TECHNISCHE UNIVERSITÄT BERLIN – FACHBEREICH INFORMATIK

Diplomarbeit

Zeitliche Ausdrücke in Terminvereinbarungsdialogen: Repräsentation und Inferenz

Ulrich Endriß

Juni 1998

Aufgabensteller:
Prof. Bernd Mahr
Dr. Wilhelm Weisweber

Kurzfassung Bei Terminvereinbarungsdialogen spielen zeitliche Ausdrücke naturgemäß eine herausragende Rolle. Ausgehend von einer ausführlichen Analyse der im Rahmen des VERBMOBIL-Projekts auftretenden sprachlichen Phänomene wird eine Repräsentationssprache für zeitliche Ausdrücke, die TEMPORAL EXPRESSION LANGUAGE, entwickelt. Die Definition einer formalen Semantik dieser Sprache bildet den Kern der Arbeit. Darauf aufbauend werden verschiedene Inferenzen über zeitlichen Ausdrücken erklärt. Schließlich wird eine Implementierung des beschriebenen Systems dokumentiert.

Danksagung Ich danke Bernd Mahr und Wilhelm Weisweber für die Stellung einer interessanten Aufgabe für meine Diplomarbeit. Darüber hinaus gilt mein besonderer Dank Manfred Stede und Uwe Küssner für ihre intensive Betreuung der Arbeit, für zahlreiche Diskussionen und die Beantwortung vieler Fragen.

Selbständigkeitserklärung Hiermit erkläre ich, die vorliegende Arbeit selbständig verfaßt und neben den angegebenen Quellen keine weiteren Hilfsmittel benutzt zu haben.

Berlin, 19. Juni 1998 Ulrich Endriß

Inhaltsverzeichnis

Kapitel 1

Einleitung

Die Zeit ist aus den Fugen ...
William Shakespeare: Hamlet, Prinz von Dänemark.

Man stelle sich vor, zwei Geschäftspartner unterschiedlicher Muttersprachen mit nur geringen Englischkenntnissen führen eine Verhandlung. Einen Teil der Konversation bewältigen sie ohne fremde Hife in englischer Sprache, in schwierigen oder kritischen Phasen steht ihnen jedoch ein Dolmetscher zur Seite. Man stelle sich weiter vor, dieser Dolmetscher ist kein Mensch, sondern ein Gerät, das in der Lage ist, dem Dialog zu folgen und bei Bedarf die Übersetzung einzelner Redebeiträge zu liefern. In dieser allgemeinen Form ist diese Vorstellung momentan wohl noch recht visionär. Auf dem langen Weg hin zu einer möglichen Realisierung bietet sich eine Einschränkung des Gesprächsgegenstandes an, zum Beispiel auf die Absprache eines Termins.

Bei solchen Terminvereinbarungsdialogen spielen zeitliche Ausdrücke naturgemäß eine herausragende Rolle. Ziel dieser Arbeit ist es, eine systematische Art der Repräsentation solcher Ausdrücke zu finden und diesen eine formale Semantik zuzuordnen. Dies geschieht im Hinblick auf die spätere Integration in ein Übersetzungssystem der beschriebenen Art.

1.1 Zeitliche Ausdrücke im VERBMOBIL-Szenario

Das Verbundvorhaben VERBMOBIL [Wahlster, 1993] ist der Übersetzung spontan gesprochener Sprache bei Verhandlungsdialogen gewidmet. In der ersten Phase des Projekts war die Gesprächsdomäne dabei auf die Vereinbarung eines Termins zwischen zwei Verhandlungspartnern beschränkt. In der zweiten Phase ist sie auf die Planung einer gemeinsamen Geschäftsreise erweitert worden. Das an der Technischen Universität Berlin angesiedelte VERBMOBIL-Teilprojekt beschäftigt sich mit der semantischen Auswertung für die Übersetzung [Quantz *et al.*, 1997]. Hier hat sich die Notwendigkeit ergeben, den zeitlichen Ausdrücken eine besondere Behandlung zukommen zu lassen.

Als zeitliche Ausdrücke werden diejenigen Teile eines Redebeitrages bezeichnet, die einen zeitlichen Aspekt, also etwa die Dauer eines Treffens oder einen bestimmten Zeitpunkt, beschreiben. Zur Repräsentation soll von den spezifischen natürlichsprachlichen Formen[1] dieser Ausdrücke so weit abstrahiert werden, daß eine einheitliche Darstellung über Sprachgrenzen hinweg möglich ist. Auf der anderen Seite ist aber auch eine gewisse Sprachnähe beabsichtigt, um die Generierung einer Übersetzung in die Zielsprache zu erleichtern. Um einen Gesprächsverlauf nachvollziehen zu können, das heißt, um den Stand einer Terminvereinbarung zu jeder Zeit während des Dialogs wiedergeben zu können, müssen verschiedene Inferenzen über zeitlichen Ausdrücken möglich sein. Dies soll durch die Definition einer formalen Semantik für zeitliche Ausdrücke erreicht werden.

1.2 Aufbau der Arbeit

Kapitel 2 beginnt mit einer Korpusanalyse, das heißt mit einer Klassifizierung der in VERBMOBIL vorkommenden zeitlichen Ausdrücke. Mit Hilfe der Sprache ZEITGRAM [Küssner und Stede, 1995] können die meisten dieser Ausdrücke bereits repräsentiert werden, es gibt allerdings noch Lücken. Diese werden ebenfalls in Kapitel 2 analysiert. Kapitel 3 gibt einen Überblick über diverse andere Möglichkeiten zur Behandlung temporaler Information. Die in der Literatur gefundenen Ansätze werden nach ihrem Nutzen in Bezug auf die Entwicklung einer inferenztauglichen Repräsentationssprache für zeitliche Ausdrücke bewertet. Es kann hier vorweggenommen werden, daß die Literraturrecherche in dieser Hinsicht keine wesentlichen Erkenntnisse erbringen konnte, weshalb die Korpusanalyse zusammen mit der Analyse von ZEITGRAM die wesentliche Grundlage für das weitere Vorgehen bildet.

Die Syntax einer neuen Repräsentationssprache für temporale Ausdrücke, der TEMPORAL EXPRESSION LANGUAGE, wird in Kapitel 4 entwickelt. Diese Sprache weist einige Redundanzen auf, was vor allem in der – beabsichtigten – Nähe zur natürlichen Sprache begründet liegt. Um nicht für unnötig viele Einzelausdrücke eine formale Semantik angeben zu müssen, wird die TEMPORAL EXPRESSION LANGUAGE in Kapitel 5 zunächst auf eine sogenannte Kernsprache reduziert, für die Kapitel 6 dann die Semantik definiert. In Kapitel 7 wird gezeigt, wie anhand der Semantik einige Inferenzen auf zeitlichen Ausdrücken erklärt werden können. Das beschriebene System wurde größtenteils in ein PROLOG-Programm umgesetzt. Die Implementierung dieses Prototyps wird in Kapitel 8 beschrieben. Das Schlußkapitel 9 gibt unter anderem einen Ausblick auf Möglichkeiten zur Fortführung des Projekts.

In Anhang A werden zahlreiche Beispiele für zeitliche Ausdrücke bei Terminvereinbarungsdialogen aufgeführt. Sie entstammen dem VERBMOBIL-Korpus, einer umfangreichen Sammlung von Transliterationen spontan gesprochener Dialoge. Anhang B stellt die Grammatiken der betrachteten formalen Sprachen zusammenfassend dar. Daran anschließend ist das Verzeichnis der verwendeten Literatur abgedruckt.

[1] Die berücksichtigten Sprachen sind Deutsch und Englisch.

Kapitel 2

Analyse

Wer den Weg gehen will,
muß den Weg finden und ihn dann gehen.
Zen-Kôan

Im Rahmen dieser Arbeit soll eine formale Repräsentationssprache für natürlichsprachliche zeitliche Ausdrücke entwickelt werden. Insbesondere soll diese Sprache zur Repräsentation des temporalen Gehalts von Äußerungen aus Terminvereinbarungsdialogen, wie sie in VERBMOBIL vorkommen, benutzt werden. Eine Auswahl solcher Äußerungen ist in Anhang A abgedruckt. Dieses Korpus wird die wesentliche Quelle der Inspiration für die Repräsentationssprache sein und soll im ersten Teil dieses Kapitels analysiert werden.

In [Küssner und Stede, 1995] wurde bereits eine Sprache mit der gleicher Zielsetzung definiert. Sie nennt sich ZEITGRAM. Größere Teile von ZEITGRAM werden für die neue Sprache in jedem Falle übernommen werden können; es gibt allerdings auch einige Kritikpunkte, die im zweiten Teil dieses Kapitels kurz zusammengefaßt werden sollen.

2.1 Korpusanalyse

Ziel dieser Arbeit ist es, für zeitliche Ausdrücke in Terminvereinbarungsdialogen eine geeignete Art der Repräsentation zu finden, welche insbesondere die Definition einer formalen Semantik solcher Ausdrücke erlaubt. Durch die im Rahmen des VERBMOBIL-Projekts gesammelten Transliterationen gesprochener Dialoge wird die Menge der zu betrachtenden zeitlichen Ausdrücke eingegrenzt. Es liegen Transliterationen sowohl in deutscher als auch in englischer Sprache vor. Außerdem gibt es Dialoge zwischen Deutsch und Englisch sprechenden Verhandlungspartnern. Tafel 2.1 zeigt einen Auszug aus einem Dialog aus dem VERBMOBIL-Korpus. Die in diesem Beispieldialog vorkommenden zeitlichen Ausdrücke sind *kursiv* gedruckt. Die mehrfach vorkommende Anapher "da", die sich jeweils auf einen zeitlichen Ausdruck bezieht, wurde dabei allerdings nicht speziell hervorgehoben.

DIALOG G024AC (AUSZUG)	
AER001	Guten Tag. Wir müssen uns noch über unsere Geschäftsreise nach Hannover kurz unterhalten.
AES002	Ja.
AER003	Ich denke, wir sollten das schon *innerhalb der nächsten drei Monate* arrangieren und ich denke auch, *anderthalb Tage* dafür einzukalkulieren. Könnten Sie mir vielleicht 'nen Vorschlag machen, wann Sie könnten in der nächsten Zeit?
AES004	Ja, also ich muß Ihnen sagen, der *August* ist bei mir sehr voll, da wird es schwer sein, einen Termin zu finden. Mir wäre *die erste Juliwoche* sehr recht.
AER007	Da wird es schwierig, weil ich *im Juli am Mittwoch* einen Gesprächstermin hab' und *am Freitag* habe ich auch einen Abendtermin. Könnten wir vielleicht noch 'nen Ausweichtermin finden?
AES008	Wie sieht es denn mit *der dritten Juliwoche* aus, zum Beispiel *am fünfzehnten, sechzehnten Juli*?
AER009	Bin ich leider auch blockiert, da hab' ich eine Fahrt nach Berlin. Und zwar genau so, daß ich nur *einen Tag* frei hab' *am Mittwoch*, und *Donnerstag* bereits wieder 'n Gespräch *um fünfzehn Uhr*. 's wird also etwas schwierig. Und *eine Woche darauf*? Vielleicht irgendwas *zwischen dem einundzwanzigsten und dreiundzwanzigsten*?
AES011	Also, *am einundzwanzigsten* könnte ich nicht, da hab' ich *sowohl vormittags als auch nachmittags* noch Termine. Aber *am zweiundzwanzigsten und dreiundzwanzigsten, am Dienstag und Mittwoch* würd' es mir passen.
AER012	Ja, prima. Da hab' ich auch nichts vor, da könnten wir auf jeden Fall fahren. Dann halten wir den doch mal fest, *zweiundzwanzigsten und dreiundzwanzigsten Juli.*

Tafel 2.1: Beispieldialog aus dem VERBMOBIL-Korpus

2.1.1 Klassifikation Zeitlicher Ausdrücke

Als *zeitliche Ausdrücke* bezeichne ich diejenigen Teile eines Redebeitrags, die dessen propositionalen Gehalt in Bezug auf temporale Aspekte des gerade verhandelten Termins beinhalten. Da anzunehmen ist, daß ein noch zu verhandelnder Termin aus Sicht der Sprecher stets in der Zukunft liegt, werden insbesondere die Tempora von Verben *nicht* untersucht werden müssen. Zeitliche Ausdrücke im engsten Sinne sind etwa "Montag", "Juli", "fünfzehn Uhr", "zwei Wochen" oder "übermorgen". Natürlich werden darüber hinaus aber auch beispielsweise Präpositionen (etwa "von ... bis", "für", "nach"), sonstige Partikeln ("oder", "nicht") und andere Wörter mehr eine Rolle spielen. Bereits in [Küssner und Stede, 1995] bemerken die Autoren, daß "... das Verb des Satzes gegebenenfalls für die Desambiguierung ... heranzuziehen (ist) ...".

Zahlreiche Beispiele aus den transliterierten VERBMOBIL-Dialogen sind in Anhang A abgedruckt. Anhand dieser Daten soll im folgenden eine Klassifizierung zeitlicher Ausdrücke vorgenommen werden (wenngleich nicht zu jedem einzelnen Phänomen ein Beispiel aufgenommen wurde). An dieser Stelle werde ich mich mit einer Grobeinteilung begnügen. Diese wird dann später bei der Definition der Syntax einer geeigneten Repräsentationssprache weiter verfeinert werden.

2.1.2 Lage und Dauer

Grundsätzlich können zeitliche Ausdrücke in zwei Hauptgruppen unterteilt werden. Entweder spezifiziert ein Ausdruck die Lage eines Zeitpunktes (beziehungsweise einer Menge von Zeitpunkten) oder aber er macht eine Aussage über eine zeitliche Dauer. Wie aus den Daten unmittelbar ersichtlich ist, stellt die erste dieser Gruppen die weitaus größere und mannigfaltigere dar. Ein einfaches Beispiel für die Spezifikation der Lage eines Termins ist folgender Satz:[1]

> "Wie wär's denn mit Donnerstag den siebzehnten Dezember um acht Uhr dreißig?"

In folgendem Beispiel hingegen wird die Dauer eines Termins beschrieben:

> "Dann kommen wir jetzt zum dritten Termin für ein fünftägiges Arbeitstreffen in der Filiale in Karlsruhe."

Zeitliche Ausdrücke zur Spezifikation einer Dauer können darüber hinaus als Teilausdrücke von Ausdrücken, die die Lage eines Zeitpunktes beschreiben, vorkommen. Im nächsten Beispiel wird durch "ein, zwei Tage" eine Dauer angegeben. Der Gesamtausdruck verweist jedoch auf die Lage eines Zeitpunktes, und zwar eines Zeitpunktes, der ein oder zwei Tage vor dem Zeitpunkt liegt, der gerade Gegenstand des Geprächs gewesen ist.

> "Geht das vielleicht ein, zwei Tage früher?"

2.1.3 Spezifikation einer Dauer

Zunächst werde ich erörtern, welche Erscheinungsformen von Dauerangaben möglich sind. Im einfachsten Falle wird eine Dauer durch eine Zahl und eine zeitliche Einheit spezifiziert.

> "Ich glaube, anderthalb Stunden müßten reichen."

[1]Dieser und alle im folgenden zitierten Beispielsätze sind auch in Anhang A (dort mit Quellenangabe) abgedruckt.

Hier ist die Einheit eine Stunde, die Zahlenangabe ist $1\frac{1}{2}$. Als Zahlen sind in diesem Zusammenhang natürliche Zahlen sowie einfache Brüche und gemischte Zahlen denkbar. Die in Frage kommenden Einheiten sind Minuten, Stunden, Tage, Wochen, Monate und Jahre. Die Dauer eines Termins wird allerdings sicherlich nicht in Monaten oder Jahren angegeben werden; jedoch in Teilausdrücken wie etwa bei "zwei Monate vor Weihnachten" kann auch die Verwendung dieser Einheiten durchaus sinnvoll sein.

Um eine Dauer einzugrenzen, können auch Bereiche einfacher Dauern angegeben werden. Durch "zwei bis drei Tage" wird eine Dauer beschrieben, deren Länge irgendwo im Bereich zwischen zwei und drei Tagen liegt. Es müssen aber nicht unbedingt immer zwei Grenzen angegeben werden. Auch durch Ausdrücke wie "mindestens drei Stunden" oder "weniger als eine Woche" werden zeitliche Dauern eingegrenzt. Außerdem können explizit unscharfe Angeben gemacht werden, wie folgendes Beispiel zeigt:

> "We need to get another meeting going for about two hours in the next few weeks."

Des weiteren kann ein Sprecher in einem Satz mehrere Möglichkeiten zur Auswahl anbieten. In solch einem Falle handelt es sich um eine Disjunktion von Dauerangaben. Schließlich kann durch den Gebrauch einer Anapher implizit eine Dauer angegeben werden.

2.1.4 Spezifikation der Lage eines Termins

Die Klasse der zeitlichen Ausdrücke zur Spezifikation der Lage eines Zeitpunktes (beziehungsweise einer Menge von Zeitpunkten) ist weitaus komplexer als die der Dauerangaben. Zunächst gibt es eine umfangreiche Menge einfacher Ausdrücke. Hierzu gehören unter anderem Ausdrücke zur Beschreibung der Uhrzeit, der Tageszeit, des Wochentages, des Datums (das heißt des Monats und des Tages innerhalb eines Monats) und des Jahres. Darüber hinaus kann zum Beispiel auch auf Feiertage Bezug genommen werden. Außerdem kommen diverse einfache deiktische Bezeichner wie "jetzt" oder "morgen" vor.

Ein oder mehrere einfache Ausdrücke können zu komplexen Ausdrücken zusammengesetzt werden. Innerhalb der komplexen Ausdrücke zur Beschreibung der Lage von Zeitpunkten lassen sich sechs Gruppen unterscheiden. Die ersten beiden Gruppen sind die der Intervalle, und zwar der *geschlossenen Intervalle* sowie der *einseitig unbegrenzten Intervalle*. Im nächsten Beispiel etwa wird durch "one to four pm" ein Intervall mit zwei Randpunkten beschrieben:

> "However on the twenty eighth, which is Wednesday, I've got one to four pm free."

Einseitig unbegrenzte Intervalle sind Intervalle mit nur einem Randpunkt. Es wird dann entweder auf alle Zeitpunkte vor oder auf alle Zeitpunkte nach dem angesprochenen Zeitraum Bezug genommen. Ein Beispiel hierfür enthält der folgende Satz:

> "Okay, how 'bout Tuesday March the sixteenth sometime after twelve o'clock pm?"

Durch "after twelve o'clock pm" werden alle Zeitpunkte nach zwölf Uhr mittags beschrieben.

Bei zwei weiteren Gruppen komplexer Ausdrücke wird der bezeichnete Zeitraum durch Referenz auf einen anderen Ausdruck angegeben. Zum einen kann ein gegebener Zeitpunkt um einen gewissen Betrag entweder in die Vergangenheit oder in die Zukunft verschoben werden. Ein Beispiel für solch eine *Verschiebung* ist der Ausdruck "drei Tage nach Neujahr". Hier wird der durch den Referenzausdruck "Neujahr" bezeichnete Zeitraum um drei Tage in die Zukunft verschoben. Beachte, daß Ausdrücke wie "in einer Woche" eine Subklasse dieser Gruppe bilden; der hier implizit gegebene Referenzzeitpunkt ist "jetzt". Der Betrag, um den verschoben wird, wird durch einen Ausdruck zur Spezifikation einer Dauer angegeben.

Auch im folgenden Beispiel wird der bezeichnete Zeitraum mit Hilfe eines Referenzausdrucks spezifiziert:

> "Ja, ich muß sagen, die Woche vor Ostern wäre sehr knapp."

Hier wird jedoch nicht wie zuvor um einen angegebenen Betrag verschoben, sondern der bezeichnete Zeitraum ("die Woche vor Ostern") wird in Relation zu einem einfacheren Ausdruck ("Ostern") genannt. Gleichbedeutend könnte man auch von "der letzten Woche vor Ostern" sprechen. Es kann auch zusätzlich eine ordinale Zahl angegeben werden und so auf "die zweite Woche nach Ostern" oder "die dritte Woche vor Ostern" Bezug genommen werden. Ausdrücke wie "die Woche um Neujahr" rechne ich ebenfalls dieser Klasse der *Relationalen Angaben* zu.

Bei der fünften Klasse handelt es sich um die *Abzählung* kleiner Zeiträume innerhalb größerer Bezugszeiträume. So wird zum Beispiel durch "die dritte Woche im Mai" der kleinere Zeitraum "Woche" innerhalb des größeren "Mai" abgezählt. Solch eine Abzählung kann auch am Ende des größeren Intervalls beginnen:

> "Vielleicht sollten wir das gleich noch in der letzten Juliwoche machen."

Schließlich gibt es noch die Gruppe der *Deiktischen Angaben.* Durch Kombination eines einfachen zeitlichen Ausdrucks mit einem deiktischen Bezeicher entsteht ein neuer zeitlicher Ausdruck, zum Beispiel "nächste Woche". Eine Übersicht über die aufgeführten sechs Klassen komplexer Ausdrücke anhand einfacher Beispiele gibt Tafel 2.2.

Ebenso wie auch schon bei Ausdrücken zur Beschreibung einer Zeitdauer können auch Angaben zu Zeitpunkten explizit unscharf sein ("ungefähr am dritten"). Darüber hinaus gibt es die Möglichkeit, Ausdrücke zu modifizieren, wie folgendes Beispiel zeigt:

SPEZIFIKATION DER LAGE EINES TERMINS	
Intervalle mit zwei Randpunkten	– "one to four pm" – "zwischen heute und dem Wochenende"
Intervalle mit einem Randpunkt	– "after twelve o'clock pm" – "bis Mittwoch einschließlich"
Verschiebungen	– "drei Tage nach Neujahr" – "two weeks ago"
Relationale Angaben	– "die Woche vor Ostern" – "der zweite Montag nach Neujahr" – "das Wochenende um den siebten"
Abzählungen	– "die dritte Woche im Mai" – "the last day of the month"
Deiktische Angaben	– "die nächste Woche" – "the next three weeks"

Tafel 2.2: Klassifikation komplexer Ausdrücke

> "Da würd' ich doch lieber vorschlagen, daß wir uns auf Ende April einigen."

Der einfache Ausdruck "April" wird hier durch das Wort "Ende" modifiziert. Weitere Modifikatoren sind zum Beispiel "früh", "spät", oder "in der Mitte". Ausdrücke zur Eingrenzung der Lage eines Zeitraumes können auch quantifiziert werden. Beispielsweise könnte von "allen Montagen im Mai" die Rede sein. Statt Zeitangaben direkt zu machen können natürlich auch wieder Anaphern benutzt werden. Im nächsten Beispiel etwa wird durch "that" auf den zeitlichen Ausdruck "from two to four thirty on that afternoon" verwiesen:

> "I'm busy from two to four thirty on that afternoon. Could you do something after that?"

Alle diese Ausdrücke können auch negiert werden, es können mehrere Ausdrücke zur Auswahl angeboten werden (Disjunktion), und es können mehrere (unterspezifizierte) Ausdrücke zu einem spezifischeren Ausdruck zusammengefaßt werden (Konjunktion). Ein Beispiel für die Disjunktion ist folgender Satz:

> "Ich habe nur Zeit vom sechsten Mai bis elften Mai, vom achtundzwanzigsten Mai bis fünften Juni und vom dreizehnten Juni bis zwanzigsten Juni."

Die meisten Redebeiträge enthalten eine Konjunktion von Einzelausdrücken, so auch der folgende:

> "How 'bout the afternoon of Monday the ninth?"

Hier werden die unterspezifizierten Ausdrücke "afternoon", "Monday" und "the ninth" zu einem spezielleren Ausdruck zusammengefaßt.

2.2 Kritik an ZEITGRAM

In [Küssner und Stede, 1995] wird die Sprache ZEITGRAM zur Repräsentation zeitlicher Ausdrücke in Terminvereinbarungsdialogen vorgestellt. Die Grammatik dieser Sprache ist in Anhang B.1 dieser Arbeit nochmals abgedruckt. Auf eine systematische Einführung soll an dieser Stelle verzichtet werden. Wie mit Hilfe von ZEITGRAM temporale Ausdrücke repräsentiert werden können, läßt sich leicht anhand einiger Beispiele demonstrieren.

- "Dann würd' ich sagen um zehn Uhr fünfzehn."
  `[tod:10:15]`
- "Tut mir leid, da hab' ich schon einen Termin in Lübeck, von Montag bis Dienstag."
  `[boundaries([dow:mon],[dow:tue],from_to)]`
- "Freitag ist prinzipiell gar nicht schlecht und zwar nachmittags um dreiviertel drei."
  `[dow:fri,pod:afternoon,tod:02:45]`
- "Also, ich dachte noch in der nächsten Woche, auf jeden Fall noch im April."
  `[counted(next,week)], [moy:apr]`
- "Treffen wir uns um die Mittagszeit?"
  `[about([pod:noon])]`

Im folgenden werde ich einige Unzulänglichkeiten von ZEITGRAM aufzeigen.

2.2.1 Abdeckungslücken

Es wurden einige Lücken in der Abdeckung von ZEITGRAM gefunden, das heißt es gibt einige natürlichsprachliche zeitliche Ausdrücke, die nicht repräsentiert werden können. Eine Auswahl der gefundenen Probleme bietet folgende Liste.

- Die im ersten Teil dieses Kapitels identifizierte Gruppe von Ausdrücken, die eine *Verschiebung* eines Zeitpunktes (oder einer Menge von Zeitpunkten) darstellen, kann durch ZEITGRAM-Ausdrücke nicht repräsentiert werden. Zu dieser Gruppe gehört zum Beispiel der Ausdruck "drei Wochen nach Neujahr", durch den auf den Tag verwiesen wird, der gerade drei Wochen nach Neujahr liegt. Beachte, daß der Ausdruck `[after(3,week,holiday:neujahr)]` eine andere Bedeutung hat. Er entspricht der natürlichsprachlichen Äußerung "die dritte Woche nach Neujahr", wodurch offfensichtlich nicht ein bestimmter Tag, sondern eine ganze Woche beschrieben wird.
- ZEITGRAM erlaubt nur für eine bestimmte Gruppe von Ausdrücken eine Modifikation (zum Beispiel "früh am Morgen"). Ein Ausdruck wie "in der ersten Hälfte der Woche um den dreizehnten" ist nicht darstellbar.

- Durch den Ausdruck `[around(week,dom:20)]` wird "die Woche um den zwanzigsten" beschrieben. Die Grammatik erlaubt hier als erstes Argument ausschließlich den Wert `week`. Gleiches gilt für die Funktoren `before` und `after`. Das heißt, Ausdrücke wie "die drei Tage nach dem zwanzigsten", "das Wochenende um den zwanzigsten" oder "der Feiertag vor dem zwanzigsten" können nicht repräsentiert werden.

