestellt. Die Enabling-Kanten, die von der Modellzeit ausgehen, sind auch im Fußwegenetz nicht dargestellt. Am Fußwegenetz (Abb. B.1) sieht man sehr gut, wie sehr die Übersichtlichkeit der Netze leidet, wenn von einer Transition oder Stelle eine große Anzahl von Kanten ausgeht. Hierbei handelt es sich um die Hyperkanten, die die Haltestellen mit den Transitionsverfeinerungen für das U- und S-Bahn-Verkehrsnetz verbinden.

Zusätzlich zu den THOR-Netzen sind auch die jeweiligen getrennten Verkehrsgraphen abgebildet.

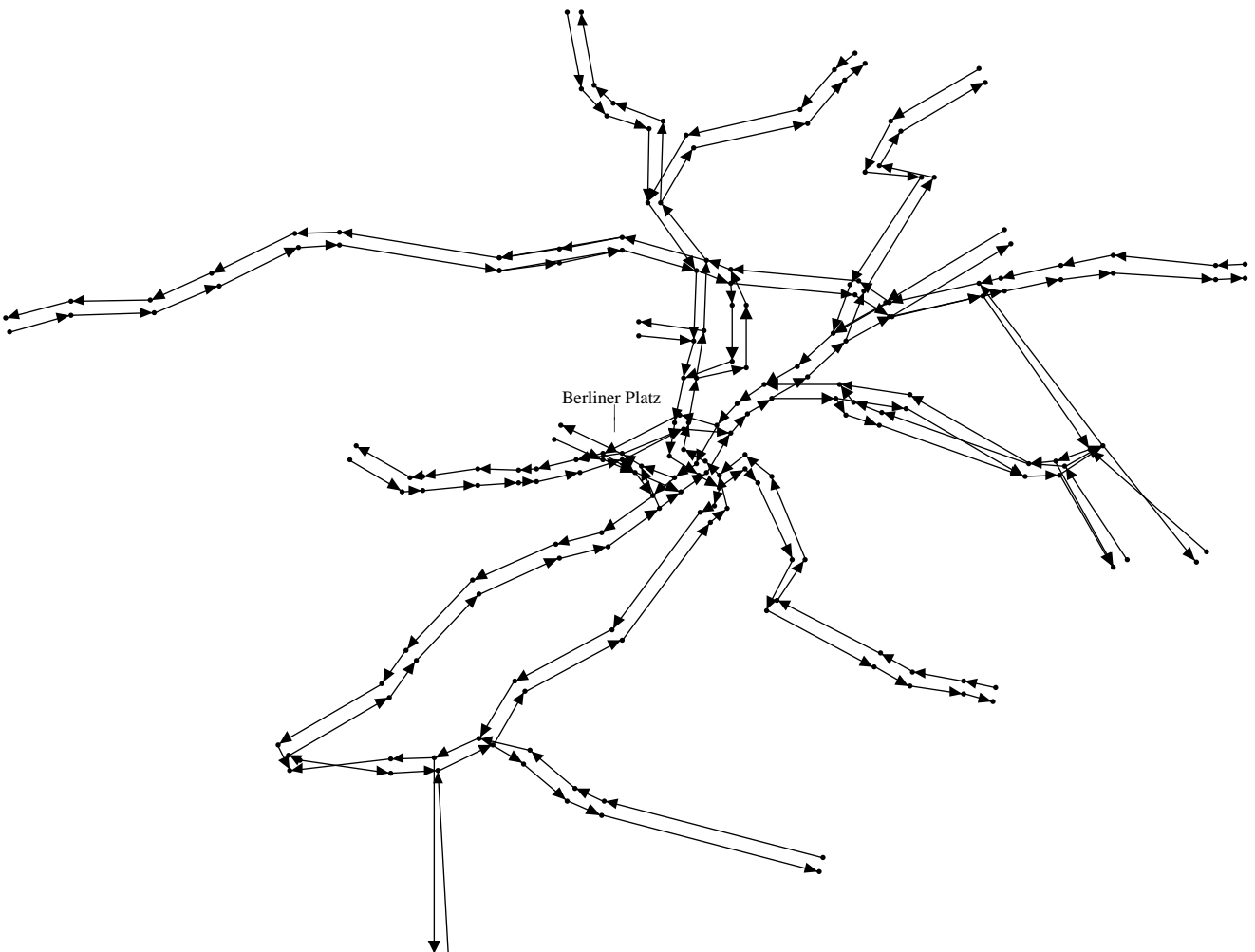


Abbildung B.2: Das getrennte U-Bahn-Verkehrsnetz



Abbildung B.3: U-Bahn-Verkehrnetz als THOR-Netz (mit Quellen & Senken)