- Die Äußerung "die erste Woche im März" wird durch den ZEITGRAM-Ausdruck `[counted(1,week,moy:mar)]` repräsentiert. Komplexere Ausdrücke können an der zweiten Argumentstelle jedoch nicht angegeben werden, so daß beispielsweise "die ersten zwei Wochen im März" nicht dargestellt werden können. Das gilt auch für Ausdrücke wie "die nächsten zwei Wochen".

- Es reicht nicht aus, bei Dauerangaben nur natürliche Zahlen zu berücksichtigen. Darüberhinaus sollten auch Äußerungen wie "zweieinhalb Wochen" oder "eine dreiviertel Stunde" darstellbar sein.

- Einseitig unbegrenzte Bereiche von Dauern (zum Beispiel "mindestens drei Stunden") können nicht angegeben werden. Für Ausdrücke zur Beschreibung der Lage eines Termins werden Intervalle mit nur einem Randpunkt zwar durch die Kategorie `LIMIT` repräsentiert, es wäre jedoch wenig sinnvoll, etwa `[in_after(dur(3,hour))]` für obiges Beispiel zu verwenden. Im allgemeinen scheint es ratsam, Dauerangaben künftig separat zu behandeln.

2.2.2 Deiktische Angaben

In ZEITGRAM werden in der Kategorie `DEIC_COUNTER` Ausdrücke wie `this`, `last` und `next` zusammengefaßt. Beispielsweise wird "der letzte Samstag" als `[last(sat)]` repräsentiert. Hierbei wird "letzte" tatsächlich deiktisch gebraucht; die Bedeutung des Ausdrucks erschließt sich nur in Zusammenhang mit dem Zeitpunkt der Äußerung. Allerdings ist die Verwendung dieser Ausdrücke nicht immer deiktisch. Die Bedeutung der Äußerung "der letzte Samstag im Mai 1998" ist unabhängig vom Sprechzeitpunkt, das Wort "letzte" wird also nicht deiktisch verwendet. Dies gilt allgemein für Ausdrücke der Form `counted(COUNTER,<week|DOW>,COUNT_UNIT_OR_MONTH)`. Die Verwendung des Bezeichners `this` wäre in diesem Zusammenhang auch aus linguistischer Sicht unsinnig.

Bei einem Ausdruck wie "die vorletzte Woche vor Ostern" handelt es sich nur um eine sprachliche Variante von "die zweite Woche vor Ostern". Auch für Ausdrücke der Form `<before|after>(COUNTER,<UNIT|DOW>,POINTLIKE)` sollte die Verwendung von `DEIC_COUNTER`-Ausdrücken also ausgeschlossen werden.

Darüber hinaus ist sicherlich fraglich, ob es sinnvoll ist, beispielsweise "den nächsten Januar" als `[next(jan)]` darzustellen, "die nächste Woche" hingegen als `[counted(next,week)]`.

2.2.3 Beschreibung eines Termins

Die bisher genannten Probleme ließen sich durch begrenzte Änderungen beziehungsweise Ergänzungen der Grammatik relativ einfach beheben. Es gibt jedoch auch ein grundlegenderes, ein strukturelles Problem. Es besteht darin, daß ZEITGRAM zwar eine Sprache zur Repräsentation zeitlicher Ausdrücke, nicht jedoch eine Sprache zur Beschreibung eines Termins ist. Ein Termin hat im wesentlichen drei Merkmale: einen Anfang, ein Ende sowie eine Dauer. Die Dauer kann in ZEITGRAM durch Ausdrücke der Kategorie `DURATION` beschrieben werden. Bei Anfang und Ende eines Termins handelt es sich um Zeitpunkte. Die Zeitpunkte selbst können selbstverständlich auch durch ZEITGRAM in vielfältiger Weise umschrieben werden. Es fehlt lediglich die Angabe, ob es sich im jeweiligen Falle um die Beschreibung des Anfangs, des Endes oder vielleicht beider zugleich handelt. Solch eine Zuordnung von ZEITGRAM-Ausdrücken zu spezifischen Merkmalen eines Termins gibt es im engeren Sinne nur bei zwei Gruppen von Ausdrücken. Die erste Gruppe ist die der `DURATION`-Ausdrücke; hier ist offensichtlich, daß es sich nur um die Beschreibung einer Termindauer handeln kann. Die zweite Gruppe sind Ausdrücke der Kategorie `BOUNDARIES`, bei denen das letzte Argument `from_to` ist. Hier ist klar, daß sich der an erster Stelle stehende zeitliche Ausdruck auf den Beginn eines Termins und der zweite auf dessen Ende bezieht.

Bei allen anderen Ausdrücken wird keine solche Zuordnung impliziert, sie muß also explizit formuliert werden. Jeder der verbleibenden Ausdrücke bezieht sich auf die Lage eines Termins, und zwar entweder auf dessen Anfang oder auf dessen Ende oder aber auf den gesamten Termin. Welche dieser drei Möglichkeiten zutrifft, muß allgemein festgelegt werden, das heißt es ist nicht zulässig, daß sich ein Ausdruck mal auf dieses, mal auf jenes Merkmal bezieht. Daß sich all jene ZEITGRAM-Ausdrücke auf das Ende eines Termins beziehen, kann sicherlich ausgeschlossen werden. Das Ende eines Termins kann also ausschließlich über die Verwendung eines `BOUNDARIES`-Ausdrucks (bei gleichzeitiger Spezifikation des Anfangspunktes) beschrieben werden.

Bleiben die Möglichkeiten, daß sich die Ausdrücke entweder auf den Terminbeginn oder auf den gesamten Termin beziehen. Die zweite Möglichkeit muß aus verschiedenen Gründen ausgeschlossen werden. Für manche Fälle (wie die Beschreibung von Uhrzeiten) kommt diese Möglichkeit ohnehin von vornherein nicht in Betracht. Aber auch ein Ausdruck wie "am Montag" kann sich nicht auf die Gesamtheit eines Termins beziehen, wenn dieser unter Umständen eine Dauer von mehreren Tagen haben kann. So erschließt sich, daß sich ZEITGRAM-Ausdrücke (von den zwei erwähnten Ausnahmen abgesehen) stets auf den Beginn eines Termins beziehen müssen. Bestimmte Varianten der Beschreibung eines Termins können durch ZEITGRAM also nicht geleistet werden, was die eingangs dieses Abschnitts gemachte Aussage, ZEITGRAM sei keine Sprache zur Repräsentation eines Termins, rechtfertigt.

Damit die hier zu entwickelnde Sprache auch zur Darstellung des temporalen Gehalts des jeweils verhandelten Termins für das Dialoggedächtnis von VERBMOBIL benutzt werden kann, sollte ZEITGRAM sicherlich entsprechend erweitert beziehungsweise so umstrukturiert werden, daß alle Merkmale eines Termins repräsentiert werden können.

2.3 Zusammenfassung

Im ersten Teil dieses Kapitels wurde eine Auswahl von Redebeiträgen aus dem umfangreichen VERBMOBIL-Korpus analysiert. Diese Analyse führte zu einer ersten groben Klassifikation zeitlicher Ausdrücke bei Terminvereinbarungsdialogen. Anhand dieser Klassifikation konnten im zweiten Teil einige Abdeckungslücken der Sprache ZEITGRAM identifiziert werden. Als Hauptkritikpunkt an dieser Sprache bleibt festzuhalten, daß in ihr die unterschiedlichen Merkmale eines Termins (wie Terminbeginn oder Termindauer) nicht explizit beschrieben werden können. Um aber ein Dialoggedächtnis aufbauen zu können, ist dieses eine notwendige Bedingung, die an eine Sprache zur Repräsentation des temporalen Gehalts von Äußerungen bei Terminabsprachen gestellt werden muß.

Aufbauend auf die durch die Analyse gewonnenen Kenntnisse wird in Kapitel 4 eine neue formale Sprache für zeitliche Ausdrücke entwickelt werden. Im nächsten Kapitel sollen zunächst aber einige für das Thema dieser Arbeit möglicherweise relevanten Ansätze aus der Literatur erörtert werden.

Kapitel 3

Literaturrecherche

... und doch ist das Zitieren alter und neuer Bücher das
Hauptvergnügen eines jungen Autors, und so ein paar
grundgelehrte Zitate zieren den ganzen Menschen.
Heinrich Heine: Ideen. Das Buch Le Grand.

Repräsentation von und Inferenzen über zeitlichen Abläufen stehen im Mittelpunkt von temporalen Logiken sowie von Planungsverfahren in der der Künstlichen Intelligenz. Auch gab es verschiedene Ansätze, temporales Schließen in Beschreibungslogiken zu integrieren. Über solche Bestrebungen soll in diesem Kapitel ein Überblick gegeben werden.

Im ersten Teil werden temporale Logiken, genauer temporale Logiken als spezielle Form modaler Aussagenlogik, betrachtet werden. Außerdem werden executable temporal logics sowie eine sogenannte chronologische Temporallogik kurz angesprochen. Sofern es im Zusammenhang mit Planungsverfahren um die Repräsentation von Zeit geht, gilt die Arbeit von James Allen (siehe unter anderem [Allen *et al.*, 1991]) als einer der wichtigsten Beiträge. Sie wird im zweiten Teil behandelt werden. Dem folgt die Darstellung einiger Ansätze, temporales Wissen in ein beschreibungslogisches System zu integrieren. Im vierten Teil des Kapitels schließlich werde ich eine Arbeit zur Modellierung zeitlicher Ausdrücke für ein Zugauskunftssystem vorstellen.

In jedem der vier Teile werden die vorgestellten Ansätze danach beurteilt werden, inwieweit sie zur Lösung des Problems der angemessenen formalen Repräsentation des temporalen Gehalts von natürlichsprachlichen Äußerungen bei Terminvereinbarungsdialogen beitragen können. In diesem Sinne ist auch die Zusammenfassung, mit der dieses Kapitel schließt, zu verstehen.

3.1 Temporale Logiken

3.1.1 Temporale Modallogik

In der Literatur finden sich viele verschiedene Ansichten, was genau unter einer temporalen Logik zu verstehen sei. Dieser Abschnitt beschreibt eine Temporallogik als Instanz modaler Aussagenlogik, wie dies etwa auch in [Goldblatt, 1992] oder [Schmitt, 1997] der Fall ist. Zunächst folgt also eine kurze Einführung in modale Logik.

3.1.1.1 Modale Aussagenlogik

Klassische Aussagenlogik erlaubt Aussagen über die Wahr- bzw. die Falschheit von Formeln A, also “A ist wahr” oder “A ist falsch”. Um darüber hinaus Aussagen wie zum Beispiel “A ist notwendigerweise wahr”, “A ist irgendwann wahr” oder auch “Marie glaubt A” machen zu können, kann die Aussagenlogik um *modale Operatoren* erweitert werden; das Ergebnis ist dann eine Modallogik (siehe [Fitting, 1993]).

Im einfachsten Falle gibt es zwei dieser modalen Operatoren: $\Box$ und $\Diamond$. Eine (von vielen möglichen) Lesarten interpretiert dabei $\Box A$ als “notwendigerweise A” und $\Diamond A$ als “möglicherweise A”. Beide Operatoren sind nicht unabhängig voneinander, insbesondere kann einer über den anderen definiert werden:

$$\Diamond A \;:\Longleftrightarrow\; \neg\Box\neg A$$

(Also: A ist genau dann möglicherweise wahr, wenn die Negation von A nicht notwendigerweise gilt.)

Die Formulierung “A ist notwendigerweise wahr” kann auch durch “A ist wahr in allen möglichen Welten” ersetzt werden.

Definition 3.1 *Ein* Kripke Rahmen *ist ein Paar $\mathcal{F} = (S, R)$ aus einer nichtleeren Menge S und einer binären Relation R auf S. Hierbei heißt S die Menge der* möglichen Welten *und R die* Zugänglichkeitsrelation*.*

Durch die Relation R wird also festgelegt, welche Welten von einer bestimmten Welt aus ‘zugänglich’ sind. Die Formel $\Box A$ ist wahr in einer Welt $w \in S$ genau dann, wenn in *jeder* Welt $w' \in S$ mit $R(w, w')$ die Formel A wahr ist, das heißt, wenn A wahr ist in all jenen Welten, die von w aus zugänglich sind. Entsprechend ist $\Diamond A$ wahr in w genau dann, wenn es *eine* Welt $w' \in S$ gibt mit $R(w, w')$, in der A wahr ist.

Werden mehrere Paare von Modaloperatoren eingeführt, so spricht man auch von multi-modaler Logik.

3.1.1.2 Lineare Temporallogik

Interpretiert man die möglichen Welten eines Kripke Rahmens als verschiedene Zeitzustände, so ergibt sich eine einfache Temporallogik. Die Zugänglichkeitsrelation $R(t, t')$ muß dann als “t liegt zeitlich vor t'” gelesen werden (und kann $<$ statt R geschrieben werden). In der Literatur finden sich unterschiedliche Zeitmodelle; für das VERBMOBIL-Szenario bietet sich eine diskrete, lineare Struktur an.

$\Box A$ wird in solch einer temporalen Logik "immer A", $\Diamond A$ als "irgendwann A" gelesen. Da sich auf diese Weise aber keine Aussagen über die Vergangenheit machen lassen, werden meist zwei Paare von modalen Operatoren definiert, für die Zukunft und für die Vergangenheit. $[F]$ steht für "immer in der Zukunft", $\langle F\rangle$ für "irgendwann in der Zukunft", analog $[P]$ für "immer in der Vergangenheit" und $\langle P\rangle$ für "irgendwann in der Vergangenheit". In einem temporalen Rahmen $\mathcal{F} = (S, <)$ mit einer Menge von Zeitpunkten S und einer darauf erklärten zeitlichen Relation $<$ gilt also zur Zeit t:

- $[F]A$ genau dann, wenn für alle $t' \in S$ mit $t < t'$ A zur Zeit t' wahr ist,
- $\langle F\rangle A$ genau dann, wenn es ein $t' \in S$ mit $t < t'$ gibt, so daß A zur Zeit t' wahr ist,
- $[P]A$ genau dann, wenn für alle $t' \in S$ mit $t' < t$ A zur Zeit t' wahr ist und
- $\langle P\rangle A$ genau dann, wenn es ein $t' \in S$ mit $t' < t$ gibt, so daß A zur Zeit t' wahr ist.

Zwei weitere wichtige temporale Operatoren sind since und until. Hierbei bedeutet $A \,\mathsf{since}\, B$, daß B irgendwann einmal wahr gewesen ist und von dieser Zeit bis jetzt A wahr ist. Analog gilt $A \,\mathsf{until}\, B$, wenn irgendwann in der Zukunft B wahr sein wird und von jetzt bis dorthin A gilt. In $\mathcal{F}$ gilt zur Zeit t:

- $A \,\mathsf{since}\, B$ genau dann, wenn es ein $t' \in S$ mit $t' < t$ gibt, so daß B zur Zeit t' gilt und zu allen Zeiten $t'' \in S$ mit $t' < t'' < t$ A wahr ist sowie
- $A \,\mathsf{until}\, B$ genau dann, wenn es ein $t' \in S$ mit $t < t'$ gibt, so daß B zur Zeit t' gilt und zu allen Zeiten $t'' \in S$ mit $t < t'' < t'$ A wahr ist.

Mit Hilfe der Operatoren since und until können leicht weitere Operatoren definiert werden. Beispielsweise wird durch $\perp \mathsf{until}\, A$ ausgedrückt, daß A zu einem zukünftigen Zeitpunkt wahr sein wird, wobei es zwischen diesem Zeitpunkt und dem aktuellen keine weiteren Zeitpunkte gibt. Das heißt, A ist wahr zum unmittelbar 'nächsten' Zeitpunkt.

Es ist offensichtlich, daß Temporallogik sicherlich nichts zur formalen Repräsentation natürlichsprachlicher Ausdrücke wird beitragen können. Eine Anwendung von Deduktionsverfahren für temporale Logiken könnte unter Umständen bei der Berechnung von Inferenzen über zeitlichen Ausdrücken hilfreich sein, eine Übersetzung aus einer formalen Sprache in geeignete Formeln scheint jedoch schwierig.

3.1.2 Executable Temporal Logics

In PROLOG (siehe [Bratko, 1990]) werden Horn-Formeln in klassischer Logik direkt ausgeführt. Entsprechende Ansätze existieren auch für temporale Logiken. Über dieses Feld der sogenannten *executable temporal logics* gibt [Fisher und Owens, 1993] einen Überblick. Da sich aber schon die temporale Logik selbst als nicht geeignet zur Behandlung der im Rahmen dieser Arbeit zu lösenden Probleme erwiesen hat, kann auf eine Einführung in die executable temporal logics hier verzichtet werden.

3.1.3 Eine Chronologische Temporallogik

In [Åqvist, 1978] wird über dem Kripke Rahmen einer linearen Temporallogik ein Maß für Zeitdauern definiert. Solch ein Rahmen wird dort als chronologischer Rahmen bezeichnet. Dieses Vorgehen erlaubt die Kombination einfacher Zeitangaben und temporallogischer Operatoren. Zur semantischen Interpretation zeitlicher Ausdrücke würden jedoch höchstens die mit Hilfe des Maßes für Dauern definierten Funktionen etwas beitragen können. Die temporale Logik selbst könnte aber auch hier wohl kaum nutzbringend eingebracht werden.

3.2 Allens Intervall-Kalkül

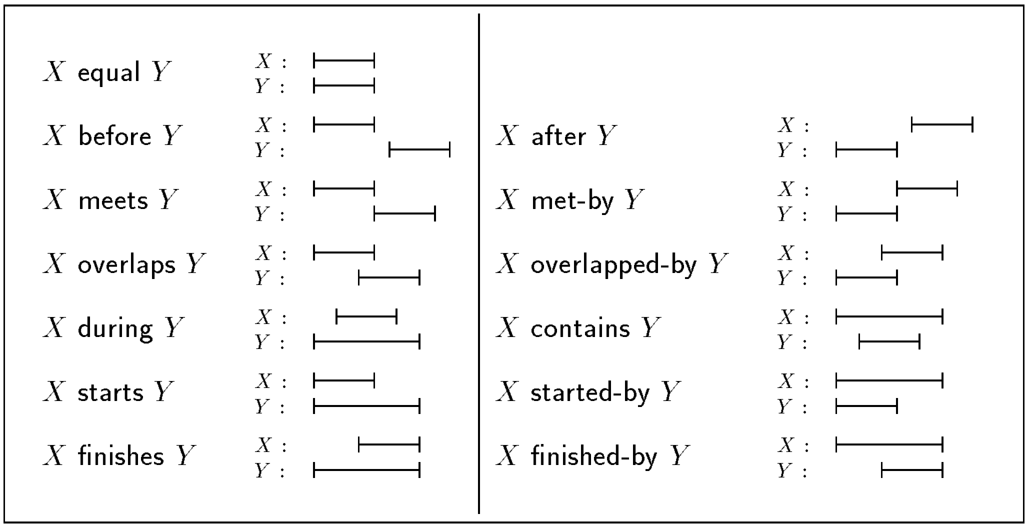
3.2.1 Intervalle

Einen der wohl bekanntesten Ansätze zur Repräsentation temporalen Wissens stellt [Allen, 1983] dar (siehe auch [Allen, 1984] und [Allen *et al.*, 1991]). Gegenstand der Betrachtung sind dort Ereignissen zugeordnete Zeiten. So werden durch den Satz "Sie öffnete den Brief, nachdem sie nach Hause gekommen war." zwei Ereignisse und eine temporale Relation zwischen ihnen eingeführt. Da solche Ereignisse stets in mehrere 'Subereignisse' unterteilt werden können, können auch die assoziierten Zeitintervalle immer weiter in kleinere Intervalle unterteilt werden. Daher scheint eine Repräsentation in Form von Zeitpunkten wenig sinnvoll. Allen wählt daher Intervalle selbst als Primitive für die Darstellung temporaler Information.

Zwei Intervalle können zueinander in Beziehung gesetzt werden durch Relationen wie equal, before oder during. Insgesamt gibt es sieben solche Relationen. Zu jeder dieser Relationen läßt sich außerdem die inverse Relation, die für equal allerdings wieder equal ist, angeben. Für ein geordnetes Paar von Intervallen existiern also 13 verschiedene mögliche Relationen. Sie sind in Tafel 3.1 aufgeführt. Zu jeder dieser Relationen kann ein Prädikat definiert werden, welches diese Relation ausdrückt (siehe hierzu insbesondere [Allen, 1984]), also zum Beispiel:

$$Before(X, Y) \;:\Longleftrightarrow\; X \;\mathsf{before}\; Y$$

Für diese Prädikate gelten eine Reihe von Axiomen; zusammen bilden sie das Intervall-Kalkül.



Tafel 3.1: Relationen zwischen zwei Intervallen

- Zu jedem Intervall X und für jedes eine der 13 Basisrelationen ausdrückende Prädikat P existiert ein Intervall Y mit $P(X, Y)$.
- Alle 13 Relationen (und damit auch alle 13 Prädikate) sind paarweise verschieden.
- Das Transitivitätsverhalten wird durch eine große Zahl von Axiomen beschrieben (siehe [Allen, 1983]). Beispiele:
  - $\mathit{Overlaps}(X_1, X_2) \wedge \mathit{Equal}(X_2, X_3) \Longrightarrow \mathit{Overlaps}(X_1, X_3)$
  - $\mathit{Before}(X_1, X_2) \wedge \mathit{Before}(X_2, X_3) \Longrightarrow \mathit{Before}(X_1, X_3)$
  - $\mathit{Meets}(X_1, X_2) \wedge \mathit{Finishes}(X_2, X_3) \Longrightarrow$
    $\mathit{During}(X_1, X_3) \vee \mathit{Starts}(X_1, X_3) \vee \mathit{Overlaps}(X_1, X_3)$

3.2.2 Erweiterungen

Ansonsten behandelt [Allen, 1983] im wesentlichen eine Inferenzprozedur zur effizienten Propagierung der Transitivitätsaxiome durch Netze von Intervallbeziehungen. Im übrigen werden einige Erweiterungen diskutiert, unter anderem die Integration einer Komponente zum Vergleichen der Längen von Intervallen (*Dauer*). Eine andere Erweiterung besteht in der Möglichkeit der Angabe absoluter Zeitangaben für Intervalle (*Datierung*).

Dauer Ähnlich den die zeitliche Abfolge von Intervallen betreffenden Relationen können auch Relationen über die Länge von Intervallen (oder die Dauer von Ereignissen) definiert werden. Um beispielsweise auszudrücken, daß Intervall X kürzer als Intervall Y ist, wird $dur(X) < dur(Y)$ geschrieben. Allen erlaubt auch numerische Faktoren, also etwa $dur(X) \leq 2 \cdot dur(Y)$, das heißt Intervall X ist höchstens doppelt so lang wie Y. Aussagen über die zeitliche Abfolge und die Dauer von Ereignissen bedingen einander (unter Umständen). Es gilt zum Beispiel:

$$During(X, Y) \vee Starts(X, Y) \vee Finishes(X, Y) \implies dur(X) < dur(Y)$$

Datierung Das Intervall-Kalkül erlaubt bisher nicht die Integration absoluter Daten. Hierfür muß zunächst ein Bereich für in Frage kommende Daten definiert werden, also etwa ein Kalender. Die auf diesem Bereich definierten Relationen können dann benutzt werden, um die auf den Intervallen definierten Relationen zu berechnen. Datierung dient also lediglich zur Vereinfachung der Berechnug von Relationen zwischen den Intervallen. Die Semantik eines komplexen Kalenders wird in diesem Zusammenhang von Allen jedoch nicht behandelt.

3.2.3 Diskussion

Wesentliches Argument für die Verwendung von Intervallen zur Repräsentation temporaler Information war, daß sich jede noch so kleine Zeitspanne immer wieder in kleinere Zeitspannen unterteilen läßt. Das ist zwar physikalisch korrekt, im Rahmen der Verarbeitung natürlichsprachlicher Äußerungen über Zeit aber eher unbedeutend. Vielmehr scheint für das VERBMOBIL-Szenario eine Granularität von einer Minute sinnvoll.

Die zentrale Rolle *relativer* zeitlicher Abfolge von Intervallen scheint unangemessen für die Domäne Terminabsprache, da hierbei der verhandelte Termin meist in Relation zu *absoluten* (oft allerdings unterspezifizierten absoluten) Daten gesetzt wird. Die Anbindung eines Bereiches von absoluten Zeitangaben wurde zwar angedeutet, die Gestaltung eines solchen Bereiches selbst jedoch nicht näher beleuchtet. Jedenfalls wird von vollständig spezifizierten, quasi-numerischen Daten ausgegangen, das heißt die mit der Repräsentation temporaler Ausdrücke aus natürlichsprachlichen Dialogen einhergehenden Probleme werden nicht behandelt. Allen selbst schreibt in [Allen, 1983], daß sein Ansatz sich vor allem für Bereiche eignet, in denen temporale Information unscharf und relativ ist und in denen Datierung (also etwa die Abbildung auf einen Kalender) nicht möglich ist.

3.3 Beschreibungslogik

Die semantische Kontextauswertung (siehe [Quantz *et al.*, 1997]) in VERBMOBIL arbeitet mit dem beschreibungslogischen System FLEX [Quantz *et al.*, 1995]. Hierbei ist es nicht gelungen, die Verarbeitung von Terminbeschreibungen in dieses System zu integrieren, was letztlich zur Entwicklung der Repräsentationssprache ZEITGRAM [Küssner und Stede, 1995] geführt hat. Im folgenden sollen nach einer kurzen Einführung in das Gebiet zwei im Hinblick auf das Problem der Verarbeitung zeitlicher Ausdrücke möglicherweise interessante Erweiterungen von beschreibungslogischen Systemen behandelt werden.

3.3.1 Allgemeines

Eine mittlerweile weitverbreitete Methode zur Repräsentation von Wissen stellen die sogenannten *KL-ONE-ähnlichen Sprachen* dar. Der Name verweist auf KL-ONE [Brachman und Schmolze, 1985], den ersten und prominentesten Vertreter solcher Systeme. Seit einiger Zeit sind auch die Bezeichnungen *Beschreibungslogiken* (im Englischen: *description logics*, daher auch *DL-Systeme*) oder *terminologische Logiken* gebräuchlich. Auf eine systematische Einführung in dieses Gebiet kann an dieser Stelle verzichtet werden (sie wird etwa in [Nebel, 1990] gegeben). Vielmehr werde ich mich auf eine 'intuitive' Darstellung beschränken.

Ein KL-ONE-ähnliches System besteht für gewöhnlich aus zwei Teilen. Die *TBox* dient dazu, terminologisches Wissen auszudrücken. Hierzu zählen Konzepte (Klassen von Individuen) und Rollen; letztere dienen zur Beschreibung einer Beziehung zwischen zwei Individuen. Eine Sonderolle kommt den *primitiven* Konzepten und Rollen zu; das sind solche, die nicht weiter (durch andere Konzepte oder Rollen) erklärt werden können. Alle übrigen werden durch Operatoren (wie Konjunktion oder Negation) aus anderen Konzepten bzw. Rollen abgeleitet. In der *ABox* wird das Objektwissen, Wissen über die einzelnen Individuen, behandelt (im Englischen: *assertional knowledge*). In der ABox werden also Individuen bestimmten Konzepten zugeordnet und es werden Rollenbeziehungen zwischen Individuen festgelegt.

Um beispielsweise auszudrücken, daß eine Frau ein menschliches Wesen ist und daß eine Mutter eine Frau mit wenigstens einem Kind ist, könnte eine TBox folgende Konzepte einführen:

$$\begin{array}{lcl} frau & :< & mensch \\ mutter & := & frau \ \mathbf{and\ atleast}(1, hat\_kind) \end{array}$$

In einer dazugehörigen ABox könnte etwa 'das Individuum' Marie eingeführt werden, von der beispielsweise bekannt sein könnte, daß sie Mutter ist:

$$marie \ :: \ mutter$$

Nachdem eine gewisse Wissensbasis in einer Beschreibungslogik explizit beschrieben worden ist, bieten die Systeme verschiedene Mechanismen an, um daraus logisch folgendes, implizites Wissen zu gewinnen. Hierzu gehört vor allem die Subsumption, also die Antwort auf die Frage, ob ein Konzept von einem anderen impliziert wird. Im obigen Beispiel sollte also etwa aus den expliziten Angaben darauf geschlossen werden können, daß Marie ein Mensch ist. Im folgenden werde ich aber zunächst nur auf die Möglichkeiten der Repräsentation mit Beschreibungslogiken eingehen. Nur wenn sich diese als adäquat für die Darstellung zeitlicher Ausdrücke, wie sie im VERBMOBIL-Korpus zu finden sind, erweisen sollten, wäre auch eine Betrachtung der Funktion der Inferenzmechanismen erforderlich.

3.3.2 Direkte Integration einer Temporalen Komponente

In diesem Abschnitt soll der in [Fischer, 1992] und [Neuwirth, 1993] vorgestellte Ansatz, temporales Wissen unmittelbar in ein beschreibunglogisches System zu integrieren, beschrieben werden. In besagten Arbeiten wird temporales Wissen in Mengen von Zeitpunkten dargestellt. Jedem Zeitpunkt entspricht eine ganze Zahl; 0 steht dabei für den aktuellen Moment. Die Schrittweite muß vor Aufbau der Wissensbasis festgelegt werden, beispielsweise auf ein Jahr. Nun kann einem Konzept ein bestimmter Gültigkeitszeitraum in Form einer Menge von (Zeitpunkte repräsentierenden) Zahlen zugewiesen werden. Alternativ kann der Gültigkeitszeitraum auch über **since** oder **until** in Bezug auf den Gültigkeitszeitraum eines anderen Terms angegeben werden.

Technisch wird diese Zuweisung als eine Erweiterung der Syntax von TBox und ABox realisiert. Die Syntax der TBox wird im wesentlichen um die Operatoren **alltime**, **sometime**, **since** und **until** ergänzt. Die Bezeichnungen entstammen der Temporallogik. So wird etwa durch

$$erwachsener \quad := \quad mensch \ \mathbf{and\ alltime}([-18, 0], lebendig)$$

ausgedrückt, daß ein Erwachsener ein Mensch ist, der von vor 18 Jahren bis jetzt zu *jedem* Zeitpunkt am Leben war bzw. ist.[1] Im Gegensatz dazu würde der Operator **sometime** verwendet werden, wenn lediglich ausgesagt werden sollte, daß etwas zu *wenigstens einem* Zeitpunkt aus einer gegebenen Menge von Zeitpunkten gelte.

Die Verwendung von **since** soll an folgendem Beispiel illustriert werden:

$$geschieden \quad := \quad unverheiratet \ \mathbf{since} \ scheidung$$

Jemand ist geschieden, wenn er seit seiner Scheidung unverheiratet geblieben ist. Der **until**-Operator wird entsprechend benutzt.

Für ein bestimmtes Individuum können auch nur bestimmte temporale Aussagen gemacht werden. Daher sind die Zeitangaben in der ABox nicht relativ (also 0 für "jetzt", etc.), sondern es werden absolute Zeitpunkte oder Mengen von absoluten Zeitpunkten angegeben. Die Syntax der ABox wurde um die Operatoren **at_all** und **at_some** erweitert. Im folgenden Beispiel wird ausgedrückt, daß Marie irgendwann zwischen 1990 und 1996 Studentin war und daß sie während 1996 und 1997 mit Paule verheiratet war:

$$\begin{array}{lll} marie & :: & studentin \ \mathbf{at\_some} \ [1990, 1996] \\ (marie, paule) & :: & verheiratet \ \mathbf{at\_all} \ [1996, 1997] \end{array}$$

Charakteristisch für den hier beschriebenen Ansatz ist, daß der zugrundeliegende Begriff von Zeit ein sehr einfacher ist: Mengen von Zeitpunkten, die wiederum ausschließlich als ganze Zahlen repräsentiert werden können. Es wird davon ausgegangen, daß einem jeden Ereignis unmittelbar ein Zeitpunkt (bzw. gegebenenfalls eine Menge von Zeitpunkten) auf der linearen Zeitachse zugeordnet

[1] $[-18, 0]$ steht hier abkürzend für die Menge der ganzen Zahlen von -18 bis 0.

werden kann. Ein wesentlicher Aspekt der im Kontext des VERBMOBIL-Projekts zu betrachtenden zeitlichen Ausdrücke ist jedoch die Vielzahl der unterschiedlichen syntaktischen Erscheinungsformen und die damit einhergehende Problematik der *nicht*-trivialen Abbildung auf eine lineare Zeitachse. Die sprachnahe Repräsentation zum einen und die Abbildung auf eine eindeutige Semantik (also gerade auf eine solche lineare Zeitachse) zum anderen kann also sicherlich nicht im Rahmen eines solchen temporalen beschreibungslogischen Systems geleistet werden. Auf der anderen Seite scheint die enorme Ausdruckskraft der rein beschreibungslogischen Komponente wiederum stark übertrieben für die gestellte Aufgabe. Die Stärke terminologischer Systeme liegt darin, sehr viele verschiedene Sachverhalte repräsentieren zu können. Bei der Auswertung temporaler Ausdrücke bei Terminvereinbarungen gibt es aber lediglich *einen* solchen 'Sachverhalt', nämlich den verhandelten Termin (im Laufe eines Dialogs eventuell auch eine kleine Anzahl unterschiedlicher Termine). Diesem müssen allerdings unter Umständen recht komplexe temporale Aussagen zugewiesen werden können. Somit scheiden der beschriebene Ansatz – und sicherlich auch verwandte Ansätze[2] – für eine Lösung der hier gestellten Aufgabe aus.

Nur ein weiterer ähnlicher Ansatz soll hier noch kurz angesprochen werden. In [Schmiedel, 1990] wird eine temporale Beschreibungslogik vorgestellt, ebenfalls ein DL-System, das einen temporalen Bereich direkt integriert. Mit den 'Zeit-Netzen' (englisch: *time nets*), die diesen Bereich bilden, können entweder Aussagen über die Gültigkeit einer Allen'schen Relation zwischen zwei Intervallen, relative Aussagen über die Dauer eines Intervalls oder aber Aussagen über seine Granularität (etwa: handelt es sich um das Vielfache einer Stunde?) gemacht werden. Die Intervalle selbst können dabei über Daten oder Daten mit Uhrzeiten benannt werden. Die Ausdrucksstärke dieses Systems kommt der geforderten also schon deutlich näher als bei [Fischer, 1992]. Etwas komplexere natürlichsprachliche Äußerungen mit temporalem Gehalt und auch einfache, aber unterspezifizierte Ausdrücke (beispielsweise lediglich eine Uhrzeit, ohne Datumsangabe) können aber auch hier nicht verarbeitet werden. Für die tatsächlich vorkommenden Ausdrücke wird die Notwendigkeit einer modelltheoretischen Semantik zwar angesprochen, diese dann aber für die Zeit-Netze selbst nicht mehr angegeben. Zur Frage, wie sprachnah formulierte zeitliche Ausdrücke auf einen klar definierten semantischen Bereich abbgebildet werden könnten, kann [Schmiedel, 1990] also auch keine Antworten liefern.

3.3.3 Integration Allgemeiner Konkreter Bereiche

Im letzten Abschnitt hat die 'temporale Komponente' in Form von einfachen Mengen von Zeitpunkten nicht ausgereicht, um die komplexen Ausdrücke, wie sie in natürlichsprachlichen Dialogen vorkommen können, angemessen repräsentieren zu können. Da im VERBMOBIL-Projekt im Rahmen der semantischen Auswertung, der auch die Behandlung temporaler Ausdrücke angehört, wie gesagt ein

[2] [Schild, 1991] stellt einen solchen Ansatz dar. Letztlich basieren hierauf auch die Arbeiten von [Fischer, 1992] und [Neuwirth, 1993].

beschreibungslogisches System zum Einsatz kommt, scheint es aber trotzdem interessant, Möglichkeiten der Einbindung einer wie auch immer gearteten temporalen Komponente in ein solches System zu untersuchen.

Hierzu liefert [Baader und Hanschke, 1991] einen sehr allgemeinen Ansatzpunkt, und zwar wird dort erörtert, auf welche Weise sogenannte *konkrete Bereiche* in KL-ONE-artige Sprachen integriert werden können. Ganz allgemein haben herkömmliche terminologische Systeme den Nachteil, daß das durch sie zu repräsentierende Wissen in Form von abstrakten logischen Zusammenhängen beschrieben werden muß. Schon einfachste Ganzzahlarithmetik kann auf diese Weise nicht behandelt werden. Das gilt natürlich erst recht für die Verarbeitung der im VERBMOBIL-Szenario relevanten zeitlichen Ausdrücke. Es wäre also wünschenswert, solche konkreten Bereiche und über ihnen definierte Prädikate mit einbeziehen zu können. Im folgenden soll erläutert werden, was unter konkreten Bereichen zu verstehen ist und welche Bedingungen an sie gestellt werden müssen, um die Integration in ein beschreibungslogisches System zu gestatten. (Die Integration als solche soll hier allerdings nicht Gegenstand der Betrachtung werden; die Leserschaft sei hierzu auf [Baader und Hanschke, 1991] verwiesen.)

Definition 3.2 *Ein* konkreter Bereich $\mathcal{D}$ *besteht aus einer Menge* $dom(\mathcal{D})$*, dem eigentlichen Bereich, und einer Menge* $pred(\mathcal{D})$*, den Prädikatsnamen von* $\mathcal{D}$*. Mit jedem dieser Prädikatsnamen* $P \in pred(\mathcal{D})$ *werden eine Stelligkeit* n *sowie ein* n*-stelliges Prädikat* $P^{\mathcal{D}} \subseteq dom(\mathcal{D})^n$ *assoziiert.*

Nun sollen die Bedingungen formuliert werden, die von einem konkreten Bereich erfüllt werden müssen, damit dieser 'zulässig', das heißt geeignet für die Integration in eine beschreibungslogische Sprache, ist. Die Menge der Prädikatsnamen $pred(\mathcal{D})$ heißt *abgeschlossen bezüglich der Negation*, wenn zu jedem P ein $Q \in pred(\mathcal{D})$ existiert mit $Q^{\mathcal{D}} = dom(\mathcal{D})^n \setminus P^{\mathcal{D}}$.

Für k (nicht notwendigerweise verschiedene) Prädikatsnamen $P_1, \ldots, P_k$ aus $pred(\mathcal{D})$ mit assoziierten Stelligkeiten $n_1, \ldots, n_k$ werde die Konjunktion

$$\bigwedge_{i=1}^{k} P_i(\underline{x}^{(i)})$$

betrachtet. Hierbei stehe $\underline{x}^{(i)}$ für ein n-Tupel von Variablen. Solch eine Konjunktion wird *erfüllbar* genannt genau dann, wenn es eine Belegung der Variablen mit Elementen aus $dom(\mathcal{D})$ gibt, für die diese Konjunktion *wahr* ist.

Definition 3.3 *Ein konkreter Bereich* $\mathcal{D}$ *heißt* zulässig *dann und nur dann, wenn er die folgenden drei Bedingungen erfüllt:*

(i) $pred(\mathcal{D})$ *ist abgeschlossen bezüglich der Negation,*

(ii) in $pred(\mathcal{D})$ *gibt es einen mit dem einstelligen Prädikat* $dom(\mathcal{D})$ *assoziierten Prädikatsnamen und*

(iii) das Erfüllbarkeitsproblem für endliche Konjunktionen der oben beschriebenen Art ist entscheidbar.

Ein konkreter Bereich, der nach dieser Definition zulässig ist, kann, wie in [Baader und Hanschke, 1991] beschrieben, erfolgreich in ein terminologisches System integriert werden. Gelingt es, eine Sprache für zeitliche Ausdrücke und geeignete auf ihr definierte Prädikate zu finden, die obige Bedingungen erfüllen, so könnte dieses System quasi als 'black box' an ein vielfältige Aufgaben wahrnehmendes DL-System angekoppelt werden. Zur Antwort auf die Frage der geeigneten Darstellung temporaler Ausdrücke selbst kann dieser Ansatz aber naturgemäß nichts beitragen. Er setzt lediglich zusätzliche Nebenbedingungen, deren Erfüllung im Hinblick auf diese Kopplung wünschenswert, ansonsten aber nicht unbedingt notwendig ist.

3.4 Zeitliche Ausdrücke bei Zugauskunftsdialogen

In [Hildebrandt *et al.*, 1993] und [Hildebrandt, 1994] wird ein System zur Verarbeitung zeitlicher Ausdrücke bei Zugauskunftsdialogen beschrieben. Einer der Hauptaspekte dieser Arbeit ist eine linguistische Detailanalyse. Für die Domäne Zugauskunft wurden vier Klassen von Ausdrücken identifiziert:

- Anfragen (zum Beispiel "wann" oder "ab wann")
- Angaben des Datums (zum Beispiel "am Montag" oder "vor dem 24.12.")
- Angaben der Tageszeit (zum Beispiel "abends")
- Angaben der Uhrzeit ("gegen halb neun" oder "um acht Uhr dreißig")

Die Klassifizierung wurde allerdings nicht zur Entwicklung einer geeigneten formalen Sprache herangezogen, vielmehr setzt die semantische Interpretation unmittelbar auf der syntaktischen Analyse auf. Die in [Hildebrandt, 1994] beschriebene Grammatik bezieht sich unmittelbar auf diejenige Teilmenge der deutschen Sprache, die die berücksichtigten Ausdrücke enthält. Ein solches Vorgehen scheint für die Domäne Terminvereinbarung nicht geeignet, da hier deutlich komplexere zeitliche Ausdrücke zu bewältigen sind.

[Hildebrandt, 1994] beschreibt eine inkrementelle semantische Interpretation, in deren erstem Schritt zu gegebenem zeitlichen Ausdruck dessen sogenannte *Kernzeit* sowie seine *Relativierung* ermittelt werden. Beispielsweise ist die dem Ausdruck "nach dem dritten Mai" entsprechende Kernzeit *3.5.*, und aus der Relativierung "nach" ist ersichtlich, daß der Zeitraum nach der Kernzeit betrachtet wird. Im nächsten Schritt werden diese Informationen auf *Zeittafeln* übertragen. Dies sind Tabellen mit je einer *von*- und einer *bis*-Spalte, deren Zeilen Angaben zum Datum (Tag und Monat) sowie zur Uhrzeit (Stunde und Minute) machen. Die Relativierung legt hierbei fest, in welche Felder die Kernzeit einzutragen ist. Enthält eine Äußerung mehrere zeitliche Ausdrücke, so können ihre Zeittafeln miteinander verschmolzen werden, was einem semantischen Schnitt entspricht. Auch die sukzessive Verfeinerung einer Zeitangabe im Verlaufe eines Dialogs kann so dargestellt werden. Die Verschmelzung der Ausdrücke "am neunten Mai" und "ab fünfzehn Uhr" wird folgendermaßen dargestellt:

	von	*bis*
Tag	9	9
Monat	5	5
Stunde		
Minute		

$\oplus$

	von	*bis*
Tag		
Monat		
Stunde	15	
Minute	00	

$\rightarrow$

	von	*bis*
Tag	9	9
Monat	5	5
Stunde	15	
Minute	00	

Für bestimmte Tageszeiten oder Feiertage sind entsprechende Zeittafeln vordefiniert. Der Verschmelzungsalgorithmus berücksichtigt auch die Ambiguität von Uhrzeiten, das heißt um eine Verschmelzung zu ermöglichen, wird der Stundenwert gegebenenfalls um 12 erhöht.

Insgesamt bietet dieser Ansatz wenig Hilfreiches zur Lösung der bei VERBMOBIL anstehenden Probleme. Aufgrund der Vielfalt der auftretenden zeitlichen Ausdrücke scheint es weiterhin ratsam, eine formale Repräsentationssprache als Zwischenstufe zwischen natürlicher Sprache und semantischer Interpretation zu definieren. Ob durch die relativ einfach strukturierten Zeittafeln die gesamte Breite der Interpretationsmöglichkeiten von zeitlichen Ausdrücken bei Terminvereinbarungsdialogen abgedeckt werden könnte, ist mehr als fraglich.

3.5 Zusammenfassung

In diesem Kapitel wurden einige Theorien vorgestellt, die zunächst vielversprechend im Hinblick auf die Verarbeitung zeitlicher Ausdrücke in Terminabsprachedialogen erschienen. Für die entscheidenden Aufgaben, nämlich die Definitionen einer Repräsentationsprache sowie einer geeigneten Semantik einer solchen Sprache, konnten jedoch keine wesentlichen Ideen extrahiert werden. Hauptgrund hierfür ist, daß in dieser Arbeit die Zeit selbst beziehungsweise die Formen ihrer Beschreibung in natürlicher Sprache von Interesse sind. Bei den vorgestellten Ansätzen hingegen ist Zeit aber eher *eine* Eigenschaft von Ereignissen oder Zuständen. In der Temporallogik etwa werden Ereignissen (Formeln) durch temporale Operatoren bestimmte Geltungszeiträume zugewiesen. Der Zeitbegriff ist dabei ein sehr primitiver; die Formeln auf der anderen Seite können für beliebige Ereignisse (Aussagen) stehen. Im VERBMOBIL-Szenario dagegen gibt es eigentlich nur ein 'Ereignis', nämlich den zu verhandelnden Termin, und Zeit kommt in zahlreichen Variationen (natürlichsprachlichen Beschreibungen) vor. Bei der Beschreibungslogik ist es ähnlich: ein recht einfacher Zeitbegriff wird mit einem fast beliebig komplexen Wissen kombiniert. Bei Allens Intervall-Kalkül hingegen geht es wirklich um die Zeit selbst. Allerdings weist er selbst in [Allen, 1983] (wie bereits erwähnt) darauf hin, daß sich sein Ansatz in erster Linie für Bereiche eignet, in denen temporale Information relativ ist und in denen Datierungen (also konkrete Zeitpunkte) keine Rolle spielen.

Da selbst die Arbeit zur Behandlung zeitlicher Ausdrücke bei Zugauskunftsdialogen keine nennenswerten Ergebnisse zur Lösung der anstehenden Probleme liefern konnte, wird Kapitel 2 die wesentliche Basis für das weitere Vorgehen sein. Im folgenden Kapitel werde ich eine ZeitGram-ähnliche Sprache, die Temporal Expression Language, vorstellen.

Kapitel 4

Syntax

Ein Satz kann in sehr verschiedenem Sinne die Zeit enthalten.
Ludwig Wittgenstein: Philosophische Grammatik.

Auf die in Kapitel 2 vorgenommene Analyse des VERBMOBIL-Korpus' aufbauend soll in diesem Kapitel eine formale Sprache zur Beschreibung zeitlicher Ausdrücke bei Terminvereinbarungsdialogen entwickelt werden. Diese Sprache werde ich TEMPORAL EXPRESSION LANGUAGE (kurz TEL) nennen. Bei ihrer Definition werden Teile von ZEITGRAM [Küssner und Stede, 1995] übernommen werden können, allerdings werden insbesondere die ebenfalls in Kapitel 2 aufgedeckten Schwachstellen zu wesentlichen Änderungen führen. Die wichtigste dieser Änderungen wird darin bestehen, künftig explizit zwischen Ausdrücken zur Beschreibung einer Dauer, solchen zur Beschreibung eines Zeitpunktes und solchen zur Beschreibung eines Termins zu unterscheiden.

Nach einigen einleitenden Bemerkungen im ersten Teil dieses Kapitels wird für jeden der drei Bereiche Dauern, Zeitpunkte und Termine durch die Angabe eines Grammatikfragments eine Sprache definiert werden. Das Kapitel endet mit einer kurzen Zusammenfassung.

4.1 Allgemeines

In diesem Kapitel wird eine umfangreiche Grammatik definiert werden. Hierbei wird allerdings kein bestimmtes Startsymbol festgelegt, das heißt, bei unterschiedlicher Wahl eines Startsymbols werden durch die Grammatik unterschiedliche Sprachen beschrieben. Die Menge all dieser Sprachen wird als TEMPORAL EXPRESSION LANGUAGE bezeichnet. Entsprechend ist jeder Ausdruck, der bei Wahl eines geeigneten Startsymbols durch die Grammatik produziert werden kann, ein TEL-Ausdruck. Natürlich werden nur einige wenige der definierbaren Subsprachen auch tatsächlich von Interesse sein, namentlich die, durch die entweder die Menge der Dauern oder die Menge der Zeitpunkte oder die Menge der Termine beschrieben werden, und auch nur für diese ausgewählten Sprachen wird später eine semantische Interpretation anzugeben sein.

4.1.1 Backus-Naur-Form

Zur Definition der Grammatik werde ich wie in [Küssner und Stede, 1995] eine erweiterte Backus-Naur-Form benutzen, die an dieser Stelle kurz beschrieben werden soll. Nonterminalsymbole beginnen mit Großbuchstaben, Terminalsymbole werden klein geschrieben. Außerdem wird die Menge der ganzen Zahlen den Terminalsymbolen zugerechnet. Eine nichtleere Liste von Ausdrücken, die alle der Kategorie `CATEGORY` angehören, wird als `CATEGORY+` notiert. Entsprechend wird mit `CATEGORY*` eine Liste bezeichnet, die auch leer sein kann.

Alle Produktionsregeln haben die folgende Grundform:

```
CATEGORY     ::=  SUBCATEGORY1 | SUBCATEGORY2 | ... | SUBCATEGORYN
```

Hierdurch wird definiert, daß ein Ausdruck zur Kategorie `CATEGORY` gehört, wenn er einer der Kategorien `SUBCATEGORY1` bis `SUBCATEGORYN` angehört. Auf der linken Seite einer Produktionsregel steht stets genau ein Nonterminalsymbol; es handelt sich also um eine kontextfreie Grammatik. Die Ausdrücke auf der rechten Seite können ebenfalls Nonterminale, Terminalsymbole oder zusammengesetzte Ausdrücke wie etwa `functor(ARGUMENT1,ARGUMENT2)` sein. Bei der Definition der Grammatik muß selbstverständlich darauf geachtet werden, daß es für jeden nonterminalen Ausdruck, der auf der rechten Seite einer Produktionsregel auftaucht, auch eine Regel gibt, bei der er auf der linken Seite steht.

Die Menge der Ausdrücke der Kategorie `CATEGORY`, also die Menge der Ausdrücke, die durch die Grammatik mit dem Startsymbol `CATEGORY` erzeugt werden können, wird als (die Sprache) L(`CATEGORY`) bezeichnet.

Mitunter haben zwei oder mehr zusammengesetzte Ausdrücke auf der rechten Seite einer Produktionsregel die gleiche Struktur und unterscheiden sich zum Beispiel nur durch eines ihrer Argumente oder durch ihren Funktor. Um solche Regeln kompakt darstellen zu können, besteht die Möglichkeit, auch innerhalb eines Ausdrucks eine Auswahl anzugeben. Mit <CATEGORY1|...|CATEGORYN> wird eine Ausdruck bezeichnet, der zu einer der Kategorien `CATEGORY1` bis `CATEGORYN` gehört.

Beispielsweise handelt es sich bei

```
CATEGORY     ::=  <functor1|functor2>(ARGUMENT1,<arg2a|arg2b>)
```

um eine kompakte Darstellung folgender Produktionsregel:

```
CATEGORY     ::=  functor1(ARGUMENT1,arg2a) |
                  functor1(ARGUMENT1,arg2b) |
                  functor2(ARGUMENT1,arg2a) |
                  functor2(ARGUMENT1,arg2b)
```

4.1.2 Umfang und Struktur

Ein Termin hat einen Anfang, ein Ende und eine Dauer. Bei Terminanfang beziehungsweise -ende handelt es sich um Zeitpunkte. Ich werde zunächst eine Sprache zur Beschreibung einer zeitlichen Dauer und sodann eine weitere Sprache zur

Beschreibung eines Zeitpunktes definieren. Diese Sprachen können dann zur Beschreibung der speziellen Merkmale eines Termins verwendet werden. In diesem Sinne ist der Aufbau der Grammatik der Temporal Expression Language modular.

Die explizite Unterscheidung zwischen Anfangs- und Endpunkten ist *notwendig*, um eine Semantik von Terminbeschreibungen definieren und ein Themengedächtnis zur Repräsentation des jeweils verhandelten Termins modelieren zu können. Solch eine Differenzierung ist allerdings nicht unbedingt erforderlich, wenn die Repräsentationssprache (nur) zur Generierung natürlicher Sprache benutzt werden soll. Zu diesem Zwecke genügt es möglicherweise, ausschließlich Ausdrücke der Sprachen zur Beschreibung von Dauern beziehungsweise Zeitpunkten zu betrachten.

Ein Ausdruck der zu entwerfenden Sprache soll stets genau *einen* Termin beschreiben. In Verbmobil gibt es zwar einige wenige Beispiele, in denen unter Umständen innerhalb ein und desselben Satzes zwei (oder auch noch mehr) Termine angesprochen werden. Statt eine solche Äußerung aber durch einen einzigen TEL-Ausdruck repräsentieren zu können, wird man zwei (beziehungsweise entsprechend mehr) Ausdrücke aufbauen müssen. Die Verwaltung der verschiedenen Terminbeschreibungen muß also von der Dialogverarbeitung geleistet werden, sie findet nicht 'innerhalb' von TEL statt. Allerdings könnte TEL möglicherweise als Subsprache für eine andere Repräsentationssprache dienen, die auch dazu in der Lage wäre.