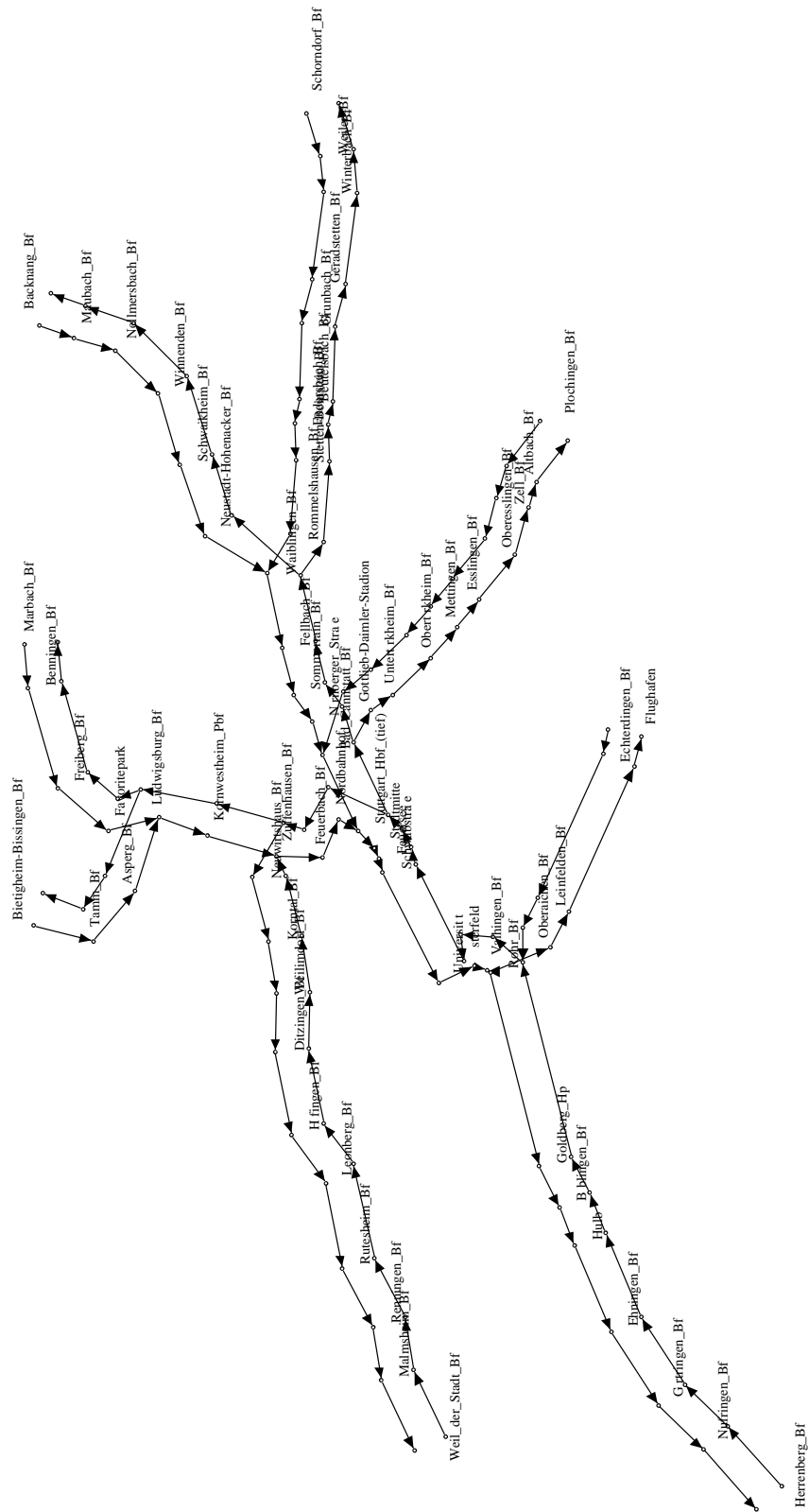


Abbildung B.4: Das getrennte S-Bahn-Verkehrsnetz

Anhang C

Bilder zur simulativen Bestwegsuche

Hier sind noch einige 3-dimensionale Darstellungen der simulativen Bestwegsuche. Ich denke, daß diese Art der Darstellung ein gutes Mittel ist, die Qualität von öffentlichen Verkehrsnetzen darzustellen. Man sieht nämlich sofort, wenn ein nahes Ziel nur mit einer großen Reisedauer erreicht werden kann.