Wie schon ZeitGram so soll auch TEL einerseits möglichst sprachnah sein und andererseits linguistische Varianten semantisch äquivalenter Ausdrücke möglichst zusammenfassen. Insbesondere sollen bei dieser Auflösung linguistischer Varianten Unterschiede zwischen den verschiedenen 'Input-Sprachen' (also Englisch und Deutsch) aufgehoben werden. Da es zum Beispiel für das deutsche "dreiviertel drei" keine direkte Entsprechung im Englischen gibt, bietet sich an, derartige Ausdrucksformen zu normalisieren. Dabei ist allerdings darauf zu achten, daß eine solche Normalisierung die auf der Temporal Expression Language noch zu definierende semantische Interpretation nicht vorwegnimmt. Müßte etwa schon während der Generierung des entsprechenden TEL-Ausdrucks inferiert werden, daß zum Beispiel "der zweite Samstag nach Ostern" im Jahre 1998 auf den 26. April fällt, so würde dies Sinn und Zweck des ganzen Unternehmens der Definition einer formalen Sprache in Frage stellen.

Einige spezielle Ausdrucksformen, die im Rahmen von Terminvereinbarungsgesprächen im Prinzip denkbar wären, im Verbmobil-Korpus aber nicht vorkommen, wurden bewußt nicht aufgenommen. Hierzu gehören zum Beispiel die Unterscheidung zwischen Sommer- und Winterzeit oder die Nennung unterschiedlicher Zeitzonen wie etwa in "three pm GMT". Uhrzeiten können maximal auf die Minute genau spezifiziert werden. Auch bei der Angabe der Länge einer Zeitdauer werden Angaben in Sekunden ausgeschlossen. Gleichermaßen werden Ausdrücke für Jahrzehnt, Jahrhundert oder Jahrtausend nicht in die Sprache aufgenommen werden. Es ist jedoch klar, daß es sich hierbei nicht um generelle Beschränkungen handelt, sondern lediglich um eine Abgrenzung gegenüber einigen Spezialfällen, die ohnehin kaum von Interesse sind.

Die im folgenden definierte Grammatik wird mitunter auch Ausdrücke mit einschließen, deren natürlichsprachliche Äquivalente nicht sinnvoll sind. Da die Grammatik aber nicht zur Generierung von Ausdrücken benutzt werden wird, ist solch eine Übergeneralisierung unkritisch. Wichtiger ist, daß es gelingt, für beliebige TEL-Ausdrücke eine semantische Interpretation zu berechnen. Ausdrücke, die aus *linguistischer* Sicht nicht sinnvoll sind, sollten erst gar nicht generiert werden (schließlich werden TEL-Ausdrücke letztlich aufgrund eines natürlichsprachlichen Inputs aufgebaut). Ausdrücke, die aus *semantischer* Sicht nicht sinnvoll sind, sollten mit Hilfe der zu entwerfenden Semantikkomponente als solche erkannt werden.

4.1.3 Beschreibung von Zahlenangaben

Bei der Beschreibung von Zeitpunkten oder Dauerangaben werden mitunter Zahlen eine Rolle spielen. Hierbei kommen natürliche Zahlen und bestimmte positive rationale Zahlen ("eine halbe Woche") in Frage. An zwei Stellen werden darüber hinaus auch ganze Zahlen benutzt werden, um bestimmte Ausdrücke in einer verallgemeinerten Form darstellen zu können. Die ganzen Zahlen selbst gehören zu den Terminalsymbolen von TEL.

Im folgenden werden die Kategorien zur Beschreibung ganzer Zahlen, zur Beschreibung natürlicher Zahlen sowie zur Beschreibung natürlicher Zahlen inklusive der 0 definiert.

```
INTEGER    ::=  0 | 1 | -1 | 2 | -2 | 3 | -3 | ...
NUMBER     ::=  1 | 2 | 3 | ...
NUMBER0    ::=  0 | NUMBER
```

Die in Frage kommenden rationalen Zahlen werden entweder als einfache Brüche oder als gemischte Zahlen angegeben. Außerdem werden auch die natürlichen Zahlen der Kategorie der positiven rationalen Zahlen zugerechnet.

```
FRACTION   ::=  NUMBER | NUMBER:NUMBER | NUMBER:NUMBER:NUMBER
```

So wird beispielsweise "dreiviertel" als `3:4` und "anderthalb" als `1:1:2` repräsentiert.

4.2 Ausdrücke zur Beschreibung einer Dauer

Das im folgenden definierte Grammatikfragment gibt an, welche Formen der Beschreibung einer zeitlichen Dauer möglich sind. Es orientiert sich an der in Abschnitt 2.1.3 vorgenommenen Klassifikation. Die Hauptkategorie für Ausdrücke zur Beschreibung einer Dauer wird weiter unten als `DURATION` definiert werden; die Sprache der zeitlichen Ausdrücke zur Angabe einer Dauer ist also $L($`DURATION`$)$.

4.2.1 Einfache Dauerangaben

Eine einfache Dauer wird durch die Angabe einer positiven rationalen Zahl sowie einer zeitlichen Einheit spezifiziert. Die möglichen Einheiten sind Jahre, Monate, Wochen, Tage, Stunden und Minuten. Darüber hinaus sind Ausdrücke wie "ein paar Wochen" möglich, bei denen keine explizite Zahlenangabe erfolgt. In diesem Falle wird anstelle einer Zahl der Ausdruck `several` verwendet.

Zur Beschreibung einfacher Dauerangaben wird die Kategorie `BASIC_DUR` definiert. Die Symbole zur Angabe der Einheit werden in der Kategorie `UNIT` zusammengefaßt.

```
BASIC_DUR    ::=  dur(<FRACTION|several>,UNIT)

UNIT         ::=  years | months | weeks | days | hours | minutes
```

Der natürlichsprachliche Ausdruck "anderthalb Stunden" wird also durch den TEL-Ausdruck `dur(1:1:2,hours)` beschrieben.

4.2.2 Bereiche

Durch die Angabe zweier Dauern kann ein Bereich von Dauern eingegrenzt werden. Die letztlich bezeichnete Dauer liegt irgendwo zwischen den beiden spezifizierten Randpunkten. Für solche Ausdrücke wird die Kategorie `RANGE` eingeführt. Die als Argumente übergebenen Randpunkte sind selbst Ausdrücke der Kategorie `DURATION`. Diese Kategorie umfaßt alle denkbaren Ausdrücke zur Beschreibung einer Dauer; die entsprechende Grammatikregel wird weiter unten angegeben werden.

```
RANGE        ::=  range(DURATION,DURATION)
```

Zum Beispiel wird ein Ausdruck wie "fünf Tage bis zwei Wochen" als `range(dur(5,days),dur(2,weeks))` repräsentiert. Meist werden erstes und zweites Argument die gleiche zeitliche Einheit haben. In der natürlichen Sprache wird die Angabe dieser Einheit für das erste Argument oft unterlassen ("zwei bis drei Tage"). Bei der Generierung des entsprechenden TEL-Ausdrucks muß diese dann ergänzt werden.

Neben den Bereichen mit zwei Randpunkten gibt es auch die sogenannten offenen Bereiche, die nur auf einer Seite durch die Angabe eines Dauerausdrucks begrenzt werden. Diese offenen Bereiche werden durch Ausdrücke der Kategorie `OPEN_RANGE` repräsentiert. Solch ein Bereich kann entweder zur linken oder zur rechten Seite hin offen sein. Im Gegensatz zu den geschlossenen Bereichen erlauben sowohl die deutsche als auch die englische Sprache Ausdrucksformen, bei denen der genannte Randpunkt selbst entweder noch zu dem eingegrenzten Bereich gehört oder bei denen er explizit außerhalb dieses Bereichs liegt. Später wird manchmal daher auch von der 'inklusiv'- und der 'exklusiv'-Variante die Rede sein. Der 'inklusiv'-Variante entsprechen die Funktoren `at_least` und `at_most`, der 'exklusiv'-Variante die Funktoren `more` und `less`.

```
OPEN_RANGE   ::=  <at_least|at_most|more|less>(DURATION)
```

Der Ausdruck “weniger als drei Stunden” wird also beispielsweise durch `less(dur(3,hours))` repräsentiert und “mindestens eine halbe Woche” durch `at_least(dur(1:2,weeks))`.

4.2.3 Unscharfe Dauerangaben

Um unscharfe Dauerangaben wie etwa “ungefähr zwei Stunden” repräsentieren zu können, wird die Kategorie `FUZZY_DUR` eingeführt.

```
FUZZY_DUR    ::=  fuzzy_dur(DURATION)
```

4.2.4 Anaphern

Auch Anaphern, die für zeitliche Ausdrücke stehen, werden in die Temporal Expression Language aufgenommen. Bevor die Berechnung der Semantik solcher Ausdrücke möglich wird, muß allerdings eine Anapherresolution, also ein Ersetzen der Anaphern durch die ihnen entsprechenden expliziten Ausdrücke erfolgen. Zu reinen Repräsentationszwecken sind aber auch TEL-Ausdrücke mit nicht aufgelösten Anaphern sinnvoll. Für Anaphern, die sich auf eine Dauer beziehen, wird die Kategorie `ANA_DUR` eingeführt.

```
ANA_DUR      ::=  ana_dur
```

Zum Beispiel könnte man “(für) genauso lang” (in geeignetem Kontext) als `ana_dur` darstellen beziehungsweise “(for) longer than that” als `more(ana_dur)`.

4.2.5 Top-Level

Die verschiedenen Ausprägungen von eine Dauer beschreibenden Ausdrücken werden nun in der Kategorie `DURATION` zusammengefaßt. Zusätzlich können durch den `set`-Operator mehrere Dauerangaben zu einer Disjunktion zusammengefügt werden. Sollten bei der Generierung eines TEL-Ausdrucks zu einem gegebenen Ausdruck in natürlicher Sprache Probleme auftauchen, den richtigen TEL-Ausdruck zu identifizieren, so gestattet die Grammatik auch die Darstellung mehrerer Alternativen durch einen einzigen Ausdruck. So können zwei oder mehr Alternativen parallel weiterverarbeitet werden. Eine Anzahl von Alternativen wird durch den `one_out_of`-Operator verknüpft.

```
DURATION     ::=  set(DURATION+) | one_out_of(DURATION+) |
                  BASIC_DUR | FUZZY_DUR | RANGE | OPEN_RANGE |
                  ANA_DUR
```

Sowohl `dur(2,weeks)` als auch `fuzzy_dur(dur(2,weeks))` haben nun also beide die Kategorie `DURATION`. Der natürlichsprachliche Ausdruck “zwei oder drei Stunden” wird dargestellt als `set([dur(2,hours),dur(3,hours)])`.

4.3 Ausdrücke zur Beschreibung eines Zeitpunktes

Die Struktur des im folgenden definierten Grammatikfragments orientiert sich an der in Abschnitt 2.1.4 vorgenommenen Klassifikation von Ausdrücken zur Beschreibung der Lage eines Termins. Die Lage eines Termins wird durch die Eingrenzung eines oder mehrerer Zeitpunkte beschrieben. Weiter unten wird die Kategorie `POINT` definiert werden. Ein Ausdruck der Sprache $L(\texttt{POINT})$ macht eine Aussage über die Lage eines Zeitpunktes. Hierbei muß der beschriebene Zeitpunkt nicht unbedingt immer vollständig spezifiziert werden, sondern seine Lage kann auch lediglich eingegrenzt werden. Daher kann ein Ausdruck der Kategorie `POINT` auch als eine (möglicherweise ebenfalls unterspezifizierte) Aussage über eine Reihe aufeinanderfolgender Zeitpunkte, also über einen Zeitraum, interpretiert werden.

4.3.1 Einfache Ausdrücke

Zur Beschreibung der Lage eines Zeitpunktes gibt es zahlreiche einfache Ausdrücke. Die meisten der im folgenden aufgeführten Kategorien wurden bereits in [Küssner und Stede, 1995] genauso oder wenigstens auf ähnliche Weise definiert. Ich diskutiere zunächst die verschiedenen Erscheinungsformen und gebe danach das entsprechende Grammatikfragment an.

Zu den einfachen Ausdrücken zur Beschreibung eines Zeitpunktes gehören unter anderem Angaben der Uhrzeit, des Wochentags, des Monats, der Jahreszeit oder des Jahres. Der Darstellung in [Küssner und Stede, 1995] folgend werden die meisten dieser Ausdrücke in der Form `TYP:WERT` repräsentiert werden. Eine solches Format ist in den Fällen, in denen der `WERT` eine Zahl ist, auch notwendig, um verschiedene Ausdruckstypen unterscheiden zu können.

4.3.1.1 Uhr- und Tageszeitangaben

Uhrzeiten werden in der Form $\texttt{tod:}h\texttt{:}m$ ('*time of day*') angegeben. Hierbei sind h und m natürliche Zahlen (inklusive 0). Natürlich sind nur bestimmte Werte für h und m sinnvoll. In diesem Falle wäre es auch möglich, diese Werte (wie in ZEITGRAM) erschöpfend aufzuzählen. Bei anderen Kategorien von Ausdrücken wird das aber teilweise nicht möglich sein. Daher wurde die Überprüfung von Zahlenwerten ganz aus der Definition der Syntax herausgenommen und in die semantische Komponente verlagert. Grundsätzlich *sollen* bei der Repräsentation einer Uhrzeit aber nur Werte zwischen `0:0` und `11:59` angegeben werden. Von linguistischen Varianten wie "dreiviertel acht" oder "zehn vor drei" wird also abstrahiert, da diese in vielen Fällen nicht sowohl im Englischen als auch im Deutschen eine direkte Entsprechung haben. Die Abbildung spezieller natürlichsprachlicher Ausdrucksformen auf die allgemeine Form $\texttt{tod:}h\texttt{:}m$ muß also im Rahmen der Generierung der TEL-Asudrücke geleistet werden. Uhrzeiten zwischen 12.00 Uhr und 23.59 Uhr sollten bei der Generierung ebenfalls auf Werte zwischen `0:0` und `11:59` normalisiert werden. In solchen Fällen kann zusätzlich durch den Ausdruck `pod:pm` (siehe unten) spezifiziert werden, daß es sich um einen Zeitpunkt nach zwölf Uhr mittags handelt. Ansonsten sind Uhrzeitangaben ambig; sie können sich stets auf einen Zeitpunkt vor oder auf einen nach zwölf Uhr mittags beziehen.

Der gerade erwähnte Ausdruck `pod:pm` gehört zur Klasse der Tageszeitangaben (‘*part of day*’). Solche Ausdrücke haben die Form `pod:POD`, wobei zur Kategorie `POD` unter anderem die Bezeichner `am` und `pm` gehören. Ihre Bedeutung entspricht dem englischen *am* beziehungsweise *pm*. [Küssner und Stede, 1995] weisen darauf hin, daß sie auch losgelöst von genauen Uhrzeitangaben als eigenständige Ausdrücke auftreten können, da im Englischen eine Formulierung wie “let's have a *pm* meeting tomorrow” durchaus gebräuchlich ist. Des weiteren gehören der Kategorie `POD` Bezeichner wie `morning` oder `evening` an, die auf die jeweilige Tageszeit verweisen. Im Englischen gibt es kein Wort für *Vormittag*; es wird meist mit *morning* übersetzt. Demnach muß sich die Semantik von “in the morning” von der von “am Morgen” unterscheiden. Erstere beinhaltet quasi die Semantiken von “am Morgen” *und* von “am Vormittag”. Also sollte es in diesem speziellen Falle auch unterschiedliche TEL-Ausdrücke geben. Neben dem einfachen `morning` werde ich `morning_ger1` für das deutsche *morgens* und `morning_ger2` für *vormittags* definieren.

Um unterspezifizierte Uhrzeitangaben wie “um viertel nach” repräsentieren zu können, gibt es Ausdrücke der Form `moh:`m (‘*minute of hour*’), wobei m wieder eine natürliche Zahl (inklusive 0) ist. Zum Beispiel wird “um viertel nach” als `moh:15` und “zur vollen Stunde” als `moh:0` dargestellt. Um unmittelbar auf den aktuellen Zeitpunkt zugreifen zu können, um also die Zeitangabe “jetzt” repräsentieren zu können, steht der Bezeichner `now` zur Verfügung.

4.3.1.2 Tagesangaben

Wochentage werden als `dow:DOW` (‘*day of week*’) dargestellt, wobei `DOW` die Terminale `mon`, `tue`, `wed`, `thu`, `fri`, `sat` und `sun` enthält. Beachte, daß etwa zwischen “am Montag” und “montags” nicht unterschieden wird. Da durch die TEMPORAL EXPRESSION LANGUAGE stets nur *ein* Termin beschrieben werden soll und da durch Ausdrücke der Sprache L(`POINT`) stets nur *ein* Zeitpunkt (beziehungsweise gegebenfalls eine Reihe unmittelbar aufeinanderfolgender Zeitpunkte) eingegrenzt werden soll, machen “am Montag” und “montags” gleichermaßen die einfache Aussage, daß der bezeichnete Zeitpunkt die Eigenschaft hat, an einem Montag zu liegen. Der im Ausdruck “montags” darüber hinaus mitschwingende Gedanke einer Frequenz findet keine Berücksichtigung. Im übrigen zeigen die für VERBMOBIL gesammelten Daten sehr deutlich, daß viele Sprecher eine derartige Unterscheidung ebenfalls *nicht* vornehmen, was an folgendem Beispiel besonders gut zu erkennen ist:

> “Soll ich Ihnen da von donnerstags auf Freitag ein Zimmer mitbuchen?”

Statt einen Wochentag zu benennen, kann ein Zeitpunkt auch als an einem Arbeitstag oder an einem Wochenende liegend beschrieben werden. Zu diesem

Zwecke gibt es Ausdrücke der Form `pow:POW` (‘*part of week*’), bei denen für `POW` entweder `workday` oder `weekend` eingesetzt wird.

Einfache deiktische Angaben, die sich auf einen bestimmten Tag beziehen, sind zum Beispiel “heute” (`today`), “morgen” (`tomorrow`) oder “gestern” (`yesterday`). Sie werden unter der Kategorie `DEICTIC_DAY` zusammengefaßt. Diese Ausdrücke können auch in der allgemeinen Form `deictic_day:INTEGER` angegeben werden. Zum Beispiel steht `deictic_day:0` für “heute”, `deictic_day:-1` für “gestern” und `deictic_day:2` für “übermorgen”. Da sich Terminvereinbarungen stets auf die Zukunft beziehen, müßten Ausdrücke wie zum Beispiel “gestern” nicht unbedingt in die Repräsentationssprache aufgenommen werden. Da dieses jedoch keine Schwierigkeiten bereitet und zusätzlich die Behandlung von (bei strenger Betrachtung domänenfremden) Ausdrücken wie “gestern war Mittwoch” gestattet, wurden sie trotzdem aufgenommen.

Tage im Monat werden durch `dom:`n (‘*day of month*’) beschrieben, wobei n eine natürliche Zahl ist. Welche Werte für n semantisch zulässig sind, hängt vom Kontext ab. Wurde beispielweise bereits spezifiziert, daß der gesuchte Zeitpunkt im Februar liegt, so führt die Angabe von `dom:30` zur Inkonsistenz.

Der Ausdruck “am fünfundzwanzigsten” wird als `dom:25` repräsentiert. Ein Ausdruck wie “der dritte” kann sich zwar auch auf den Tag eines Monats beziehen (und genau das wird in den allermeisten Fällen auch der Fall sein), ist allerdings im Prinzip ambig. Wurde etwa gerade nach dem “zweiten Montag im Mai” gefragt, so könnte sich “der dritte” schließlich auch auf den dritten Monatg im Mai beziehen. In ZEITGRAM wurde für solche Fälle der unterspezifizierte Ausdruck `xoy:`n (‘x *of* y’) eingeführt. Problematisch an dieser Lösung ist aber, daß sich Ordinale nicht zwangsläufig auf zeitliche Angaben beziehen müssen. Ein TEL-Ausdruck, der in etwa die gleiche Bedeutung hat wie `xoy:`n ist (der weiter unten definierte) Ausdruck `of(`n`,ana_point,ana_point)`. Diesem Ausdruck kann ganz offensichtlich erst nach einer zweifachen Anapherresolution eine Semantik zugeordnet werden. Das gleiche gilt natürlich auch für `xoy:`n, ist dort nur nicht unmittelbar ersichtlich. Weiter unten wird auch gezeigt werden, wie mehrere Interpretationsalternativen eines einen Zeitpunkt beschreibenden Ausdrucks zu einem einzigen TEL-Ausdruck zusammengefaßt werden können. Auf diese Weise können die verschiedenen Möglichkeiten, wofür der jeweilige `xoy`-Ausdruck stehen könnte, explizit dargestellt werden. Die Identifizierung aller im jeweiligen Falle denkbaren Alternativen erfordert allerdings eine linguistische Analyse, die *vor* der Generierung des TEL-Ausdrucks vorzunehmen ist. Beispielsweise kann sich “der dritte” nicht auf eine bestimmte Woche beziehen, “the third” hingegen schon.

4.3.1.3 Feiertage

Ein Zeitpunkt kann auch durch die Angabe eines Feiertages spezifiziert werden. Die für die TEMPORAL EXPRESSION LANGUAGE berücksichtigten Feiertage sind in Tafel 4.1 zusammengestellt. Ein einen Feiertag bezeichnender TEL-Ausdruck wird durch die Typkennung `holiday` eingeleitet. Zum Beispiel wird “am ersten Advent” durch den Ausdruck `holiday:first_advent` repräsentiert.

Neben den Feiertagen im engeren Sinne wurden auch noch sechs Zeiträume, die länger als ein Tag sind, aufgenommen: die Faschingszeit (`carnival`), die Karwoche (`holy_week`), die Osterzeit (`easter`), Pfingsten (`whitsun`), die Adventszeit (`advent`) und Weihnachten (`christmas`). Außerdem kann durch den einfachen TEL-Ausdruck `holiday` ausgedrückt werden, daß der beschriebene Zeitpunkt an einem Feiertag liegt (wenngleich auch unklar ist, an welchem). So ist es zum Beispiel auch möglich den Ausdruck "sonn- und feiertags" zu formalisieren.

4.3.1.4 Wochen-, Monats- und Jahresangaben

Bei Terminabsprachen ist des öfteren auch von Kalenderwochen die Rede. Solche Ausdrücke werden als `woy:`n ('*week of year*') für eine natürliche Zahl n dargestellt. Also steht zum Beispiel `woy:10` für "die zehnte Kalenderwoche" oder natürlich auch "die zehnte Woche des Jahres".

Monatsangaben werden in der Form `month:MONTH` gemacht. Die Kategorie `MONTH` umfaßt die Symbole `jan`, `feb`, `mar`, etc. Bei Äußerungen wie "am vierzehnten ersten" wird der sich auf den Monat beziehende Teilausdruck als `month:jan` repräsentiert, das heißt es gibt keine separate Kategorie für durch Ordinale spezifizierte Monate.

In seltenen Fällen könnten auch Jahreszeiten bei der Beschreibung eines Zeitpunktes eine Rolle spielen. Sie werden durch `season:SEASON` dargestellt, wobei `SEASON` einen der Werte `spring`, `summer`, `fall` beziehungsweise `winter` annimmt. Außerdem sollen Ausdrücke wie "im dritten Quartal" oder "im ersten Halbjahr" dargestellt werden können. Die Semantik solcher Ausdrücke ist eindeutig und insbesondere nicht vom Zeitpunkt der Äußerung abhängig (im Gegensatz etwa zu dem deiktischen Ausdruck "im letzten halben Jahr"). Die Repräsentation erfogt durch Ausdrücke der Form `qoy:`n ('*quarter of year*') beziehungsweise `hoy:`n ('*half-year of year*') für natürliche Zahlen n zwischen 1 und 4 beziehungsweise zwischen 1 und 2.

Jahreszahlen werden in der Form `year:`n notiert. Hierbei müssen für n vierstellige Angaben gemacht werden, das heißt "im Jahre achtundneunzig" wird als `year:1998` repräsentiert und nicht etwa als `year:98`.

4.3.1.5 Grammatik

Die gerade diskutierten grundlegenden TEL-Ausdrücke zur Beschreibung eines Zeitpunktes werden in der Kategorie `BASIC` zusammengefaßt. Im folgenden ist das entsprechende Grammatikfragment angegeben.

```
BASIC     ::=  now | moh:MOH | tod:TOD | pod:POD | dow:DOW |
               pow:POW | DEICTIC_DAY | dom:DOM | woy:WOY|
               month:MONTH | season:SEASON | qoy:QOY |
               hoy:HOY | year:YEAR | holiday | holiday:HOLIDAY

MOH       ::=  NUMBER0

TOD       ::=  NUMBER0:NUMBER0
```

Erster Advent	*First Sunday of Advent*
Zweiter Advent	*Second Sunday of Advent*
Dritter Advent	*Third Sunday of Advent*
Vierter Advent	*Fourth Sunday of Advent*
Allerheiligen	*All Saints Day*
Aschermittwoch	*Ash Wednesday*
Buß- und Bettag	*Day of Prayer and Repentance*
Christi Himmelfahrt	*Ascension Day*
Dreikönigsfest	*Epiphany*
Erntedankfest	*Harvest Festival*
Faschingsdienstag	*Shrove Tuesday*
Fronleichnam	*Corpus Christi*
Gründonnerstag	*Maundy Thursday*
Heilig Abend	*Christmas Eve*
Karfreitag	*Good Friday*
Karsamstag	*Holy Saturday*
Maifeiertag	*May Day*
Mariä Himmelfahrt	*Assumption Day*
Muttertag	*Mother's Day*
Neujahr	*New Year's Day*
Nikolaustag	*St Nicholas' Day*
Ostermontag	*Easter Monday*
Ostersonntag	*Easter Sunday*
Pfingstmontag	*Whit Monday*
Pfingstsonntag	*Whit Sunday*
Reformationstag	*Reformation Day*
Rosenmontag	*Monday before Lent*
Silvester	*New Year's Eve*
Tag der Deutschen Einheit	*German Unity Day*
Totensonntag	*Last Sunday before Advent*
Valentinstag	*St Valentine's Day*
Volkstrauertag	*National Day of Mourning*
Erster Weihnachtsfeiertag	*Christmas Day*
Zweiter Weihnachtsfeiertag	*Boxing Day*

Tafel 4.1: Liste der berücksichtigten Feiertage

```
POD          ::=  am | pm | morning | morning_ger1 | morning_ger2 |
                  midday | afternoon | evening | night | daytime
DOW          ::=  mon | tue | wed | thu | fri | sat | sun
POW          ::=  workday | weekend
DEICTIC_DAY  ::=  today | tomorrow | yesterday |
                  deictic_day:INTEGER
DOM          ::=  NUMBER
WOY          ::=  NUMBER
MONTH        ::=  jan | feb | mar | apr | may | jun | jul | aug |
                  sep | oct | nov | dec
SEASON       ::=  spring | summer | fall | winter
QOY          ::=  NUMBER
HOY          ::=  NUMBER
YEAR         ::=  NUMBER
HOLIDAY      ::=  advent | first_advent | second_advent |
                  third_advent | fourth_advent | all_saints_day |
                  ascension_day | ash_wednesday | assumption_day |
                  boxing_day | carnival | christmas |
                  christmas_day | christmas_eve | corpus_christi |
                  day_of_prayer_and_repentance | easter |
                  easter_monday | easter_sunday | epiphany |
                  german_unity_day | good_friday |
                  harvest_festival | holy_saturday | holy_week |
                  last_sunday_before_advent | maundy_thursday |
                  may_day | monday_before_lent | mother_s_day |
                  national_day_of_mourning | new_year_s_day |
                  new_year_s_eve | reformation_day |
                  shrove_tuesday | st_nicholas_day |
                  st_valentine_s_day | whitsun | whit_monday |
                  whit_sunday
```

4.3.2 Generische Ausdrücke

Bestimmte Ausdrücke wie “Woche” oder “Monat” stellen für sich genommen keine sinnvolle Beschreibung eines Zeitpunktes dar, können sehr wohl aber als Argumente komplexerer Ausdrücke auftreten. Solche Ausdrücke werde ich im folgenden als *generisch* bezeichnen, sie fallen unter die Kategorie `GENERIC`.

```
GENERIC      ::=  day | week | month | quarter_year | half_year |
                  year
```

Beipielsweise kann von “dieser Woche” (anaphorisch oder deiktisch) oder von der “dritten Woche nach Ostern” die Rede sein. Auch durch einen Ausdruck wie "am Anfang der Woche" wird ein Zeitpunkt eingegrenzt. Der Ausdruck "Woche" an sich, sollte er tatsächlich zur Beschreibung eines Zeitpunktes verwendet werden, macht allenfalls die Aussage, daß der gesuchte Zeitpunkt "in (irgend)einer Woche" liegt. Es könnte also jeder Zeitpunkt gemeint sein. Daher dürfen generische Ausdrücke nicht alleine stehen. `GENERIC` ist dementsprechend auch keine Subkategorie von `POINT`.

Durch den Ausdruck "wöchentlich" werden zwei Arten von Information transportiert. Zum einen handelt es sich um Frequenz; es wird ein immer wiederkehrendes Ereignis beschrieben. Die Repräsentation von Frequenz liegt aber außerhalb des Umfangs von TEL. Was bleibt, ist die Aussage, daß der beschriebene Zeitpunkt "in einer Woche" liegt[1], was wie gesagt überhaupt nicht repräsentiert zu werden braucht.

4.3.3 Komplexe Ausdrücke

Als nächstes sollen die in Abschnitt 2.1.4 identifizierten Klassen komplexer Ausdrücke zur Beschreibung eines Zeitpunktes formalisiert werden (siehe dort insbesondere Tafel 2.2). Weiter unten werden die diesen Klassen entsprechenden TEL-Kategorien definiert werden. Es sind dies im einzelnen `LIMIT` (Intervalle mit einem Randpunkt), `LIMITS` (Intervalle mit zwei Randpunkten), `SHIFTED` (Verschiebungen), `RELATED` (Relationale Angaben), `ORDINAL` (Abzählungen) sowie `DEICTIC` (Deiktische Angaben). Sie alle werden in die Kategorie `COMPLEX` gefaßt.

```
COMPLEX      ::=  LIMIT | LIMITS | SHIFTED | RELATED | ORDINAL |
                  DEICTIC
```

Als Argumente treten in den meisten Fällen Ausdrücke der Kategorien `POINT` und `GENERIC` auf.

4.3.3.1 Intervalle

Es werden zwei Arten von Intervallen unterschieden: solche mit zwei und solche mit einem Randpunkt. Letztere bezeichne ich auch als einseitig unbegrenzte Intervalle; ihnen entspricht die Kategorie `LIMIT`. Diese einseitig unbegrenzten Intervalle können entweder zur linken Seite hin (also in Richtung der Vergangenheit) oder zur rechten Seite hin (also in Richtung der Zukunft) unbegrenzt sein. In beiden Fällen kann darüber hinaus zwischen einer 'inklusiv'- und einer 'exklusiv'-Variante unterschieden werden. Ein Beispiel für ein rechtsseitig unbegrenztes 'inklusiv'-Intervall ist etwa der Ausdruck "ab dem dritten". Er wird als `in_after(dom:3)` formalisiert. Die möglichen Funktoren sind unter der Kategorie `BEFORE_AFTER` zusammengefaßt. Ist nicht erkennbar, ob es sich um ein 'inklusiv'- oder ein 'exklusiv'-Intervall handelt, so sollen `before` beziehungsweise `after` verwendet werden.

[1]Vergleiche hierzu auch die Nicht-Unterscheidung der Ausdrücke "montags" und "am Montag", siehe Abschnitt 4.3.1.2.

```
LIMIT        ::=  BEFORE_AFTER(POINT)

BEFORE_AFTER ::=  before | after | in_before | in_after |
                  ex_before | ex_after
```

Intervalle mit zwei Randpunkten werden als Ausdrücke der Kategorie `LIMITS` repräsentiert. Hier unterscheide ich nicht zwischen 'inklusiv'- oder 'exklusiv'-Varianten, da es hierzu weder im Deutschen noch im Englischen (jedenfalls keine eindeutigen) Entsprechungen gibt. Es kann davon ausgegangen werden, daß Angaben wie "vom dritten bis zum zehnten" oder "zwischen Samstag und Dienstag" im allgemeinen 'inklusiv' gemeint sind. Der letzte Ausdruck kann zum Beispiel als `between(dow:sat,dow:tue)` formalisiert werden.

Mit "von Dienstag bis Donnerstag" könnte im Prinzip jeder Zeitraum gemeint sein, der an einem Dienstag beginnt und an irgendeinem (sich zeitlich danach befindlichen) Donnerstag endet. Gerade im Bereich von Terminvereinbarungsdialogen kann aber davon ausgegangen werden, daß normalerweise der auf den jeweiligen Dienstag unmittelbar folgende Donnerstag gemeint ist. Um ausdrücken zu können, daß die in einem `LIMITS`-Ausdruck angegebenen Randpunkte unmittelbar aufeinander folgen, daß sie also so zu konkretisieren sind, daß ihr Abstand minimal wird, wird zusätzlich der Funktor `min_between` eingeführt. In der Praxis ist diese Variante zu bevorzugen.

```
LIMITS       ::=  between(POINT,POINT) | min_between(POINT,POINT)
```

4.3.3.2 Verschiebungen

Zur Klasse der Verschiebungen zählen Ausdrücke wie "drei Wochen nach Ostern". Der Betrag, um den verschoben wird, wird als ein Ausdruck der Kategorie `DURATION` angegeben. Der Referenzzeitpunkt hat den Typ `POINT`. Eine Verschiebung in die Zukunft wird durch den Funktor `pos_shift` angedeutet. Die Entsprechung für die Vergangenheit ist `neg_shift`. Beachte, daß der durch den Gesamtausdruck beschriebene Zeitpunkt trotzdem in der Zukunft liegen kann: "zwei Jahre vor dem Jahr 2010" wird zum Beispiel dargestellt als `neg_shift(dur(2,years),year:2010)`.

Ein Spezialfall sind solche Verschiebungen, bei denen der Referenzzeitpunkt identisch ist mit dem Sprechzeitpunkt und dabei normalerweise nicht explizit genannt wird. Beispiele hierfür sind "in drei Stunden" oder "vor einem Monat". In diesen Fällen genügt die Angabe der Dauer, um deren Betrag verschoben werden soll. Als Funktoren stehen `in` und `ago` zur Auswahl. Letzterer wird mit der gleichen Begründung wie auch schon deiktische Tagesangaben, die sich auf Vergangenes beziehen, aufgenommen.

```
SHIFTED      ::=  <pos_shift|neg_shift>(DURATION,POINT) |
                  <in|ago>(DURATION)
```

4.3.3.3 Abzählbare Ausdrücke (als Argumente)

Bei den verbleibenden drei Klassen komplexer Ausdrücke (den Relationalen Angaben, den Abzählungen und den Deiktischen Angaben) spielen bestimmte Ausdrücke eine Rolle, die die Eigenschaft haben, *abzählbar* zu sein. Dies sind immer diejenigen Ausdrücke, auf die sich der Gesamtausdruck letztlich bezieht. Zum Beispiel werden zur Interpretation des Ausdrucks "der dritte Samstag nach Neujahr" drei Samstage abgezählt. Gleiches gilt für den "dritten Samstag im Mai". Da "der übernächste Montag" auch durch den Ausdruck "der zweite Montag nach dem jetzigen Augenblick" beschrieben werden kann, handelt es sich auch hier um eine Abzählung (in diesem Fall von Montagen).

Die zur Abzählung in Frage kommenden Arten von Ausdrücken werden zur Kategorie `COUNTABLE` zusammengefaßt. Hierzu gehören insbesondere die generischen Ausdrücke ("die dritte Woche im Juni"). Komplexe Ausdrücke selbst können sicherlich nicht abgezählt werden, und auch nur eine begrenzte Menge der einfachen Ausdrücke kommt in Frage. Der Ausdruck "die ersten drei Tage nach dem achten Mai" bezeichnet einen Zeitraum, welcher die Dauer von drei Tagen hat und nach dem achten Mai liegt. Der bezeichnete Ausdruck ist also selbst keine Dauer, sondern eben ein Zeitraum (oder ein Intervall), dessen Dauer angegeben wird. Solche Teilausdrücke haben in TEL die Form `int:DURATION`. Hierbei hat die Kennung `int` die Funktion einer Typkonvertierung.[2] Um auch Ausdrücke wie "der dritte im Juni" darstellen zu können, wird auch das (bisher noch nicht definierte) Symbol `ana_point` zur Kennzeichnung eines anaphorischen Bezeichners in die Kategorie `COUNTABLE` aufgenommen. Aus Gründen, die in Abschnitt 4.3.5 erklärt werden sollen, wird die "Zwischenkategorie" `COUNTABLE1` eingeführt, die alle abzählbaren Ausdrücke außer denen der Form `int:DURATION` enthält.

```
COUNTABLE    ::=  COUNTABLE1 | int:DURATION

COUNTABLE1   ::=  GENERIC | pod:POD | dow:DOW | pow:POW |
                  dom:DOM | season:SEASON | holiday |
                  holiday:HOLIDAY | ana_point
```

Zwar sind auf der linguistischen Ebene nicht alle so produzierbaren Ausdrücke immer sinnvoll (bei "der fünfte Nachmittag nach Pfingsten" ist dies zum Beispiel fraglich), jedoch ist sichergestellt, daß die bezeichneten Ausdrücke im intuitiven Sinne abzählbar sind und somit stets eine formale Semantik berechnet werden kann. Eine gewisse Übergeneralisierung ist hier also unkritisch. Für die Klasse der Deiktischen Angaben sind alle Ausdrücke der Kategorie `COUNTABLE` sinnvolle Argumente.[3]

4.3.3.4 Relationale Angaben

Die Klasse der Relationalen Angaben wird durch die Kategorie `RELATED` formalisiert. Der entscheidende Unterschied zu den Verschiebungen ist hier, daß die

[2] Beachte den Unterschied zur Definition von `SHIFTED`, bei der solch eine Konvertierung bewußt *nicht* erfolgt.

[3] Vielleicht mit Ausnahme von `pod:am`, `pod:pm` und `pod:daytime`.

Art des beschriebenen Zeitraumes direkt angegeben wird. Bei der Verschiebung "zwei Wochen nach Ostern" ist der letztlich beschriebene Zeitraum nicht explizit benannt, bei Relationalen Angaben wie "das *Wochenende* um den zehnten" oder "der *Sonntag* in zwei Wochen" hingegen schon.

Zur ersten Gruppe von Relationalen Angaben zählen Ausdrücke wie "die Woche um den siebenundzwanzigsten". Solche Ausdrücke werden durch den Funktor `the_around` eingeleitet. Das erste Argument ist vom Typ `COUNTABLE` (im Beispiel "Woche"), das zweite eine beliebige Beschreibung eines Referenzzeitpunktes. Einen Sonderfall stellen Ausdrücke dar, bei denen eine Woche durch die Angabe zweier Referenzzeitpunkte spezifiziert wird: "die Woche vom dritten bis zum neunten". Hierfür gibt es den Funktor `week_between`. Für andere Objekte (als Wochen) ist eine derartige Konstruktion nicht möglich.

Um einen Zeitraum vor oder nach einem Referenzzeitraum zu spezifizieren, gibt es die Funktoren `the_before` und `the_after`. Der spezifizierte Zeitraum muß ein Ausdruck der Kategorie `COUNTABLE` sein. Der *wievielte* Zeitraum der spezifizierten Art gemeint ist, wird im allgemeinen durch die Angabe einer Zahl bestimmt. So wird zum Beispiel "die zweite Woche nach Pfingsten" als `the_after(2,week,holiday:whitsun)` repräsentiert. Die Angabe einer Zahl kann aber auch unterbleiben, wie beispielsweise bei "die Woche nach Pfingsten".

Einen weiteren Sonderfall stellt die Angabe eines Wochentags in Bezug auf einen Referenzzeitpunkt dar. Seine Lage wird durch die Angabe einer Dauer (den Abstand zwischen Referenzpunkt und beschriebenem Punkt) und einen der Funktoren `dow_before` und `dow_after` spezifiziert. Ein Beispiel hierfür ist "der Montag drei Wochen nach Ostern". Dieser Ausdruck wird dargestellt als `dow_after(dow:mon,dur(3,weeks),holiday:easter)`. Auch Ausdrücke wie "Montag in drei Wochen" gehören zu dieser Gruppe. Der hier unterschlagene Referenzzeitpunkt wird implizit mit dem aktuellen Augenblick gleichgesetzt.

```
RELATED      ::=  the_around(COUNTABLE,POINT) |
                  week_between(POINT,POINT) |
                  <the_before|the_after>(NUMBER,COUNTABLE,POINT) |
                  <the_before|the_after>(COUNTABLE,POINT) |
                  <dow_before|dow_after>(dow:DOW,DURATION,POINT) |
                  <dow_before|dow_after>(dow:DOW,DURATION)
```

4.3.3.5 Abzählungen

Bereits bei Ausdrücken der Kategorie `RELATED` findet in einigen Fällen eine Abzählung statt. Als zur Klasse der Abzählungen gehörig bezeichne ich jedoch nur solche Ausdrücke, bei denen gewisse Objekte (Ausdrücke der Kategorie `COUNTABLE`) innerhalb eines größeren Bezugszeitraumes abgezählt werden. Diese Ausdrücke bilden die Kategorie `ORDINAL`. "Der zweite Sonntag im Mai" wird zum Beispiel als `of(2,dow:sun,month:may)` dargestellt. In diesem Falle beginnt die Abzählung am Anfang des Bezugszeitraumes. Es kann aber auch an dessen Ende begonnen werden. Für solche Fälle wird der Funktor `last_of` eingeführt. Beispielsweise wird "der vorletzte Sonntag des Jahres" als `last_of(2,dow:sun,year)` geschrieben. Aus diesem Beispiel ist ersichtlich, daß als Bezugszeiträume auch generische Ausdrücke in Frage kommen. Soll auf das letzte Objekt der bezeichneten Art im Bezugszeitraum verwiesen werden, so kann die Angabe einer Zahl auch entfallen.

```
ORDINAL      ::=  of(NUMBER,COUNTABLE,<POINT|GENERIC>) |
                  last_of(NUMBER,COUNTABLE,<POINT|GENERIC>) |
                  last_of(COUNTABLE,<POINT|GENERIC>)
```

Mit Einführung des `of`-Operators werden einige einfache Ausdrücke, wie etwa solche der Form `dom:DOM`, redundant. Zum Beispiel könnte statt `dom:5` auch `of(5,day,month)` geschrieben werden. In die Kategorie `BASIC` wurden diejenigen Ausdrücke, die normalerweise ohne Bezugszeitraum genannt werden, trotz dieser Redundanz aufgenommen.

4.3.3.6 Deiktische Angaben

Deiktische Angaben sind solche Ausdrücke, bei denen ein deiktischer Bezeichner auf einen zeitlichen Ausdruck (der Kategorie `COUNTABLE`) angewandt wird. Sie werden als Ausdrücke der Kategorie `DEICTIC` formalisiert. Als deiktische Bezeichner stehen `this`, `next` und `last` zur Verfügung. Außerdem können, wie auch schon bei Ausdrücken der Kategorie `DEICTIC_DAY`, über den Funktor `deictic` und die Angabe einer ganzen Zahl Deiktische Angaben in einer verallgemeinerten Form repräsentiert werden. Zum Beispiel steht `deictic(-2,dow:fri)` für den Ausdruck "am vorletzten Freitag".

```
DEICTIC      ::=  this(COUNTABLE) | next(COUNTABLE) |
                  last(COUNTABLE) | deictic(INTEGER,COUNTABLE)
```

4.3.3.7 Übersicht: Komplexe Ausdrücke

Die folgende Liste bietet anhand einiger einfacher Beispiele eine Übersicht über die gerade eingeführten Kategorien von TEL-Ausdrücken zur Repräsentation komplexer Beschreibungen der Lage eines Zeitpunktes.

- Kategorie `LIMIT` (Intervalle mit einem Randpunkt)
  - "vor dem ersten Advent"
    `ex_before(holiday:first_advent)`
  - "ab heute"
    `in_after(today)`
- Kategorie `LIMITS` (Intervalle mit zwei Randpunkten)
  - "von morgen bis zum ersten"
    `min_between(tomorrow,dom:1)`
  - "zwischen September und Januar"
    `min_between(month:sep,month:jan)`

- Kategorie `SHIFTED` (Verschiebungen)
  - "drei Tage vor Ostern"
    `neg_shift(dur(3,days),holiday:easter)`
  - "in zweieinhalb Wochen"
    `in(dur(2:1:2,weeks))`
- Kategorie `RELATED` (Relationale Angaben)
  - "das Wochenende um den vierten"
    `the_around(pow:weekend,dom:4)`
  - "die Woche vom ersten bis zum fünften"
    `week_between(dom:1,dom:5)`
  - "der zweite Mittwoch vor Pfingsten"
    `the_before(2,dow:wed,holiday:whitsun)`
  - "der Tag nach Allerheiligen"
    `the_after(day,holiday:all_saints_day)`
  - "der Sonntag zwei Wochen vor Pfingsten"
    `dow_before(dow:sun,dur(2,weeks),holiday:whitsun)`
  - "Montag in acht Tagen"
    `dow_after(dow:mon,dur(8,days))`
- Kategorie `ORDINAL` (Abzählungen)
  - "die zweite Juliwoche"
    `of(2,week,month:jul)`
  - "der vorletzte Tag des Monats"
    `last_of(2,day,month)`
  - "die letzten zehn Tage im Mai"
    `last_of(int:dur(10,days),month:may)`
- Kategorie `DEICTIC` (Deiktische Angaben)
  - "der nächste Feiertag"
    `next(holiday)`
  - "die nächsten anderthalb Stunden"
    `next(int:dur(1:1:2,hours)`
  - "vorletzte Woche"
    `deictic(-2,week)`

4.3.4 Unscharfe und Modifizierte Ausdrücke

Zeitliche Ausdrücke zur Beschreibung eines Zeitpunktes können durch Modifikatoren wie zum Beispiel "früh" oder "am Anfang von" modifiziert werden. Die in die TEMPORAL EXPRESSION LANGUAGE aufgenommenen Modifikatoren werden in die Kategorie `MOD` gefaßt. Sie können auf Ausdrücke der Kategorien `POINT` und `GENERIC` angewandt werden. Das Ergebnis ist ein Ausdruck der Kategorie `MODIFIED`.

```
MODIFIED    ::=  MOD(<POINT|GENERIC>)

MOD         ::=  early | late | begin | middle | end |
                 firsthalf | secondhalf
```

Beispielsweise wird "früh am Abend" als `early(pod:evening)` und "am Anfang der Woche" als `begin(week)` repräsentiert. In ZEITGRAM dürfen Ausdrücke der Kategorie `MOD` auch ohne Argumente gebraucht werden. Eine Interpretation ist ohne Angabe eines Arguments jedoch nicht möglich. In den Fällen, in denen für ZEITGRAM beispielsweise der Ausdruck `end` generiert würde, sollte für TEL `end(ana_point)` gewählt werden (`ana_point` verweist auf eine Anapher). Beachte, daß mit "früh" oder "spät" gegebenenfalls auch eine Tageszeit gemeint sein kann. Dann können wahlweise auch `pod:morning` beziehungsweise `pod:evening` verwendet werden. Zum Beispiel könnte "sonntags früh" sowohl als `early(dow:sun)` als auch als `[dow:sun,pod:morning]` (das ist die Konjunktion der Ausdrücke `dow:sun` und `pod:morning`) repräsentiert werden.

Unscharfe Angaben werden durch Ausdrücke der Kategorie `FUZZY` repräsentiert. Der Funktor `fuzzy` wird analog zu den Modifikatoren gebraucht. Allerdings kommen generische Ausdrücke hier als Argumente nicht in Frage. Der Ausdruck "um den dritten" wird genauso wie "ungefähr am dritten" als `fuzzy(dom:3)` dargestellt.

```
FUZZY       ::=  fuzzy(POINT)
```

4.3.5 Quantifizierte Ausdrücke

Quantifizierte Ausdrücke spielen in der TEMPORAL EXPRESSION LANGUAGE lediglich eine untergeordnete Rolle. Das liegt daran, daß TEL-Ausdrücke dazu dienen, Aussagen über genau *einen* Zeitpunkt zu machen. Aus Ausdrücken wie "an allen Montagen", "nur montags" oder "an einem Montag" kann im wesentlichen nur die Information, daß es um einen an einem Montag liegenden Zeitpunkt geht, extrahiert werden. Die Bedutung des Ausdrucks "montags nicht" hingegen unterscheidet sich von der von "nicht an allen Montagen". Im zweiten Falle werden nur manche Montage ausgeschlossenen. Ähnlich verhält es sich mit "an zwei Montagen". Dadurch wird ausgedrückt, daß der gesuchte Zeitpunkt an einem von zwei Montagen liegt, wobei allerdings nicht bekannt ist, um welche Montage es sich handelt.

Als Quantoren kommen Zahlen ("an (nur) einem Montag", "an zwei Montagen", etc.), `several` ("an einigen Monatgen") sowie `every` ("an allen Montagen") in Frage. Außerdem können diese Quantoren durch `not` negiert werden, um zum Beispiel zwischen "an zwei Montagen nicht" (Negation des Gesamtausdrucks) und "nicht an zwei Montagen" (Negation des Quantors "zwei") unterscheiden zu können. Die quantifizierten Ausdrücke bilden die Kategorie `QUANTIFIED`.[4] Auch Ausdrücke wie “an jedem zweiten Montag” oder “an jedem dritten Montag” werden in diese Kategorie gefaßt. Quantoren haben für solche Ausdrücke die Form `every:`n mit einer natürlichen Zahl n. Auf diese Weise quantifizierte Ausdrücke müssen natürlich abzählbar sein, weshalb das an zweiter Stelle stehende Argument auf Ausdrücke der Kategorie `COUNTABLE1` (siehe Abschnitt 4.3.3.3) beschränkt wird. An dieser Stelle kommen Ausdrücke der Form `int:DURATION` nicht in Frage. Das ist der Grund, weshalb die Kategorie `COUNTABLE1` eingeführt wurde. Die Negation von Quantoren der Form `every:`n kann ausgeschlossen werden.

```
QUANTIFIED   ::=  quantified(QUANTIFIER,<POINT|GENERIC>) |
                  quantified(not(QUANTIFIER),<POINT|GENERIC>) |
                  quantified(every:NUMBER,COUNTABLE1)

QUANTIFIER   ::=  NUMBER | several | every
```

4.3.6 Anaphern

Um anaphorische Ausdrücke, die sich auf die Lage von Zeitpunkten beziehen, formalisieren zu können, wird die Kategorie `ANA_POINT` eingeführt. Im einfachsten Falle ist die Konstante `ana_point` zu verwenden. Darüber hinaus stehen die Funktoren `that` und `other` zur Verfügung. Zum Beispiel wird “an jenem Samstag” durch den Ausdruck `that(dow:sat)` dargestellt. Bei der Anapherresolution steht dann die Information, daß es sich bei dem gesuchten Ausdruck um die Beschreibung eines an einem Samstag liegenden Zeitpunktes handelt, zur Verfügung. Gleiches gilt für den Ausdruck “an einem anderen Samstag”, der durch `other(dow:sat)` formalisiert wird.

```
ANA_POINT    ::=  ana_point | that(<POINT|GENERIC>) |
                  other(<POINT|GENERIC>)
```

4.3.7 Top-Level

Die gerade eingeführten Kategorien (außer `GENERIC`) werden in der Kategorie `POINT` zusammengefaßt. Genau wie schon für L(`DURATION`) werden auch über L(`POINT`) ein `set`-Operator zur Darstellung einer Disjunktion und ein `one_out_of`-Operator zur Aufzählung mehrerer Interpretationsalternativen eingeführt. Eine Konjunktion wird als eine einfache Liste von `POINT`-Ausdrücken repräsentiert. Zum Beispiel handelt es sich bei dem Ausdruck “Mittwoch nachmittags” um eine Konjunktion der Ausdrücke “Mittwoch” und “nachmittags”. Er wird also durch `[dow:wed,pod:afternoon]` repräsentiert. Zur Negation von Aussagen über einen Zeitpunkt wird der `not`-Operator eingeführt. Zum Beispiel wird “am Ende des Monats, aber nicht am dreißigsten” als

[4] Weiter unten wird noch ein `only`-Operator eingeführt werden, der allerdings nicht der Kategorie `QUANTIFIED` zugerechnet wird. Der wesentliche Unterschied wird in Kapitel 6 klar werden. Durch Ausdrücke der Kategorie `QUANTIFIED` wird die Gesamtheit der Zeit lediglich auf verschiedene Alternativen ‘verteilt’; durch einen `only`-Ausdruck hingegen kann die Lage eines Zeitpunktes *wirklich* eingegrenzt werden.