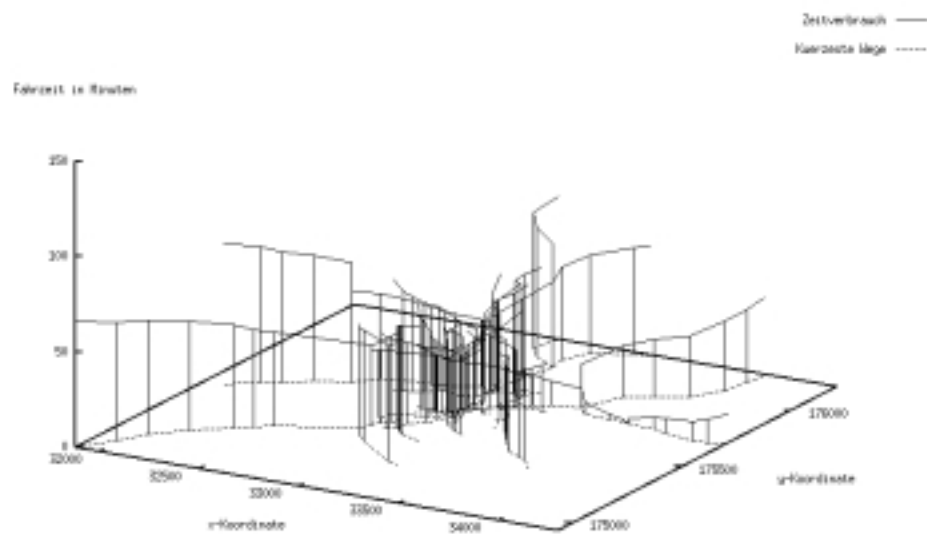


Abbildung C.1: Starthaltestelle „Beutelsbach“, Abfahrt 8.00 Uhr

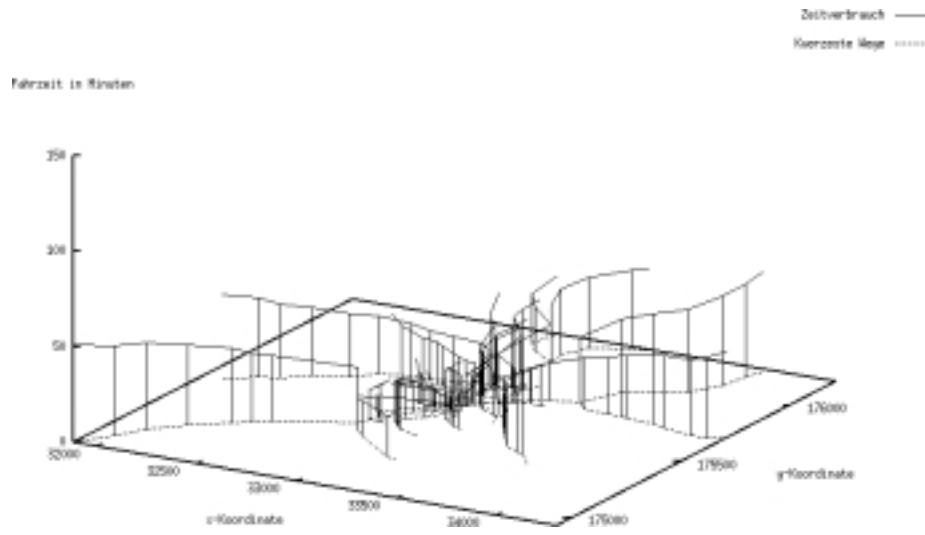


Abbildung C.2: Starthaltestelle „Stadtmitte“, Abfahrt 6.30 Uhr

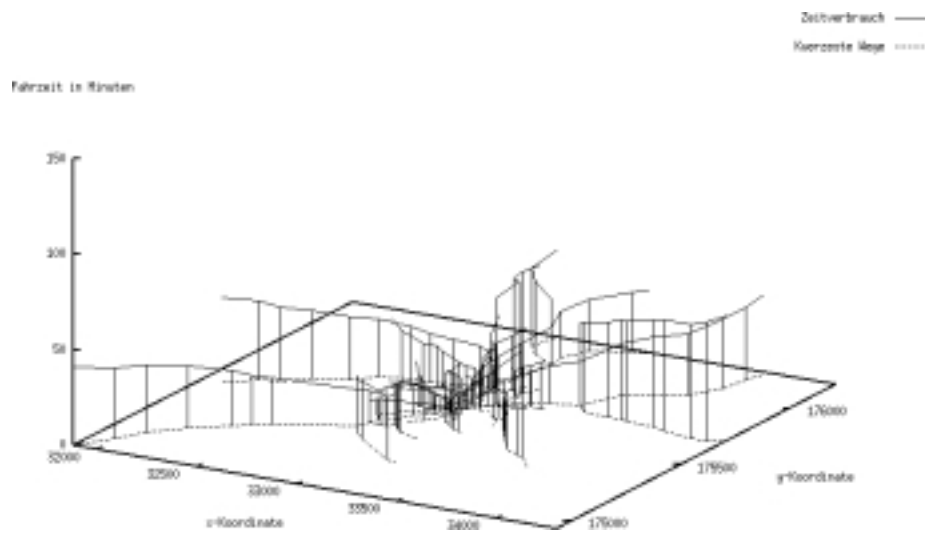


Abbildung C.3: Starthaltestelle „Stadtmitte“, Abfahrt 8.00 Uhr

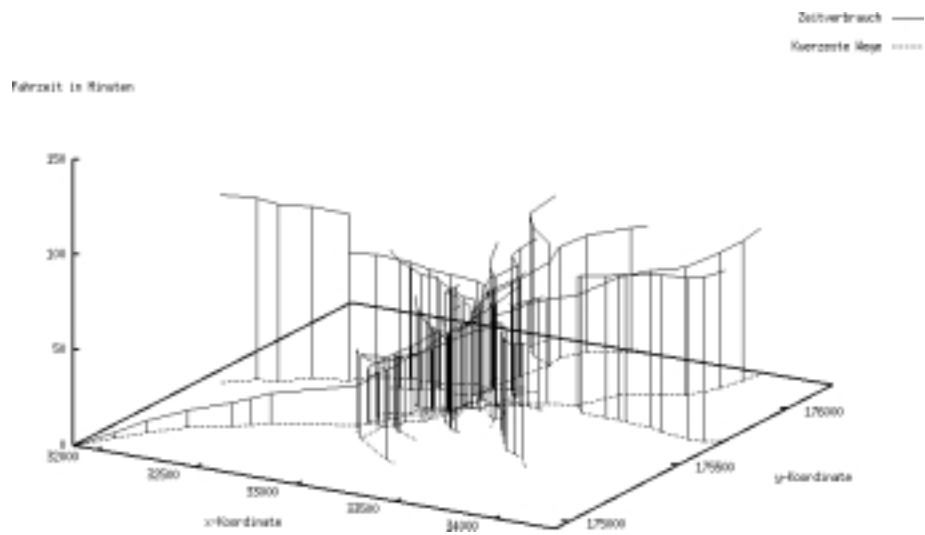


Abbildung C.4: Starthaltestelle „Herrenberg“, Abfahrt 6.46 Uhr

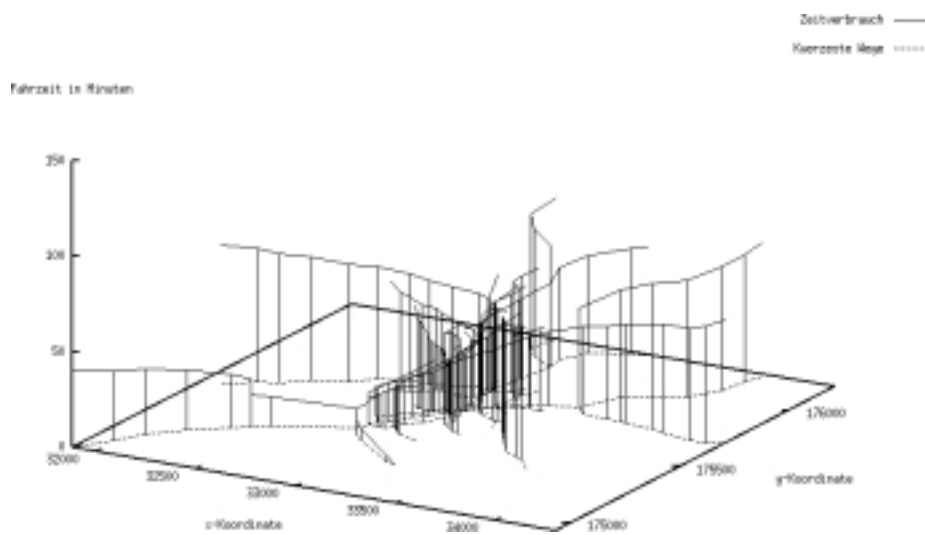


Abbildung C.5: Starthaltestelle „Flughafen“, Abfahrt 8.01 Uhr

Anhang D

Abkürzungen

IEUV = Institut für Eisenbahn- und Verkehrswesen (an der Universität Stuttgart)

IV = Individualverkehr

MFG = Mitfahrgemeinschaft

MIV = Motorisierter Individualverkehr

OFFIS = Oldenburger Forschungs- und Entwicklungsinstitut für Informatik-Werkzeuge und Systeme

ÖPNV = Öffentlicher Personennahverkehr

ÖV = Öffentlicher Verkehr

THORN/THOR-Netz = Timed Hierarchical Object-Related Net