`[end(month),not(dom:30)]` geschrieben. Um Restriktionen zu beschreiben, gibt es den zweistelligen `only`-Operator. Durch `only(pod:afternoon,dow:wed)` etwa wird der Ausdruck "am Mittwoch nur nachmittags" formalisiert. Beachte, daß hier über andere Wochentage als den Mittwoch keine Aussage gemacht wird. So wird etwa der Donnerstag Morgen nicht ausgeschlossen (durch `[dow:wed,pod:afternoon]` hingegen schon).

```
POINT        ::=  POINT* | set(POINT+) | one_out_of(POINT+) |
                  not(POINT) | only(POINT,POINT) | BASIC |
                  COMPLEX | MODIFIED | FUZZY | QUANTIFIED |
                  ANA_POINT
```

4.4 Ausdrücke zur Beschreibung eines Termins

Ein Termin hat im wesentlichen drei Merkmale: einen Beginn, ein Ende und eine Dauer. Jedes dieser Merkmale kann durch Ausdrücke der Kategorien `POINT` beziehungsweise `DURATION` beschrieben werden. Darüber hinaus kann auch durch einen einzigen `POINT`-Ausdruck eine Aussage über den Termin als Ganzes, das heißt über die Lage aller seiner Zeitpunkte gemacht werden. Im folgenden wird die TEL-Kategorie `DATE` definiert werden. Ausdrücke der Sprache L(`DATE`) werden im wesentlichen aus Listen von Ausdrücken der Kategorien `POINT` und `DURATION` bestehen, jeweils versehen mit einem Hinweis darauf, welches Merkmal durch den jeweiligen Teilausdruck beschrieben wird.

4.4.1 Zuordnung eines Namens

Jedem Ausdruck der Kategorie `DATE` kann durch den `tempex`-Operator ein eindeutiger Bezeichner zugeordnet werden. Eine derartige Zuordnung wird als ein Ausdruck der Kategorie `TEMPEX` repräsentiert. Der Bezeichner ist vom Typ `NAME`. Welches Ausdrücke dafür in Frage kommen, hängt im wesentlichen von der Realisierung der Grammatik ab. Bei einer Implementierung in Prolog bietet sich etwa die Menge der Atome an.

```
TEMPEX       ::=  tempex(NAME,DATE)
```

Ob und in welcher Weise von der Möglichkeit einer solchen Zuweisung in der Praxis Gebrauch gemacht wird, hängt natürlich von der konkreten Arbeitsweise des Dialogverarbeitungssystems ab. Gegebenenfalls könnten auch für `DURATION` und `POINT` ähnliche Konstrukte definiert werden.

4.4.2 Label

Ein Ausdruck der Kategorie `DURATION` kann sich ausschließlich auf die Dauer eines Termins beziehen. Durch das Label `for` wird eine Typkonvertierung von `DURATION` nach `DATE` angezeigt. Das bedeutet, wenn δ ein Ausdruck der Sprache L(`DURATION`) ist, so ist `for:`δ ein Ausdruck der Sprache L(`DATE`).

Für Ausdrücke der Kategorie `POINT` gibt es keine solche eindeutige Zuordnung zu einem bestimmten Merkmal eines Termins. Ein einen Zeitpunkt beschreibender Ausdruck kann sich zum Beispiel sowohl auf den Anfang eines Termins als auch auf das Ende eines solchen beziehen. Um diese Zuordnungen in der TEMPORAL EXPRESSION LANGUAGE darzustellen, werden die Label `from` und `to` eingeführt. Ist etwa $\pi \in L(\texttt{POINT})$, so ist `from:`$\pi$ ein Ausdruck der Kategorie `DATE`, der angibt, daß der Anfangspunkt des beschriebenen Termins durch den Ausdruck π eingegrenzt wird. Neben `from` und `to` wird auch noch das Label `during` definiert. Ein durch `during` eingeleiteter Ausdruck macht nicht nur eine Aussage über *einen* speziellen mit dem beschriebenen Termin assoziierten Zeitpunkt, sondern er grenzt vielmehr die gesamte Lage des Termins ein, das heißt, er bezieht sich sowohl auf dessen Beginn als auch auf sein Ende sowie auf sämtliche dazwischenliegende Zeitpunkte. Da er sich eben auch auf die dazwischenliegenden Zeitpunkte bezieht, läßt sich `during` *nicht* durch eine Kombination von `from` und `to` ersetzen.

Durch die beschriebenen Label zusammen mit Ausdrücken aus $L(\texttt{DURATION})$ beziehungsweise $L(\texttt{POINT})$ lassen sich im Prinzip schon alle zeitlichen Ausdrücke bei Terminvereinbarungsdialogen repräsentieren. Es bleibt allerdings noch eine kleine Unschönheit zu korrigieren. Ein Ausdruck wie etwa "von drei bis fünf Uhr" kann (wenigstens) zwei unterschiedliche Bedeutungen haben. Zum einen könnte durch ihn die Lage des Beginns eines Termins eingegrenzt werden ("Treffen wir uns irgendwann in der Zeit von drei bis fünf Uhr."). Eine andere Möglichkeit bestünde darin, daß durch diesen Ausdruck Anfangs- und Endpunkt eines Termins beschrieben werden ("Ich schlage ein Treffen von drei bis fünf Uhr vor."). Beide Varianten lassen sich mit den bisher eingeführten Mitteln[5] beschreiben:

- "Treffen wir uns irgendwann in der Zeit von drei bis fünf Uhr."
  `from:between(tod:3:0,tod:5:0)`
- "Ich schlage ein Treffen von drei bis fünf Uhr vor."
  `[from:tod:3:0, to:tod:5:0]`

Bei der Generierung von TEL-Ausdrücken könnte das zu Problemen führen. Es wäre sicherlich wünschenswert, zunächst einen `POINT`-Ausdruck aufbauen zu können und *danach* erst entscheiden zu müssen, welches Label zu wählen ist. Dieses Problem wird durch die Einführung eines weiteren Labels, nämlich des `interval`-Labels, gelöst. Seine Verwendung gestattet es, in Fällen wie bei obigem Beispiel prinzipiell zunächst einen Ausdruck der Kategorie `LIMITS` aufzubauen. Wird diesem dann das Label `interval` vorangestellt, so entspricht die Semantik des Gesamtausdrucks der der zweiten Variante. Im allgemeinen soll das `interval`-Label immer dann verwendet werden, wenn der nachstehende Ausdruck in seiner natürlichsprachlichen Form zwar die Struktur eines Intervalls (eines geschlossenen oder auch eines einseitig unbegrenzten) hat, sich seine Teilausdrücke aber tatsächlich direkt auf Anfangs- beziehungsweise Endpunkt des Termins beziehen. Durch dieses Label wird also quasi die Ambiguität von bestimmten Präpositionen

[5] Im zweiten Beispiel mache ich allerdings auch von der weiter unten noch zu definierenden Konjunktion von Ausdrücken der Kategorie `DATE` (durch Listenbildung) Gebrauch.

wie "von", "bis" oder "zwischen" aus der Subsprache L(`POINT`) herausgezogen und auf die nächsthöhere Ebene (das heißt L(`DATE`)) gehoben. Das löst das Problem der Auflösung einer solchen Ambiguität zwar nicht, erlaubt aber zumindest eine systematische Behandlung des Phänomens.

Im VERBMOBIL-Korpus finden sich auch einige Beispiele, bei denen sich ein zeitlicher Ausdruck nicht unmittelbar auf einen verhandelten Termin bezieht:

> "Ich sehe gerade, am zweiten Juno ist ein Feiertag."

Durch diesen Satz wird die Feststellung, daß der zweite Juni auf einen Feiertag fällt, zum Ausdruck gebracht. Ein Termin wird jedoch nicht beschrieben. Obwohl es sich bei solchen Sätzen also eigentlich um domänenfremde Ausdrücke handelt, ist es sicherlich trotzdem sinnvoll, sie ebenfalls durch TEL-Ausdrücke zu repräsentieren. Insbesondere könnten in dieser Form geäußerte zeitliche Ausdrücke schließlich auch im Nachhinein für eine Anapherresolution relevant sein, wobei ein einheitliches Format zweifelsohne von Vorteil wäre. `POINT`-Ausdrücke der Klasse der Feststellungen über die Lage von Zeitpunkten werden durch das Label `statement` eingeleitet. Im Prinzip könnten solche Statements natürlich auch über Dauern gemacht werden ("Eine Woche hat sieben Tage."). Da es hierfür aber keine Beispiele im Korpus gibt und da sie auch kaum sinnvoll wären, wird auf die Einführung eines entsprechenden Labels für Ausdrücke der Kategorie `DURATION` verzichtet.

Die aufgeführten Label zur Kennzeichnung eines Ausdrucks der Kategorie `POINT` werden in die Kategorie `POINTLABEL` gefaßt:

```
POINTLABEL   ::=  from | to | during | interval | statement
```

4.4.3 Die Kategorie DATE

Nun kann die Kategorie `DATE` definiert werden, durch die Beschreibungen von Terminen repräsentiert werden können. Grundlegende Ausdrücke der Kategorie `DATE` haben entweder die Form `for:DURATION` zur Spezifikation der Dauer eines Termins oder `POINTLABEL:POINT` zur Eingrenzung seiner Lage (beziehungsweise im Falle des Labels `statement` zur Repräsentation einer Feststellung über Zeitpunkte).

Darüber hinaus besteht auch in dieser Sprache (analog zu L(`POINT`)) die Möglichkeit der Konjunktion (durch Listenbildung), der Disjunktion (mit Hilfe des `set`-Operators), der Negation (mit Hilfe des `not`-Operators) und der Darstellung mehrerer Interpretationsalternativen (mit Hilfe des `one_out_of`-Operators). Unterscheiden sich zwei oder mehr Interpretationsalternativen lediglich in der Wahl des Labels, so kann die abkürzende Schreibweise `POINTLABEL+:POINT` benutzt werden. Jedes Label der Liste steht dabei für eine der Alternativen. Sei etwa $\pi \in L$(`POINT`). Dann sind zum Beispiel die beiden Ausdrücke `one_out_of([from:`π`,interval:`π`])` und `[from,interval]:`π semantisch äquivalent. Der Unterschied besteht darin, daß die Berechnung der Semantik im zweiten Falle effizienter gestaltet werden kann, da hier die Semantik von π nur einmal berechnet werden muß.

```
DATE        ::=  DATE* | set(DATE+) | one_out_of(DATE+) |
                 not(DATE) | for:DURATION | POINTLABEL:POINT |
                 POINTLABEL+:POINT
```

4.5 Zusammenfassung

In diesem Kapitel wurde die Grammatik der TEMPORAL EXPRESSION LANGUAGE definiert. Sie ist in Anhang B.2 nochmals abgedruckt. Abhängig von der Wahl eines Startsymbols (das heißt einer Kategorie) definiert sie eine Sprache. Von besonderem Interesse sind hierbei die Sprachen L(`DURATION`), L(`POINT`) und L(`DATE`). Ausdrücke der Kategorie `DURATION` beschreiben eine Dauer, Ausdrücke der Kategorie `POINT` beschreiben einen Zeitpunkt und Ausdrücke der Kategorie `DATE` beschreiben einen Termin als die Summe seiner Merkmale.

Wichtigstes Ziel dieser Arbeit ist, für die gerade definierten Sprachen eine formale Semantik anzugeben, die insbesondere einige einfache Inferenzen zuläßt. Vor allem die Sprache zur Beschreibung eines Zeitpunktes ist (aus semantischer Sicht) reich an redundanten Ausdrücken. Um die Definition der semantischen Interpretation zu vereinfachen, wird im nächsten Kapitel die TEMPORAL EXPRESSION LANGUAGE zunächst auf ihren sogenannten Kern reduziert werden.

Kapitel 5

Reduktion

Alles sollte so einfach wie möglich gemacht werden,
aber nicht einfacher.
Albert Einstein

In diesem Kapitel soll die TEMPORAL EXPRESSION LANGUAGE auf eine überschaubarere Sprache, im folgenden auch Kernsprache genannt, reduziert werden. Hierzu werden Transformationsregeln definiert, die Ausdrücke aus TEL auf bedeutungsäquivalente Ausdrücke des Kerns[1] abbilden. Ziel dieses Vorgehens ist es, nur für möglichst wenige TEL-Ausdrücke eine Interpretationsfunktion definieren zu müssen. Außerdem kann auf diese Weise vermieden werden, einige zwangsläufig recht willkürliche Festlegungen innerhalb der eigentlichen Semantikkomponente vornehmen zu müssen (und damit bei Bedarf nur schwer revidierbar zu machen).

Zunächst werde ich zeigen, wie Feiertage beschreibende Ausdrücke auf grundlegendere Ausdrücke reduziert werden können. Hierbei geht es unter anderem um die nicht-triviale Berechnug des Osterdatums. Im zweiten Teil wird gezeigt werden, wie sich einige modifizierte Ausdrücke in einer allgemeinen Form über Lambda-Funtionen darstellen lassen. Die übrigen Transformationsregeln für Ausdrücke der Kategorien `DURATION` und `POINT` sind im dritten Teil aufgeführt. Im vierten Teil schließlich wird die Behandung des (bekanntlich redundanten) `interval`-Labels beschrieben. Das Kapitel endet mit einer kurzen Zusammenfassung.

5.1 Feiertage und Jahreszeiten

Unter den einfachen Ausdrücken der Kategorie `POINT` sind einige, die der Beschreibung von Feiertagen oder Jahreszeiten dienen. Hierbei handelt es sich um Ausdrücke mit sehr spezieller Semantik, und insbesondere die Menge der Feiertagsausdrücke wird man sicherlich von Zeit zu Zeit anpassen müssen. Um dies

[1] Das Wort Kern deutet an, daß es sich um eine Teilmenge von TEL handelt. Dieses ist allerdings nur fast richtig: ich werde auch einige wenige neue Ausdrücke definieren und der Kernsprache zuschreiben.

Neujahr	1. Januar
Dreikönigsfest	6. Januar
Valentinstag	14. Februar
Maifeiertag	1. Mai
Mariä Himmelfahrt	15. August
Tag der Deutschen Einheit	3. Oktober
Reformationstag	31. Oktober
Allerheiligen	1. November
Nikolaustag	6. Dezember
Heilig Abend	24. Dezember
Erster Weihnachtsfeiertag	25. Dezember
Zweiter Weihnachtsfeiertag	26. Dezember
Sylvester	31. Dezember
Erntedankfest	Sonntag nach dem 29. September
Volkstrauertag	zweiter Sonntag vor dem ersten Advent
Buß- und Bettag	zweiter Mittwoch vor dem ersten Advent
Totensonntag	Sonntag vor dem ersten Advent
Erster Advent	vierter Sonntag vor dem 25. Dezember
Zweiter Advent	dritter Sonntag vor dem 25. Dezember
Dritter Advent	zweiter Sonntag vor dem 25. Dezember
Vierter Advent	Sonntag vor dem 25. Dezember

Tafel 5.1: Unbewegliche und von ihnen abhängige Feiertage

zu vereinfachen, soll die Interpretation solcher Ausdrücke nicht im Rahmen der eigentlichen Semantikkomponente definiert werden. Stattdessen kann ihre Bedeutung durch andere, weniger spezielle Ausdrücke umschrieben werden. Diese Reduktion wird durch sogenannte Transformationsregeln angegeben werden.

5.1.1 Unbewegliche und von ihnen abhängige Feiertage

Einige Feiertage, wie zum Beispiel die Weihnachtsfeiertage, Sylvester oder der Tag der Deutschen Einheit werden jedes Jahr am gleichen Datum begangen. Daneben gibt es eine Reihe von Feiertagen, deren Lage unmittelbar von diesen unbeweglichen Feiertagen abhängt. Hierzu gehören unter anderem die Adventstage und der Buß- und Bettag. Tafel 5.1 gibt einen Überblick.

Am einfachsten lassen sich Transformationsregeln für die unbeweglichen Feiertage angeben. Die jeweilige Regel bildet das den betreffenden Feiertag bezeichnende Akronym auf eine zweielementige Liste aus Tages- und Monatsangabe ab. Beispielsweise fällt der Neujahrstag jedes Jahr auf den ersten Januar. Die entsprechende Transformationsregel hat also folgende Gestalt:

```
holiday:new_year_s_day ⟼ [dom:1,month:jan]
```

Die Regeln für die übrigen unbeweglichen Feiertage werden im analoger Weise formuliert:

```
holiday:epiphany           ⟼ [dom:6,month:jan]
holiday:st_valentine_s_day ⟼ [dom:14,month:2]
holiday:may_day            ⟼ [dom:1,month:may]
holiday:assumption_day     ⟼ [dom:15,month:aug]
holiday:german_unity_day   ⟼ [dom:3,month:oct]
holiday:reformation_day    ⟼ [dom:31,month:oct]
holiday:all_saints_day     ⟼ [dom:1,month:nov]
holiday:st_nicholas_day    ⟼ [dom:6,month:dec]
holiday:christmas_eve      ⟼ [dom:24,month:dec]
holiday:christmas_day      ⟼ [dom:25,month:dec]
holiday:boxing_day         ⟼ [dom:26,month:dec]
holiday:new_year_s_eve     ⟼ [dom:31,month:dec]
```

Die von den unbeweglichen Feiertagen unmittelbar abhängigen Feiertage lassen sich in der TEMPORAL EXPRESSION LANGUAGE allesamt durch Ausdrücke der Kategorie `RELATED` umschreiben. Zum Beispiel fällt das Erntedankfest stets auf den Sonntag unmittelbar nach dem 29. September. In Form einer Transformationsregel läßt sich dieser Sachverhalt folgendermaßen ausdrücken:

```
holiday:harvest_festival ⟼ the_after(dow:sun,[dom:29,month:sep])
```

Die Transformationsregeln für die verbleibenden Feiertage von Tafel 5.1 sind im folgenden aufgeführt. Beachte, daß hierbei auch auf der rechten Seite einer Transformationsregel ein einen Feiertag bezeichnender Ausdruck auftreten kann. Dies ist zum Beispiel bei der Definition der Regel für die Lage des Volkstrauertages der Fall; `holiday:national_day_of_mourning` wird zunächst in einen Ausdruck, der den Teilausdruck `holiday:first_advent` enthält, umgeformt. Erst eine weitere Anwendung einer Transformationsregel wandelt diesen Ausdruck in einen Ausdruck der Kernsprache um. Dabei ist natürlich darauf zu achten, daß keine Zyklen entstehen.

```
holiday:national_day_of_mourning ⟼ the_before(2,dow:sun,
                                    holiday:first_advent)
holiday:day_of_prayer_and_repentance ⟼ the_before(2,dow:wed,
                                        holiday:first_advent)
holiday:last_sunday_before_advent ⟼ the_before(dow:sun,
                                     holiday:first_advent)
holiday:first_advent  ⟼ the_before(4,dow:sun,[dom:25,month:dec])
holiday:second_advent ⟼ the_before(3,dow:sun,[dom:25,month:dec])
holiday:third_advent  ⟼ the_before(2,dow:sun,[dom:25,month:dec])
holiday:fourth_advent ⟼ the_before(dow:sun,[dom:25,month:dec])
```

5.1.2 Ostern und davon abhängige Feiertage

Die Lage aller weiteren berücksichtigten Feiertage ist abhängig von der Lage des Ostersonntages im jeweiligen Jahr. Ostern wird am ersten Sonntag nach dem

DIE GAUSS'SCHE OSTERREGEL

Eingabe: $J > 1582$
Ausgabe: $g(J)$

Im Jahre J liegt Ostern $g(J)$ Tage nach dem 21. März.

Quelle: [Hemme, 1991]

$$
\begin{array}{rcl}
a & := & J \bmod 19 \\
b & := & J \bmod 4 \\
c & := & J \bmod 7 \\
m & := & (8 \cdot (J \operatorname{div} 100) + 13) \operatorname{div} 25 - 2 \\
s & := & (J \operatorname{div} 100) - (J \operatorname{div} 400) - 2 \\
M & := & (15 + s - m) \bmod 30 \\
N & := & (6 + s) \bmod 7 \\
d & := & (M + 19 \cdot a) \bmod 30 \\
D & := & \begin{cases} 28 & \text{falls } d = 29 \\ 27 & \text{falls } d = 28 \wedge a \geq 11 \\ d & \text{sonst} \end{cases} \\
e & := & (2 \cdot b + 4 \cdot c + 6 \cdot D + N) \bmod 7 \\
g(J) & := & D + e + 1
\end{array}
$$

Tafel 5.2: Algorithmus zur Berechnung des Ostersonntags

1990	15. April	1997	30. März	2004	11. April
1991	31. März	1998	12. April	2005	27. März
1992	19. April	1999	4. April	2006	16. April
1993	11. April	2000	23. April	2007	8. April
1994	3. April	2001	15. April	2008	23. März
1995	16. April	2002	31. März	2009	12. April
1996	7. April	2003	20. April	2010	4. April

Tafel 5.3: Die Ostersonntage der Jahre 1990 bis 2010

ersten Vollmond im Frühling gefeiert. Das bekannteste Verfahren zur Berechnung des Osterdatums ist die Gauß'sche Osterregel. Tafel 5.2 zitiert sie nach [Hemme, 1991]. Ein ähnliches Verfahren wird in [Knuth, 1962] vorgestellt. Der Ostersonntag liegt stets zwischen dem 22. März und dem 25. April. Mit Hilfe der Osterregel kann in Abhängigkeit von einer Jahreszahl berechnet werden, um wieviele Tage Ostern nach dem 21. März des jeweiligen Jahres liegt. Die Regel in der in Tafel 5.2 angegebenen Form ist ab dem Jahre 1583, also seit der Gregorianischen Kalenderreform, gültig. Sie wurde benutzt, um die in Tafel 5.3 zusammengestellten Ostertermine der Jahre 1990 bis 2010 zu berechnen.

Um die Semantik eines TEL-Ausdrucks, der sich auf den Ostersonntag bezieht, zu bestimmen, könnte die Gauß'sche Osterregel zwar benutzt werden, durch eine Transformationsregel läßt sie sich jedoch nicht implementieren. Da im VERBMOBIL-Szenario nur eine begrenzte Menge von Jahreszahlen eine Rolle spielt, genügt es, für diese Jahre die Osterdaten in Form einer Tabelle anzugeben. Dieses Vorgehen hat den Vorteil, daß die Angabe einer einfachen Transformati-

Rosenmontag	– 48 Tage	Ostersonntag	± 0 Tage
Faschingsdienstag	– 47 Tage	Ostermontag	+ 1 Tag
Aschermittwoch	– 46 Tage	Christi Himmelfahrt	+ 39 Tage
Gründonnerstag	– 3 Tage	Pfingstsonntag	+ 49 Tage
Karfreitag	– 2 Tage	Pfingstmontag	+ 50 Tage
Karsamstag	– 1 Tag	Fronleichnam	+ 60 Tage

Tafel 5.4: Feiertage in Relation zum Ostersonntag

onsregel ausreicht. Die folgende Regel bildet `holiday:easter_sunday` auf die Disjunktion der Osterdaten der Jahre 1990 bis 2010 ab. Es ist offensichtlich, daß eine Erweiterung beziehungsweise Einschränkung bei Bedarf jederzeit problemlos möglich ist.

```
holiday:easter_sunday  ⟼  set([[dom:15,month:apr,year:1990], ...
                           ... [dom:4,month:apr,year:2010]])
```

Tafel 5.4 gibt an, wie eine Reihe weiterer Feiertage in Abhängigkeit vom Ostersonntag bestimmt werden. Zum Beispiel ergibt sich der Rosenmontag stets als der 48. Tag vor Ostern. In der Temporal Expression Language entspricht dies einer Verschiebung, die entsprechende Transformationsregel lautet also:

```
holiday:monday_before_lent  ⟼  neg_shift(dur(48,days),
                                 holiday:easter_sunday)
```

Es folgen die Transformationsregeln für die übrigen Feiertage aus Tafel 5.4:

```
holiday:shrove_tuesday    ⟼  neg_shift(dur(47,days),
                              holiday:easter_sunday)
holiday:ash_wednesday     ⟼  neg_shift(dur(46,days),
                              holiday:easter_sunday)
holiday:maundy_thursday   ⟼  neg_shift(dur(3,days),
                              holiday:easter_sunday)
holiday:good_friday       ⟼  neg_shift(dur(2,days),
                              holiday:easter_sunday)
holiday:holy_saturday     ⟼  neg_shift(dur(1,days),
                              holiday:easter_sunday)
holiday:easter_monday     ⟼  pos_shift(dur(1,days),
                              holiday:easter_sunday)
holiday:ascension_day     ⟼  pos_shift(dur(39,days),
                              holiday:easter_sunday)
holiday:whit_sunday       ⟼  pos_shift(dur(49,days),
                              holiday:easter_sunday)
holiday:whit_monday       ⟼  pos_shift(dur(50,days),
                              holiday:easter_sunday)
holiday:corpus_christi    ⟼  pos_shift(dur(60,days),
                              holiday:easter_sunday)
```

Muttertag fällt normalerweise auf den zweiten Sonntag im Mai. Nur wenn dies der Pfingstsonntag ist, dann wird Muttertag auf den ersten Sonntag im Mai vorverlegt. Auch dies läßt sich durch die Angabe einer Transformationsregel beschreiben. Die gerade gemachte Aussage über die Lage des Muttertages ist nämlich äquivalent mit der folgenden: Muttertag ist der letzte Sonntag des Zeitraumes, der vom ersten bis zum zweiten Sonntag im Mai dauert, jedoch den Pfingstsonntag nicht enthält. Die entsprechende Transformationsregel lautet:

```
holiday:mother_s_day  ⟼ last_of(dow:sun,
    [min_between(of(1,dow:sun,month:may),of(2,dow:sun,month:may)),
    not(holiday:whit_sunday)])
```

5.1.3 Zeiträume um Feiertage

Neben den Feiertagsusdrücken, die sich jeweils auf genau einen Tag beziehen, wurden im letzten Kapitel auch einige Ausdrücke zur Beschreibung von längeren Zeiträumen, deren Lage von einem Feiertag abhängig ist, eingeführt. Es sind dies `carnival`, `holy_week`, `easter`, `whitsun`, `advent` und `christmas`.

Die nächste Transformationsregel definiert die Karnevalszeit als den Zeitraum vom Donnerstag vor dem Faschingsdienstag bis zum Faschingsdienstag selbst:

```
holiday:carnival  ⟼  min_between(
                       the_before(dow:thu,holiday:shrove_tuesday),
                       holiday:shrove_tuesday)
```

Die Karwoche ist die Woche um den Karfreitag:

```
holiday:holy_week  ⟼  the_around(week,holiday:good_friday)
```

Das Akronym `holiday:easter` soll die Zeit vom Karsamstag bis zum Ostermontag umschreiben:

```
holiday:easter  ⟼  min_between(holiday:holy_saturday,
                     holiday:easter_monday)
```

Mit `holiday:whitsun` werde die Zeit vom Samstag vor dem Pfingstmontag bis zum Pfingstmontag selbst bezeichnet:

```
holiday:whitsun  ⟼  min_between(
                      the_before(dow:sat,holiday:whit_monday),
                      holiday:whit_monday)
```

Die Adventszeit beginnt am ersten Advent und endet am 24. Dezember:

```
holiday:advent  ⟼  min_between(holiday:first_advent,
                     [dom:24,month:dec])
```

Der Ausdruck "Weihnachten" wird meist für die Zeit vom 24. bis zum 26. Dezember gebraucht:

```
holiday:christmas  ⟼  min_between([dom:24,month:dec],
                        [dom:26,month:dec])
```

JAHRESZEITEN AUF DER NORDHALBKUGEL			
Frühling:	vom 21. März	bis zum 22. Juni	= 92,8 Tage
Sommer:	vom 22. Juni	bis zum 23. September	= 93,6 Tage
Herbst:	vom 23. September	bis zum 22. Dezember	= 89,8 Tage
Winter:	vom 22. Dezember	bis zum 21. März	= 89,0 Tage

Tafel 5.5: Ungefähre Lage und Dauer der Jahreszeiten nach [Moore, 1989]

Für alle Ausdrücke der Form `holiday:HOLIDAY` konnte eine Transformationsregel angegeben werden, das heißt, sie alle gehören nicht zur Kernsprache, ihre Semantik muß also nicht mehr explizit definiert werden. Es bleibt noch der Ausdruck `holiday`, durch den der natürlichsprachliche Ausdruck "feiertags" formalisiert wird. Da ein Zeitpunkt genau dann an *irgendeinem* Feiertag liegt, wenn er entweder am ersten Advent, oder am zweiten Advent, oder am ... liegt, kann für diesen Ausdruck also unter Verwendung des `set`-Operators ebenfalls eine Transformationsregel angegeben werden:

```
holiday  ⟼  set([holiday:first_advent,holiday:second_advent, ...
              ... holiday:whit_sunday])
```

5.1.4 Jahreszeiten

Entgegen einem weit verbreiteten Irrglauben ist Frühlingsanfang nicht immer am 21. März, vielmehr sind Dauer und Beginn beziehungsweise Ende der Jahreszeiten säkularen Schwankungen unterworfen. Ihre ungefähre Lage wird jedoch in Tafel 5.5 beschrieben.

Da Jahreszeiten bei der Beschreibung von Terminen aber ohnehin nur eine untergeordnete Rolle spielen, habe ich darauf verzichtet, diese Schwankungen bei der Festlegung geeigneter Transformationsregeln zu berücksichtigen. Selbstverständlich könnte dies jederzeit (etwa analog zur Behandlung des Ostersonntags) nachgeholt werden.

```
season:spring ⟼ min_between([dom:21,month:mar],[dom:21,month:jun])
season:summer ⟼ min_between([dom:22,month:jun],[dom:22,month:sep])
season:fall   ⟼ min_between([dom:23,month:sep],[dom:21,month:dec])
season:winter ⟼ min_between([dom:22,month:dec],[dom:20,month:mar])
```

5.2 Lambda-Funktionen

Es bietet sich an, Ausdrücke der Kategorie `MODIFIED` möglichst außerhalb des Kerns von TEL zu halten, da ihre Interpretation (in einigen Fällen jedenfalls) kaum allgemeingültig festgelegt werden kann beziehungsweise sollte. Gelingt es aber, die speziellen Formen der Modifikation mit Transformationsregeln zu kodieren, so können diese bei Bedarf jederzeit problemlos geändert werden. Ohne allzuweit vorgreifen zu müssen, kann sicherlich davon ausgegangen werden, daß bei der Formulierung einer Semantik für TEL Intervalle eine wesentliche Rolle spielen werden. Bei der Interpretation modifizierter Ausdrücke wird jedes Intervall der Interpretation des übergebenen Ausdrucks geeignet zu modifizieren sein. Für firsthalf(dow:sat) heißt das zum Beispiel, daß für alle Intervalle aus der Interpretation von dow:sat (das sind alle Samstage) die Menge der ersten Hälften dieser Intervalle die Interpretation des modifizierten Ausdrucks ergibt. Allgemein lassen sich solche Modifikationen von Intervallen als Lambda-Funktionen formulieren. Im übrigen lassen sich unscharfe Ausdrücke (also solche der Kategorie FUZZY) in gleicher Weise behandeln.

5.2.1 Die Struktur von Lambda-Funktionen

Ich ändere für die Kernsprache also die Definition des Typs MODIFIED. An die Stelle der bisherigen speziellen Modifikationen tritt ein allgemeiner Ausdruck mit dem Funktor modified, dessen erstes Element eine (geeignet kodierte) Lambda-Funktion und dessen zweites Element der zu modifizierende Ausdruck ist:

```
MODIFIED     ::=  modified(LAMBDA,<POINT|GENERIC>)
```

Nun muß noch die Grammatik für Lambda-Funktionen definiert werden. Natürlich müssen nicht beliebige Lambda-Funktionen beschrieben werden können, es genügt eine recht einfache Klasse dieser Funktionen zu berücksichtigen. Solch eine Lambda-Funktion hat zwei Argumente, die linke und die rechte Intervallgrenze. Als Ergebnis liefert sie ebenfalls zwei Intervallgrenzen. Diese können entweder mit einem der übergebenen Argumente identisch sein (left beziehungsweise right) oder aber sich durch die Addition oder Subtraktion eines gewissen Bruchteils der Länge des Intervalls zu beziehungsweise von einem der Werte left oder right ergeben. Dieser Bruchteil wird durch einen Ausdruck der Kategorie FRACTION repräsentiert.

```
LAMBDA       ::=  lambda(LAMBDA_ARG,LAMBDA_ARG)

LAMBDA_ARG   ::=  left | right | plus(<left|right>,FRACTION) |
                  minus(<left|right>,FRACTION)
```

Per definitionem soll bei Bedarf für plus stets abgerundet und für minus stets aufgerundet werden.

5.2.2 Modifizierte Ausdrücke (MODIFIED)

Nun können Transformationsregeln für die Ausdrücke der Kategorie MODIFIED angegeben werden. Zu einem gegebenen MODIFIED-Ausdruck der Gesamtsprache liefern sie einen MODIFIED-Ausdruck der Kernsprache.

Sei $\pi \in L(\texttt{POINT}) \cup L(\texttt{GENERIC})$; es wird festgelegt:

$\texttt{early}(\pi) \longmapsto \texttt{modified(lambda(left,plus(left,1:4)),}\pi\texttt{)}$
$\texttt{late}(\pi) \longmapsto \texttt{modified(lambda(minus(right,1:4),right),}\pi\texttt{)}$
$\texttt{begin}(\pi) \longmapsto \texttt{modified(lambda(left,plus(left,1:3)),}\pi\texttt{)}$

middle(π) $\longmapsto$ modified(lambda(plus(left,1:3),minus(right,1:3)),π)
end(π) $\longmapsto$ modified(lambda(minus(right,1:3),right),π)
firsthalf(π) $\longmapsto$ modified(lambda(left,plus(left,1:2)),π)
secondhalf(π) $\longmapsto$ modified(lambda(minus(right,1:2),right),π)

Durch `early` wird also beispielsweise das erste Viertel eines Zeitraumes 'ausgeschnitten'. Diese Festlegungen sind wie gesagt einigermaßen willkürlich und wurden genau deswegen auch außerhalb der eigentlichen Semantik gemacht.

5.2.3 Unscharfe Ausdrücke (FUZZY)

Zur Interpretation eines Ausdrucks der Kategorie `FUZZY` scheint die Ausdehnung aller Intervalle in beide Richtungen um jeweils die Länge des Intervalls angemessen. Dann würde zum Beispiel "ungefähr Donnerstag" wie "Mittwoch bis (darauffolgenden) Freitag" interpretiert. Nur wenn das übergebene Intervall überhaupt kein Intervall im eigentlichen Sinne ist, wenn es sich nämlich um einen einzelnen Zeitpunkt handelt, so sollte das (entartete) Intervall doch wenigstens um einen Mindestbetrag gestreckt werden, etwa eine Stunde in jede Richtung. Diese grundsätzliche Verlängerung kann eigentlich prinzipiell vorgenommen werden, in den anderen Fällen ändert sie das Ergebnis nur geringfügig, was, da es sich ohnehin um unscharfe Angaben handelt, nicht weiter von Bedeutung ist, dafür aber eine uniforme Darstellung erlaubt.

Sei $\pi \in L(\texttt{POINT})$. Durch `min_between` wird jedes Intervall in jede Richtung um eine halbe Stunde verlängert. Durch die Lambda-Funktion wird das so entstandene Intervall dann wiederum in beide Richtungen um den Betrag seiner eigenen Länge ausgedehnt.

fuzzy(π) $\longmapsto$ modified(lambda(minus(left,1),plus(right,1)),
min_between(neg_shift(dur(1:2,hours),π),
pos_shift(dur(1:2,hours),π)))

5.2.4 Einseitig unbegrenzte Intervalle (LIMIT)

Zwar werde ich im nächsten Kapitel eine Semantik für Ausdrücke der Kategorie `LIMIT`, also für Intervalle mit nur einem Randpunkt angeben, ihre Berechnung wird jedoch in vielen Fällen wenig nützlich sein. Auf der einen Seite stellt "vor dem sechsten Oktober achtundneunzig" natürlich eine sinnvolle Einschränkung eines Zeitpunktes dar und ebendies wird durch die noch anzugebende Semantik auch reflektiert werden; auf der anderen Seite trifft es aber wohl auf jeden nur denkbaren Zeitpunkt zu, daß er "nach dem Wochenende" liegt. Selbst Zeitpunkte, die selbst an einem Wochende liegen, liegen wenigstens nach dem vorherigen Wochenende. Natürlich intendiert eine Aussage wie "nach dem Wochende", daß der gesuchte Zeitpunkt nicht nur nach einem Wochenende, sondern sicherlich auch in einer gewissen Nähe zu eben diesem Wochenende liegen sollte. Einen derartigen Gedanken in einer formalen Semantik auszudrücken, dürfte allerdings schwierig werden. Das Problem wird nicht vollständig gelöst werden können, jedoch lassen sich mit Hilfe von Lambda-Funktionen immerhin gewisse Heuristiken in Form von Transformationsregeln formulieren.

Eine mögliche Transformationsregel für `ex_after` ist etwa die folgende. Hierbei sei $\pi \in L(\texttt{POINT})$.

```
ex_after(π)   ⟼  modified(lambda(right,plus(right,6)),
                 min_between(neg_shift(dur(1,hours),π),π))
```

Wird durch π etwa eine Uhrzeit spezifiziert, so beschreibt der transformierte Ausdruck den sechsstündigen Zeitraum nach dieser Uhrzeit. Beschreibt das Argument einen bestimmten Tag, so beschreibt der neue Ausdruck (in etwa) die sechs Tage nach diesem Tag. Wird der Tag zum Beispiel als Tag eines Monats angegeben, so ist dies in den meisten Fällen sicherlich ein sinnvolles Ergebnis. Wird jedoch nur ein Wochentag angegeben, so erfährt die Menge der möglichen Zeitpunkte durch den Ausdruck kaum eine Einschränkung. Das gilt erst recht für obiges Beispiel "nach dem Wochenende". Allgemein läßt sich sagen, daß die Heuristik umso besser greift, je spezifischer der Ausdruck π ist.

Entsprechende Regeln für `in_after`, `ex_before` und `in_before` werden im folgenden gegeben (sei wieder $\pi \in L(\texttt{POINT})$).

```
in_after(π)    ⟼  modified(lambda(left,plus(right,6)),
                  min_between(π,pos_shift(dur(1,hours),π)))
ex_before(π)   ⟼  modified(lambda(minus(left,6),left),
                  min_between(π,pos_shift(dur(1,hours),π)))
in_before(π)   ⟼  modified(lambda(minus(left,6),right),
                  min_between(neg_shift(dur(1,hours),π),π))
```

Eine mögliche Erweiterung des Systems bestünde darin, für unterschiedliche Klassen von Ausdrücken an der Stelle von π unterschiedliche Heuristiken, also unterschiedliche Transformationsregeln, aufzustellen. Dem sollte allerdings eine gründliche Evaluierung des in dieser Arbeit beschriebenen Systems (ohne beziehungsweise mit den vorgestellten simplen Heuristiken) vorangehen.

Transformationsregeln für `before` und `after` werden im nächsten Teil dieses Kapitels angegeben werden (Abschnitt 5.3.4).

5.2.5 Unscharfe Dauerangaben (FUZZY_DUR)

Um Unschärfe in Bezug auf eine Dauerangabe darzustellen, genügt die Angabe zweier rationaler Zahlen. Multipliziert man sie mit der ursprünglichen Dauer, so erhält man eine untere beziehungsweise eine obere Grenze für die Semantik des unscharfen Ausdrucks. Im Prinzip könnte auch dies als die Anwendung einer Lambda-Funktion aufgefaßt werden. Es genügt aber wie gesagt die Angabe der zwei Zahlen. Sie werden als Ausdrücke der Kategorie `FRACTION` repräsentiert. Es wird die Kategorie `MODIFIED_DUR` eingeführt, um auf die gerade beschriebene Art modifizierte `DURATION`-Ausdrücke darzustellen.

```
MODIFIED_DUR ::=  modified_dur(FRACTION,FRACTION,DURATION)
```

Ein Ausdruck der Kategorie `FUZZY_DUR` kann nun als Instanz eines `MODIFIED_DUR`-Ausdrucks angegeben werden. Als Zahl zur Berechnung der unteren Grenze wähle ich $\frac{1}{2}$, für die obere Grenze $1\frac{1}{2}$. Dies geschieht durch Anwendung der folgenden Transformaionsregel.

Sei $\delta \in L(\texttt{DURATION})$. Es wird festgelegt:

$\texttt{fuzzy\_dur}(\delta) \longmapsto \texttt{modified\_dur(1:2,1:1:2,}\delta\texttt{)}$

5.3 Weitere Transformationsregeln

5.3.1 Ausdrücke der Kategorie `DURATION`

Unter den verbleibenden Ausdrücken der Kategorie `DURATION` können nur noch die der Form `dur(several,UNIT)` reduziert werden. Ich lege fest, daß `several` für ein Zahl zwischen 2 und 5 stehen soll. Dementsprechend wird ein solcher Ausdrück in einen `RANGE`-Ausdruck transformiert.

Für $\omega \in L(\texttt{UNIT})$ wird festgelegt:

$\texttt{dur(several,}\omega\texttt{)} \longmapsto \texttt{range(dur(2,}\omega\texttt{),dur(5,}\omega\texttt{))}$

5.3.2 Ausdrücke der Kategorie `POINT`: Top-Level

Auch für den Fall, daß `several` als Quantor gebraucht wird, identifiziere ich es mit den Zahlen zwischen 2 und 5. Die Transformationsregel bildet hier auf eine Disjunktion ab, in deren Elemente `several` durch 2, 3, 4 beziehungsweise 5 ersetzt wurde.

Sei $\pi \in L(\texttt{POINT}) \cup L(\texttt{GENERIC})$; es gilt:

$$\begin{aligned}\texttt{quantified(several,}\pi\texttt{)} \longmapsto\ & \texttt{set([quantified(2,}\pi\texttt{),quantified(3,}\pi\texttt{),}\\ & \texttt{quantified(4,}\pi\texttt{),quantified(5,}\pi\texttt{)])}\end{aligned}$$

Ausdrücke, bei denen `several` negiert wird, interpretiere ich wie Ausdrücke mit Quantor `1`, das heißt, für $\pi \in L(\texttt{POINT}) \cup L(\texttt{GENERIC})$ wird festgelegt:

$\texttt{quantified(not(several),}\pi\texttt{)} \longmapsto \texttt{quantified(1,}\pi\texttt{)}$

Durch die Ausdrücke "montags im Mai" und "nur montags im Mai" wird gleichermaßen beschrieben, daß der bezeichnete Zeitpunkt an einem Montag und im Mai liegt. Eine etwas andere Bedeutung hat jedoch "im Mai nur montags". Hierdurch wird ausgedrückt, daß im Mai zwar nur Montage in Frage kommen; Zeitpunkte, die nicht im Mai liegen, werden aber nicht weiter eingeschränkt. Es wird natürlich schwierig sein, solche Bedeutungsnuancen bei der Generierung eines TEL-Ausdrucks zu erkennen. Wenn es jedoch möglich ist, so kann etwa "im Mai nur montags" durch `only(dow:mon,month:may)` repräsentiert werden.

Den Ausdruck $\texttt{only(}\pi_1\texttt{,}\pi_2\texttt{)}$ könnte man auch als "wenn π_2, dann auch π_1" lesen. Beachte folgende Äquivalenz:

$$P_2 \rightarrow P_1 \iff (P_1 \wedge P_2) \vee \neg P_2$$

Seien also $\pi_1, \pi_2 \in L($POINT$)$; dann definiert folgende Transformationsregel die Semantik des `only`-Operators:

only(π_1,π_2) $\longmapsto$ set([[π_1,π_2],not(π_2)])

5.3.3 Einfache Ausdrücke der Kategorie POINT

Die TEL-Ausdrücke zur Beschreibung einer Tageszeit (außer `pod:am` und `pod:pm`) können auf einfache Intervalle reduziert werden. Zum Beispiel soll `pod:afternoon` für den Zeitraum zwischen 14.00 Uhr und 18.00 Uhr eines Tages stehen. Durch die Verwendung von `min_between` (statt `between`) wird sichergestellt, daß die beiden angegebenen Randpunkte am selben Tag liegen (beziehungsweise im Falle von `pod:night` an aufeinanderfolgenden Tagen).

```
pod:morning      ⟼ min_between([tod:6:0,pod:am],[tod:0:0,pod:pm])
pod:morning_ger1 ⟼ min_between([tod:6:0,pod:am],[tod:10:0,pod:am])
pod:morning_ger2 ⟼ min_between([tod:10:0,pod:am],[tod:0:0,pod:pm])
pod:midday       ⟼ min_between([tod:11:0,pod:am],[tod:2:0,pod:pm])
pod:afternoon    ⟼ min_between([tod:2:0,pod:pm],[tod:6:0,pod:pm])
pod:evening      ⟼ min_between([tod:5:0,pod:pm],[tod:11:0,pod:pm])
pod:night        ⟼ min_between([tod:10:0,pod:pm],[tod:6:0,pod:am])
pod:daytime      ⟼ min_between([tod:6:0,pod:am],[tod:10:0,pod:pm])
```

Wochentage können auch durch ihre Lage innerhalb einer Woche umschrieben werden, indem sie in `ORDINAL`-Ausdrücke transformiert werden.

```
dow:mon ⟼ of(1,day,week)
dow:tue ⟼ of(2,day,week)
dow:wed ⟼ of(3,day,week)
dow:thu ⟼ of(4,day,week)
dow:fri ⟼ of(5,day,week)
dow:sat ⟼ of(6,day,week)
dow:sun ⟼ of(7,day,week)
```

Arbeitstage sind die Tage von Montag bis Freitag. Also kann `pow:workday` durch eine Disjunktion dieser fünf Wochentage ersetzt werden. Als Wochenende bezeiche ich den Zeitraum von Freitag bis Sonntag. Der Freitag wird mitaufgenommen, da wenigstens der Freitag Abend oder auch Nachmittag oft als zum Wochenende gehörig angesehen wird.

```
pow:workday ⟼ set([dow:mon,dow:tue,dow:wed,dow:thu,dow:fri])
pow:weekend ⟼ min_between(dow:fri,dow:sun)
```

Die Bezeichner `today`, `tomorrow`, `yesterday` sind offensichtlich redundant, da sie durch entsprechende Ausdrücke mit `deictic_day` ersetzt werden können.

```
today     ⟼ deictic_day(0)
tomorrow  ⟼ deictic_day(1)
yesterday ⟼ deictic_day(-1)
```

Ausdrücke der Form `deictic_day(INTEGER)` selbst können wiederum auf Ausdrücke mit dem allgemeineren Funktor `deictic` abbgebildet werden. Für $z \in L($INTEGER$)$ gilt:

deictic_day(z) $\longmapsto$ deictic(z,day)

Auch die Redundanz von Ausdrücken der Form dom:DOM und dergleichen wurde im letzten Kapitel bereits angesprochen. Sie können durch Ausdrücke der Kategorie ORDINAL umschrieben werden.

Sei $n \in L$(NUMBER):

dom:n $\longmapsto$ of(n,day,month)
woy:n $\longmapsto$ of(n,week,year)
qoy:n $\longmapsto$ of(n,quarter_year,year)
hoy:n $\longmapsto$ of(n,half_year,year)

Genauso können Monate statt durch Angabe ihres Namens durch ihre Position innerhalb eines Jahres spezifiziert werden.

month:jan $\longmapsto$ of(1,month,year)
month:feb $\longmapsto$ of(2,month,year)
month:mar $\longmapsto$ of(3,month,year)
month:apr $\longmapsto$ of(4,month,year)
month:may $\longmapsto$ of(5,month,year)
month:jun $\longmapsto$ of(6,month,year)
month:jul $\longmapsto$ of(7,month,year)
month:aug $\longmapsto$ of(8,month,year)
month:sep $\longmapsto$ of(9,month,year)
month:oct $\longmapsto$ of(10,month,year)
month:nov $\longmapsto$ of(11,month,year)
month:dec $\longmapsto$ of(12,month,year)

5.3.4 Komplexe Ausdrücke der Kategorie POINT

Kann für ein Intervall mit nur einem Randpunkt nicht eindeutig bestimmt werden, ob der angegebene Randpunkt selbst noch mit zu dem beschriebenen Zeitraum gehört, so werden die Funktoren before und after benutzt. Durch die folgenden Transformationsregeln wird ein solcher Ausdruck in eine Menge von zwei Interpretationsalternativen übersetzt, eine für die 'inklusiv'- und eine für die 'exklusiv'-Variante.

Für $\pi \in L$(POINT) wird festgelegt:

before(π) $\longmapsto$ one_out_of([in_before(π),ex_before(π)])
after(π) $\longmapsto$ one_out_of([in_after(π),ex_after(π)])

Ausdrücke der Form in(DURATION) und ago(DURATION) sind Verschiebungen, bei denen das zweite Argument, der Refernzzeitpunkt, nicht explizit mitangegeben wird. Er ist jedoch stets mit dem aktuellen Zeitpunkt gleichzusetzen, das heißt für $\delta \in L$(DURATION) können die folgenden Regeln angewandt werden:

in(δ) $\longmapsto$ pos_shift(δ,now)
ago(δ) $\longmapsto$ neg_shift(δ,now)

Um Ausdrücke wie "die Woche vom ersten bis zum sechsten" darstellen zu können, gibt es den Funktor week_between. Die letzte Äußerung könnte auch durch "die Woche um den Zeitraum vom ersten bis zum sechsten" umschrieben werden. Die entsprechende Transformationsregel hat (für $\pi_1, \pi_2 \in L$(`POINT`)) die folgende Form:

`week_between(`π_1`,`π_2`)` $\longmapsto$ `the_around(week,min_between(`π_1`,`π_2`))`

Die Funktoren `the_before` und `the_after` gibt es in drei- und zweistelliger Ausführung. In letzterem Falle unterbleibt die Angabe einer Zahl; sie ist mit 1 gleichzusetzen. Zum Beispiel ist "der Dienstag nach Ostern" identisch mit "dem ersten Dienstag nach Ostern".

Für $\gamma \in L$(`COUNTABLE`) und $\pi \in L$(`POINT`) gilt:

`the_before(`γ`,`π`)` $\longmapsto$ `the_before(1,`γ`,`π`)`
`the_after(`γ`,`π`)` $\longmapsto$ `the_after(1,`γ`,`π`)`

Statt "der Montag zwei Wochen nach Allerheiligen" könnte man auch "der Montag um den Zeitraum zwei Wochen nach Allerheiligen" sagen. Entsprechend sind die Transformationsregeln für `dow_before` und `dow_after` aufgebaut. Werden diese Funktoren zweistellig gebraucht, so ist das fehlende dritte Argument mit dem aktuellen Zeitpunkt (also `now`) gleichzusetzen.

Es seien $\gamma \in L$(`dow : DOW`), $\delta \in L$(`DURATION`) und $\pi \in L$(`POINT`):

`dow_before(`γ`,`δ`,`π`)` $\longmapsto$ `the_around(`γ`,neg_shift(`δ`,`π`))`
`dow_after(`γ`,`δ`,`π`)` $\longmapsto$ `the_around(`γ`,pos_shift(`δ`,`π`))`
`dow_before(`γ`,`δ`)` $\longmapsto$ `dow_before(`γ`,`δ`,now)`
`dow_after(`γ`,`δ`)` $\longmapsto$ `dow_after(`γ`,`δ`,now)`

Wird `last_of` zweistellig gebraucht, so ist an erster Stelle eine 1 einzufügen. Für $\gamma \in L$(`COUNTABLE`) und $\pi \in L$(`POINT`) $\cup\, L$(`GENERIC`) gilt also:

`last_of(`γ`,`π`)` $\longmapsto$ `last_of(1,`γ`,`π`)`

Die deiktischen Bezeichner `this`, `next` und `last` können durch den allgemeinen Funktor `deictic` und eine ganze Zahl ersetzt werden. Für $\gamma \in L$(`COUNTABLE`) gilt:

`this(`γ`)` $\longmapsto$ `deictic(0,`γ`)`
`next(`γ`)` $\longmapsto$ `deictic(1,`γ`)`
`last(`γ`)` $\longmapsto$ `deictic(-1,`γ`)`

Auch Ausdrücke der Form `deictic(INTEGER,COUNTABLE)` selbst können noch reduziert werden. Die folgenden Transformationsregeln formen sie in geeignete Ausdrücke der Kategorie `RELATED` um, wobei das letzte Argument `now` ist.

Sei $n \in L$(`NUMBER`):

`deictic(0,`γ`)` $\longmapsto$ `the_around(`γ`,now)`
`deictic(`n`,`γ`)` $\longmapsto$ `the_after(`n`,`γ`,now)`
`deictic(`$-n$`,`γ`)` $\longmapsto$ `the_before(`n`,`γ`,now)`

5.4 Behandlung des `interval`-Labels

Bei der Einführung des Labels `interval` im letzten Kapitel wurde bereits darauf hingewiesen, daß es sich hierbei um eine redundante Konstruktion handelt. Die Transformation zu einem bedeutungsäquivalenten Ausdruck ohne `interval` ist allerdings weitaus komplexer als die bisher aufgestellten Regeln.

In den mesiten Fällen hat das `interval`-Label die gleiche Bedeutung wie `during`. Nur wenn der folgende `POINT`-Ausdruck eine Intervallbeschreibung enthält, so sollen die Randpunkte dieses Intervalls direkt mit den Randpunkten des Termins identifiziert werden. Beispielsweise besagt `during:[dow:mon,min_between(tod:6:0,tod:8:0)]`, daß das gesamte Treffen an einem Montag in der Zeit zwischen sechs und acht Uhr liegen soll, ein Treffen von halb sieben bis sieben würde diese Forderung zum Beispiel erfüllen. Verwendet man aber `interval` statt `during`, so macht der entsprechende Ausdruck die Aussage, daß Treffen solle (wie bisher) an einem Montag stattfinden, um sechs Uhr beginnen und um acht Uhr enden.