VVS = Verkehrs- und Tarifverbund Stuttgart

Literaturverzeichnis

- [Bau90] Bernd Baumgarten. *Petri-Netze : Grundlagen und Anwendungen*. BI-Wiss.-Verl., Mannheim, 1990.
- [DKGL94] Frank DiCesare, Paul T. Kulp, Michael Gile, and George List. The Application of Petri Nets to the Modeling, Analysis and Control of Intelligent Urban Traffic Networks. In *Application and Theory of Petri Nets 1994*, pages 2–15, 1994.
- [DSH95] Harry Dobeschinsky, Friedrich Schedel, and Jürgen Hoffmann. Verkehrsprognose im öffentlichen Verkehr. Vorlesungsskript, Institut für Eisenbahn- und Verkehrswesen, Stuttgart, WS 1994/95.
- [FHKP93] Holger Fender, Albrecht Hönerloh, Jochen Kuhl, and Stephan Preuß. Anwendung von PACE auf ein Problem der Verkehrsplanung. In Jörg Biethahn, Wilhelm Hummeltenberg, Bernd Schmidt, and Thomas Witte, editors, *Simulation als betriebliche Entscheidungshilfe*, chapter 16, pages 233–247. Vieweg Verlag, 1993.
- [Fis95] Paul A. Fishwick. *Simulation model design and execution : building digital worlds*. Prentice Hall international series in industrial and systems engineering. Prentice Hall, Englewood Cliffs, NJ, 1995.
- [FLSW93] Hans Fleischhack, Ulrike Lichtblau, Michael Sonnenschein, and Ralf Wieting. Generische Definition {hierarchischer} {zeitbeschrifteter} {höherer} Petrinetze. Technical Report Nr. AIS-13, Arbeitsgruppe Informatik-Systeme, FB Informatik, Universität Oldenburg, Dezember 1993.
- [Hen76] Hartmut Hensel. *Wörterbuch und Modellsammlung zum Algorithmus der Verkehrsprognose*. Institut für Stadtbauwesen RWTH Aachen, 1976.
- [Hil84] Thilo Hilpert. *LeCorbusiers 'Charta von Athen' : Texte u. Dokumente*. Vieweg, Braunschweig, 1984.
- [HMD⁺] Gerhard Heimerl, Werner Meier, Harry Dobeschinsky, Hans-Ulrich Mann, and Bettina Götz. Standardisierte Bewertung von Verkehrsweginvestitionen des öffentlichen Personennahverkehrs. Erstellt im Auftrag des Bundesministers für Verkehr.

- [Hof94] Jürgen Hoffmann. Entwicklung eines Programmsystems zur Wegsuche in Verkehrsnetzen des Öffentlichen und Individualverkehrs unter besonderer Berücksichtigung des P+R-Verkehrs. Diplomarbeit, Fakultät Informatik, Univ., Stuttgart, 1994. Diplomarbeit Nr.1094.
- [Mäc64] P.A. Mäcke. *Das Prognoseverfahren in der Straßenverkehrsplanung*. Bauverlag GmbH, Wiesbaden Berlin, 1964.
- [Men75] Rudolf Menke. *Stadtverkehrsplanung: Ein neues Konzept für die städtische Generalverkehrsplanung*. Kohlhammer, Stuttgart, 1975.
- [Opp95] Norbert Oppenheim. *Urban Travel Demand Modeling*. John Wiley & Sons, Inc., New York, 1995.
- [Pre93] Stephan Preuß. Simulation eines Verkehrsnetzes auf der Basis von Petri-Netzen. Diplomarbeit, Lehrstuhl für Wirtschaftsinformatik, Universität Göttingen, 1993.
- [Rei86] Wolfgang Reisig. *Petrinetze : eine Einführung*. Studienreihe Informatik. Springer, Berlin [u.a.], 2., überarb. u. erw. edition, 1986. IX.
- [Sch90] Friedrich Schedel. Programmsystem zur Durchführung von Umlage-rechnungen in Netzen des öffentlichen Personennahverkehrs und deren Darstellung auf postscriptfähigen Laserdrucker. Diplomarbeit, Fakultät Informatik, Univ., Stuttgart, 1990. Diplomarbeit Nr. 760.
- [Sch95] Werner Schumacher. Vergleichende Untersuchung von Verkehrsmittelwahlmodellen mit Erweiterung der bisherigen bimodalen Modelle. Diplomarbeit, Fakultät Informatik, Univ., Stuttgart, 1995. Diplomarbeit Nr. 1286.
- [Sta73] Herbert Stachowiak. *Allgemeine Modelltheorie*. Springer, Wien[u.a.], 1973.
- [Ste94] Gerd Steierwald. *Stadtverkehrsplanung : Grundlagen, Methoden, Ziele*. Springer-Verlag Berlin Heidelberg, 1994.
- [Tan89] Andrew S. Tanenbaum. *Computer networks*. Prentice-Hall, Englewood Cliffs, NJ, 2. ed. edition, 1989.
- [Ves90] Frederic Vester. *Ausfahrt Zukunft*. Wilhelm Heyne Verlag, München, 1990.
- [Ves95] Frederic Vester. *Crashtest Mobilität, Die Zukunft des Verkehrs, Fakten – Strategien – Lösungen*. Wilhelm Heyne Verlag, München, 1995.
- [Wer94] M Wermuth. Modellvorstellungen zur Prognose. In Gerd Steierwald and Hans-Dieter Künne, editors, *Stadtverkehrsplanung : Grundlagen, Methoden, Ziele*, pages 221–274. Springer-Verlag Berlin Heidelberg, 1994.

- [Wie95a] Ralf Wieting. *Handbuch zur THORN Entwicklungsumgebung*. OF-FIS, Oldenburg, Dezember 1995.
- [Wie95b] Ralf Wieting. Verkehrsmodellierung mit THOR-Netzen. In *Workshop Algorithmen und Werkzeuge für Petrinetze der GI-Fachgruppe 0.0.1*, Oldenburg, Okt. 1995.
- [WLD93] Huaitien Wang, George F. List, and Frank DiCesare. Modeling and Evaluation of Traffic Signal Control Using Timed Petri Nets. In *IEEE International Conference on Systems, Man and Cybernetic*, volume 2, pages 180–185, 1993.
- [Zei95] Thomas Zeitler. Vergleichende Untersuchung von Modellen zur Verkehrserzeugung und Verkehrsverteilung. Studienarbeit, Fakultät Informatik, Univ., Stuttgart, 1995. Studienarbeit Nr.1432.

Erklärung

Hiermit versichere ich, diese Arbeit
selbständig verfaßt und nur die
angegebenen Quellen benutzt zu haben.

(Markus Schaal)