Um alle `interval`-Label aus einem Ausdruck zu eliminieren, sind im einzelnen die im folgenden beschriebenen Schritte zu unternehmen. Hierbei seien $\pi, \pi_1, \ldots, \pi_n \in L(\texttt{POINT})$ und $\ell_1, \ldots, \ell_m \in L(\texttt{POINTLABEL})$.

Taucht `interval` in einer Label-Liste auf, so muß es zur weiteren Bearbeitung aus dieser herausgetrennt werden:

$$[\ell_1, \ldots, \texttt{interval}, \ldots, \ell_m]\texttt{:}\pi \longmapsto \texttt{one\_out\_of([interval:}\pi\texttt{,[}\ell_1, \ldots, \ell_m\texttt{]:}\pi\texttt{])}$$

Bezieht sich `interval` auf eine Konjunktion, eine Disjunktion oder eine Liste von Interpretationsalternativen, so muß das Label in die entsprechende Liste 'hineingezogen' werden:

$$\texttt{interval:[}\pi_1, \ldots, \pi_n\texttt{]} \longmapsto \texttt{[interval:}\pi_1\texttt{,...,interval:}\pi_n\texttt{]}$$

$$\texttt{interval:set([}\pi_1, \ldots, \pi_n\texttt{])} \longmapsto \texttt{set([interval:}\pi_1\texttt{,...,interval:}\pi_n\texttt{])}$$

$$\texttt{interval:one\_out\_of([}\pi_1, \ldots, \pi_n\texttt{])} \longmapsto \texttt{one\_out\_of([interval:}\pi_1\texttt{,...,interval:}\pi_n\texttt{])}$$

Entsprechendes gilt auch für den `not`-Operator, er wird vor das Label gezogen:

$$\texttt{interval:not(}\pi\texttt{)} \longmapsto \texttt{not(interval:}\pi\texttt{)}$$

Für den `only`-Operator ist das nicht möglich, da er über $L(\texttt{DATE})$ nicht definiert wurde. Die in Abschnitt 5.3.1 definierte Transformation muß also gleichzeitig mit einem Nach-Innen-Ziehen von `interval` durchgeführt werden:

$$\texttt{interval:only(}\pi_1, \pi_2\texttt{)} \longmapsto \texttt{set([[interval:}\pi_1\texttt{,interval:}\pi_2\texttt{],not(interval:}\pi_2\texttt{)])}$$

Nun geht es um die `POINT`-Ausdrücke, die tatsächlich eine Veränderung erfahren, wenn sie durch das `interval`-Label eingeleitet werden. Im Falle von `between` erhält das erste Argument das Label `from`, das zweite das Label `to`:

$$\texttt{interval:between(}\pi_1\texttt{,}\pi_2\texttt{)} \longmapsto \texttt{[from:}\pi_1\texttt{,to:}\pi_2\texttt{]}$$

Bei `min_between` ist die Situation entsprechend. Hier wird aber zusätzlich ausgedrückt, daß die beiden bezeichneten Randpunkte einen minimalen Abstand haben. Dies wird im Zielausdruck durch den entsprechenden `during`-Ausdruck sichergestellt:

$$\texttt{interval:min\_between}(\pi_1,\pi_2) \longmapsto \begin{array}{l}[\texttt{from:}\pi_1\texttt{,to:}\pi_2\texttt{,} \\ \texttt{during:min\_between}(\pi_1,\pi_2)]\end{array}$$

Die 'inklusiv'-Varianten der einseitig unbegrenzten Intervalle werden so interpretiert, daß sich das Argument unmittelbar auf Anfangs- beziehungsweise Endpunkt des Termins bezieht. Auf die 'exklusiv'-Varianten hat `interval` keinen besonderen Einfluß. Zum Beispiel ist die Semantik von "nach fünf Uhr" (exklusiv) klar. "Ab fünf Uhr" (inklusiv) hingegen könnte entweder besagen, daß der gesamte Termin in der Zeit ab fünf Uhr liegt, oder aber daß er genau um fünf Uhr beginnt. Die Transformation von `before` und `after` wird dementsprechend festgelegt:

$$\texttt{interval:in\_before}(\pi) \longmapsto \texttt{to:}\pi$$

$$\texttt{interval:in\_after}(\pi) \longmapsto \texttt{from:}\pi$$

$$\texttt{interval:before}(\pi) \longmapsto \texttt{one\_out\_of([to:}\pi\texttt{,during:ex\_before(}\pi\texttt{)])}$$

$$\texttt{interval:after}(\pi) \longmapsto \texttt{one\_out\_of([from:}\pi\texttt{,during:ex\_after(}\pi\texttt{)])}$$

In allen verbleibenden Fällen soll `interval` einfach durch `during` ersetzt werden:

$$\texttt{interval:}\pi \longmapsto \texttt{during:}\pi$$

Es ist wichtig, daß die Transformationsregeln 'von außen nach innen' angewandt werden, das heißt, erst nachdem durch die in diesem Teil vorgestellten Regeln jedes Vorkommen von `interval` aus einem Ausdruck eliminiert wurde, dürfen die Transformationsregeln für Ausdrücke der Kategorie `POINT` benutzt werden. Andernfalls könnte es passieren, daß zum Beispiel `pod:morning` zunächst in einen Ausdruck mit `min_between` umgewandelt würde und, wenn dann noch ein `interval`-Label davorstünde, wäre das Ergebnis der Transformation nicht mehr bedeutungsäquivalent mit der Eingabe.

5.5 Zusammenfassung

In diesem Kapitel wurde beschrieben, wie Ausdrücke der TEMPORAL EXPRESSION LANGUAGE durch die sukzessive Anwendung von Transformationsregeln auf bedeutungsäquivalente Ausdrücke der sogenannten Kernsprache abgebildet werden können. Die Grammatik der Kernsprache ist in Anhang B.3 angegeben. In den meisten Fällen konnten durch diese Reduktion bestimmte TEL-Kategorien vereinfacht oder auch ganz gestrichen werden. Nur in zwei Fällen wurde eine neue Kategorie hinzugefügt beziehungsweise die ursprüngliche erweitert. Es sind dies `MODIFIED_DUR` und `MODIFIED`. Durch letztere werden die durch Lambda-Funktionen modifizierten Ausdrücke zur Beschreibung der Lage eines Zeitpunktes repräsentiert.

Die Reduktions-Komponente ist so konzipiert, daß Änderungen mit relativ geringem Aufwand möglich sind. Eine denkbare Erweiterung des Systems besteht darin, Heuristiken über die Bedeutung von TEL-Ausdrücken in Form von Transformationsregeln zu integrieren. Ein Beispiel hierfür ist der in Abschnitt 5.2.4 vorgestellte Ansatz, die durch einseitig unbegrenzte Intervalle beschriebenen Zeiträume sinnvoll einzugrenzen.

Im folgenden Kapitel wird für die Kernsprache eine Semantik definiert werden. Implizit wird dadurch natürlich auch die Semantik der gesamten Temporal Expression Language beschrieben.

Kapitel 6

Semantik

O, wer weiß,
Was in der Zeiten Hintergrunde schlummert?
Friedrich Schiller: Don Karlos, Infant von Spanien.

Die Formulierung einer Semantik für die TEMPORAL EXPRESSION LANGUAGE erfordert im wesentlichen zweierlei. Erstens muß ein geeigneter semantischer Bereich festgelegt werden, und zweitens muß eine Interpretationsfunktion von den TEL-Ausdrücken auf diesen Bereich definiert werden. Beides sind Themen dieses Kapitels.

Im ersten Teil werde ich einige grundsätzliche Eigenschaften des Kalenders und damit auch des zu definierenden semantischen Bereichs beschreiben. Letztlich interessiert, wie Ausdrücke aus $L(\texttt{DATE})$ zu interpretieren sind; auf dem Wege dorthin müssen natürlich vor allem die Sprachen $L(\texttt{DURATION})$ und $L(\texttt{POINT})$ betrachtet werden. Nach einigen allgemeinen Bemerkungen zur Semantik von TEL wird je ein Teil des Kapitels der Interpretation einer der drei Subsprachen gewidmet sein. Das Kapitel schließt mit einer kurzen Zusammenfassung der wichtigsten Ergebnisse.

6.1 Eine Kalendersemantik

Bei Terminvereinbarungsdialogen werden sämtliche Aussagen über Zeitpunkte in Bezug auf einen festen Kalender gemacht. Im Falle der für VERBMOBIL gesammelten Dialoge ist dies der Gregorianische Kalender.[1] Der Umstand, daß dieser Kalender erst seit dem 15.10.1582 gültig ist und daß er seine Gültigkeit möglicherweise irgendwann einmal verlieren wird, darf und wird ignoriert werden. Auch werde ich ohne weitere Erwähnung – wie schon bei der Definition der Syntax von TEL – sich durch den Wechsel von Sommer- und Winterzeit ergebende Verschiebungen nicht berücksichtigen.

In diesem Abschnitt sollen zunächst einmal losgelöst von der TEL-Syntax einige Eigenschaften unseres Kalenders diskutiert und in Definitionen niedergelegt werden.

[1] Angaben zu Geschichte und Struktur des Gregorianischen Kalenders finden sich beispielsweise in [Bien, 1988].

6.1.1 Schaltjahre

Zur Umkreisung der Sonne benötigt die Erde etwas mehr als 365 Tage. Um diese Ungenauigkeit auszugleichen, gibt es Schaltjahre (im Englischen: *leap year*). Normalerweise ist jedes vierte Jahr ein Schaltjahr. Dies wäre aber immer noch nicht genau genug. Daher fällt alle 100 Jahre das Schaltjahr aus beziehungsweise findet alle 400 Jahre doch wieder statt.

Definition 6.1 *Die Funktion* $\mathrm{leap} : \mathbb{Z} \to \{\top, \bot\}$ *wird definiert wie folgt:*

$$\mathrm{leap} : y \longmapsto (y \bmod 4 = 0) \wedge ((y \bmod 100 = 0) \to (y \bmod 400 = 0))$$

Es ist also für $y \in \mathbb{Z}$ genau dann $\mathrm{leap}(y) = \top$, wenn das Jahr y im Gregorianischen Kalender ein Schaltjahr ist.

6.1.2 Zeitpunkte im Kalender

Jedem Zeitpunkt entspricht ein Datum sowie eine Uhrzeit. Für Uhrzeiten ist die Angabe von Stunde und Minute ausreichend. Also können Zeitpunkte (eindeutig) als Quintupel aus Jahr, Monat, Tag, Stunde und Minute dargestellt werden. Die Menge der möglichen Jahre ist hierbei die Menge der ganzen Zahlen, wenngleich in der Praxis ausschließlich Zeitpunkte im zwanzigsten beziehungsweise einundzwanzigsten Jahrhundert zu betrachten sein werden. Für die übrigen vier Angaben kommen nur Zahlen zwischen 0 und 59 in Frage. Dabei gilt es zu beachten, daß lediglich Schaltjahre einen 29. Februar haben, einen 35. Mai hingegen gibt es nie und ähnliches mehr. Welches genau die Menge der gültigen Zeitpunkte im Kalender ist, wird durch folgende Definition formalisiert.

Definition 6.2 *Die Menge* $\mathcal{C} \subset \mathbb{Z}^5$ *heißt die Menge der Zeitpunkte im Kalender und wird wie folgt definiert:*

$$\begin{aligned}\mathcal{C} \;:=\; \{(y,m,d,h,i) \in \mathbb{Z}^5 \mid h \in \{0,\ldots,23\} \;\wedge\; i \in \{0,\ldots,59\} \wedge \\ ((m \in \{1,3,5,7,8,10,12\} \wedge d \in \{1,\ldots,31\}) \vee \\ (m \in \{4,6,9,11\} \wedge d \in \{1,\ldots,30\}) \vee \\ (m = 2 \wedge d \in \{1,\ldots,28\}) \vee \\ (\mathrm{leap}(y) \wedge m = 2 \wedge d = 29))\}\end{aligned}$$

Darüber hinaus werden noch $\infty_{\mathcal{C}} := (\infty,\infty,\infty,\infty,\infty)$ *und* $-\infty_{\mathcal{C}} := (-\infty,-\infty,-\infty,-\infty,-\infty)$ *der Menge* $\mathcal{C}$ *zugeschrieben.*[2]

Die Menge $\mathcal{C}$ stellt also einen semantischen Bereich für den Gregorianischen Kalender dar. Später wird es notwendig werden, aus einem gegebenen Zeitpunkt beispielsweise die Jahres- oder die Monatsangabe zu extrahieren. Hierfür werden fünf Hilfsfunktionen definiert, von denen jede für $t \in \mathcal{C}$ den Wert einer bestimmten Stelle des Quintupels t liefert.

[2] Ich werde im folgenden darauf verzichten, jede einzelne Definition explizit auf $\infty_{\mathcal{C}}$ und $-\infty_{\mathcal{C}}$ auszudehnen (es handelt sich um triviale und für die Praxis unerhebliche Sonderfälle).

Definition 6.3 *Für $(y, m, d, h, i) \in \mathcal{C}$ wird definiert:*

$$\begin{array}{rcl}
\mathrm{year}((y, m, d, h, i)) & := & y \\
\mathrm{month}((y, m, d, h, i)) & := & m \\
\mathrm{day}((y, m, d, h, i)) & := & d \\
\mathrm{hour}((y, m, d, h, i)) & := & h \\
\mathrm{minute}((y, m, d, h, i)) & := & i
\end{array}$$

Für zwei (nicht identische) Zeitpunkte läßt sich stets bestimmen, ob der erste zeitlich vor dem zweiten liegt. Auf $\mathcal{C}$ soll nun eine Ordnung erklärt werden, die diese zeitliche Relation ausdrückt.

Definition 6.4 *Seien $(y_1, m_1, d_1, h_1, i_1), (y_2, m_2, d_2, h_2, i_2) \in \mathcal{C}$ beliebig. Dann wird auf der Menge $\mathcal{C}$ die Relation $\leq_{\mathcal{C}}$ erklärt durch*

$$\begin{array}{rcl}
(y_1, m_1, d_1, h_1, i_1) \leq_{\mathcal{C}} (y_2, m_2, d_2, h_2, i_2) & :\Longleftrightarrow & y_1 \leq y_2 \wedge \\
& & (y_1 = y_2 \rightarrow (m_1 \leq m_2 \wedge \\
& & (m_1 = m_2 \rightarrow (d_1 \leq d_2 \wedge \\
& & (d_1 = d_2 \rightarrow (h_1 \leq h_2 \wedge \\
& & (h_1 = h_2 \rightarrow i_1 \leq i_2))))))
\end{array}$$

Wie leicht nachgeprüft werden kann, ist $\leq_{\mathcal{C}}$ reflexiv, antisymmetrisch und transitiv. Außerdem gilt für alle $t_1, t_2 \in \mathcal{C}$ stets entweder $t_1 \leq_{\mathcal{C}} t_2$ oder $t_2 \leq_{\mathcal{C}} t_1$. Also ist $\leq_{\mathcal{C}}$ eine *Totalordnung* beziehungsweise $(\mathcal{C}, \leq_{\mathcal{C}})$ ist eine *total geordnete Menge.*[3]

Das bedeutet unter anderem, daß jede endliche Teilmege T von $\mathcal{C}$ sowohl ein Maximum als auch ein Minimum hat.

Definition 6.5 *Sei $T \subset \mathcal{C}$ mit $|T| < \infty$. Dann heißen* $\min_{\mathcal{C}}(T)$ *das* Minimum *und* $\max_{\mathcal{C}}(T)$ *das* Maximum *von T. Sie werden definiert als diejenigen Elemente von T, für die gilt:*

$$\begin{array}{rcl}
\min_{\mathcal{C}}(T) = t \in T & :\Longleftrightarrow & \forall t' \in T : t \leq_{\mathcal{C}} t' \\
\max_{\mathcal{C}}(T) = t \in T & :\Longleftrightarrow & \forall t' \in T : t' \leq_{\mathcal{C}} t
\end{array}$$

Auf der totalgeordneten Menge $\mathcal{C}$ kann auf die übliche Weise die Menge der geschlossenen Intervalle über $\mathcal{C}$ definiert werden. (Beachte, daß dies für eine Menge ohne assoziierte Totalordnung *nicht* möglich wäre.) Außerdem werden durch die folgende Definition sogenannte einseitig unbegrenzte Intervalle eingeführt werden. So lassen sich die Menge aller Zeitpunkte ab einem bestimmten Referenzzeitpunkt sowie die Menge aller Zeitpunkte bis zu einem Referenzzeitpunkt umschreiben.

Definition 6.6 *Über $\mathcal{C}$ werden vier Arten von Intervallen definiert:*

[3] Zur Erläuterung der algebraischen Begriffe siehe [Lipson, 1981].

(i) Seien $t_1, t_2 \in \mathcal{C}$ mit $t_1 \leq_{\mathcal{C}} t_2$. Dann heißt die Menge

$$[t_1, t_2] := \{t \in \mathcal{C} \mid t_1 \leq_{\mathcal{C}} t \leq_{\mathcal{C}} t_2\}$$

das geschlossene Intervall *von t_1 bis t_2.*

(ii) Sei $t \in \mathcal{C}$. Dann heißt die Menge

$$[t, \infty_{\mathcal{C}}] := \{t' \in \mathcal{C} \mid t \leq_{\mathcal{C}} t'\}$$

das rechtsseitig unbegrenzte Intervall *ab t.*

(iii) Sei wieder $t \in \mathcal{C}$. Dann heißt die Menge

$$[-\infty_{\mathcal{C}}, t] := \{t' \in \mathcal{C} \mid t' \leq_{\mathcal{C}} t\}$$

das linksseitig unbegrenzte Intervall *bis t.*

(iv) Auf analoge Weise kann auch die gesamte Menge $\mathcal{C}$ notiert werden. Es heißt

$$[-\infty_{\mathcal{C}}, \infty_{\mathcal{C}}] := \mathcal{C}$$

dann auch das beidseitig unbegrenzte Intervall.

Die Vereinigung der Mengen der geschlossenen und der diversen unbegrenzten Intervalle über $\mathcal{C}$ heißt die Menge der Intervalle über $\mathcal{C}$ *und wird $Int(\mathcal{C})$ geschrieben.*

Da die Zeitpunkte in $\mathcal{C}$ diskret sind, impliziert das Vorhandensein einer totalen Ordnung auch, daß für alle Elemente aus $\mathcal{C}$ sowohl ein Nachfolger als auch ein Vorgänger existiert. Sie werden nachfolgend konstruktiv definiert.

Definition 6.7 *Auf der Menge $\mathcal{C}$ werden die einstelligen Operatoren* succ *und* pred *definiert.*

(i) Für alle $t = (y, m, d, h, i) \in \mathcal{C}$ heißt succ(t) *der* Nachfolger *von t mit*

$$\mathrm{succ}(t) := \begin{cases} (y+1, 1, 1, 0, 0) & \text{für } i = 59 \wedge h = 23 \wedge d = 31 \wedge m = 12 \\ (y, m+1, 1, 0, 0) & \text{für } i = 59 \wedge h = 23 \wedge \\ & \quad ((d = 31 \wedge m \in \{1,3,5,7,8,10\}) \vee \\ & \quad\; (d = 30 \wedge m \in \{4,6,9,11\}) \vee \\ & \quad\; (d = 28 \wedge m = 2 \wedge \neg \mathrm{leap}(y)) \vee \\ & \quad\; (d = 29 \wedge m = 2 \wedge \mathrm{leap}(y))) \\ (y, m, d+1, 0, 0) & \text{sonst für } i = 59 \wedge h = 23 \\ (y, m, d, h+1, 0) & \text{sonst für } i = 59 \\ (y, m, d, h, i+1) & \text{sonst} \end{cases}$$

(ii) *Für alle* $t = (y, m, d, h, i) \in \mathcal{C}$ *heißt* $\mathrm{pred}(t)$ *der* Vorgänger *von* t *mit*

$$\mathrm{pred}(t) := \begin{cases} (y-1, 12, 31, 23, 59) & \text{für } i = 0 \wedge h = 0 \wedge d = 1 \wedge m = 1 \\ (y, m-1, 31, 23, 59) & \text{für } i = 0 \wedge h = 0 \wedge d = 1 \wedge \\ & \quad m \in \{2, 4, 6, 8, 9, 11\} \\ (y, m-1, 30, 23, 59) & \text{für } i = 0 \wedge h = 0 \wedge d = 1 \wedge \\ & \quad m \in \{5, 7, 10, 12\} \\ (y, 2, 28, 23, 59) & \text{für } i = 0 \wedge h = 0 \wedge d = 1 \wedge \\ & \quad m = 3 \wedge \neg\mathrm{leap}(y) \\ (y, 2, 29, 23, 59) & \text{für } i = 0 \wedge h = 0 \wedge d = 1 \wedge \\ & \quad m = 3 \wedge \mathrm{leap}(y) \\ (y, m, d-1, 23, 59) & \text{sonst für } i = 0 \wedge h = 0 \\ (y, m, d, h-1, 59) & \text{sonst für } i = 0 \\ (y, m, d, h, i-1) & \text{sonst} \end{cases}$$

Es ist anschaulich klar, daß es sich tatsächlich um eine Nachfolger- beziehungsweise Vorgängerrelation im herkömmlichen Sinne handelt. Für alle $t \in \mathcal{C}$ ist $\mathrm{succ}(t)$ ebenfalls in $\mathcal{C}$ und zwar "später" als t. Entsprechend ist auch $\mathrm{pred}(t) \in \mathcal{C}$ für alle $t \in \mathcal{C}$ und zwar "vor" t. Des weiteren gibt es keine Elemente zwische t und seinem Nachfolger $\mathrm{succ}(t)$ und auch keine Elemente zwischen dem Vorgänger $\mathrm{pred}(t)$ und t. Vielmehr gilt:

$$\forall t \in \mathcal{C} : (t \leq_{\mathcal{C}} \mathrm{succ}(t) \wedge (\neg\exists t' \in \mathcal{C} \setminus \{t, \mathrm{succ}(t)\} : t \leq_{\mathcal{C}} t' \leq_{\mathcal{C}} \mathrm{succ}(t)))$$

Für die Vorgängerrelation gilt entsprechend:

$$\forall t \in \mathcal{C} : (\mathrm{pred}(t) \leq_{\mathcal{C}} t \wedge (\neg\exists t' \in \mathcal{C} \setminus \{\mathrm{pred}(t), t\} : \mathrm{pred}(t) \leq_{\mathcal{C}} t' \leq_{\mathcal{C}} t))$$

6.1.3 Verschieben von Zeitpunkten um Minuten

Die Anwendung des succ-Operators auf ein Element $t \in \mathcal{C}$ bedeutet anschaulich das Verschieben des im Kalender definierten Zeitpunktes t um eine Minute in die Zukunft. Es handelt sich also um eine Art Addition. Diese kann von bisher einer Minute auf eine beliebige Anzahl von Minuten erweitert werden, indem der succ-Operator entsprechend oft angewandt wird. Das heißt, $\mathrm{succ}^{(n)}(t)$ bezeichnet den Zeitpunkt, der genau n Minuten nach Zeitpunkt t liegt.

Definition 6.8 *Um einen Zeitpunkt im Kalender um eine Zahl von Minuten in die Zukunft oder die Vergangenheit zu verschieben, werden eine Addition und eine Subtraktion erklärt.*

(i) *Die Addition einer Zahl von Minuten* $n \in \mathbb{N}_0$ *zu einem Zeitpunkt* $t \in \mathcal{C}$ *wird über die* n*-fache Anwendung des Nachfolge-Operators* succ *definiert:*

$$t \oplus_{min} n := \mathrm{succ}^{(n)}(t)$$

(ii) *Ebenfalls für* $t \in \mathcal{C}$ *und* $n \in \mathbb{N}_0$ *wird analog eine Subtraktion folgendermaßen definiert:*

$$t \ominus_{min} n := \mathrm{pred}^{(n)}(t)$$

Also bezeichnet zum Beipiel $(1999, 12, 31, 23, 55) \oplus_{min} 5$ den Zeitpunkt fünf Minuten nach 23 Uhr 55 am Abend des 31. Dezembers 1999.

6.1.4 Monate und Minuten

Beziehungen zwischen Zeitpunkten lassen sich mitunter durch Zeiteinheiten, also Jahre, Monate, Wochen, Tage, Stunden oder Minuten, ausdrücken. Ein wichtiges Phänomen des Kalenders im Zusammenhang mit auf ihm erklärten Zeiteinheiten ist, daß ein Monat keine konstante Zahl von Tagen hat. Im letzten Abschnitt wurde die Minute quasi als Grundeinheit des Kalenders eingeführt, was vielleicht suggerieren könnte, die anderen Einheiten ließen sich durch sie eindeutig darstellen. Das stimmt aber nur bedingt. Eine Stunde hat 60 Minuten, ein Tag 1440 und eine Woche hat 10080 Minuten. Auch hat der Monat Januar stets 31 Tage und damit 44640 Minuten. Das Jahr 1998 hat 365 Tage, also 525600 Minuten. Es wäre aber falsch zu sagen, *ein* Jahr habe 365 Tage, da dies für manche Jahre, nämlich für die Schaltjahre, nicht wahr ist. Entsprechendes gilt natürlich auch für Monate. Hier sind die Unterschiede, beispielsweise zwischen Januar und Februar, noch größer.

Dennoch, die allgemeinen Aussagen über das Verhältnis von Minuten zu Stunden, von Minuten zu Tagen und von Minuten zu Wochen sind wahr, auch losgelöst von konkreten Zeitpunkten im Kalender. Ebenso ist es wahr, daß ein Jahr stets zwölf Monate hat. Auch dieses gilt unabhängig von der konkreten Lage dieser Zeitspanne innerhalb des Kalenders. Somit gibt es zwei Grundeinheiten bei der Messung von Längen von Zeitdauern. Eine *exakte* Umrechnung zwischen den beiden ist im allgemeinen Falle *nicht* möglich, sondern nur im Kontext einer konkreten Angabe von Zeitpunkten, etwa der Anfänge der zu vergleichenden Dauern.

Werden Dauerangaben allerdings als Bruchteile von Monaten gemacht, wie zum Beispiel in "zweieinhalb Monate", so scheint eine etwas ungenaue Umrechnug in Tage auch losgelöst vom Kontext eines bestimmten Zeitpunktes zulässig, ja sogar unvermeidlich. Mit "einen Monat nach dem zehnten Februar" wird sicherlich immer der zehnte März gemeint sein, übrigens unabhängig davon, ob es sich um ein Schaltjahr handelt oder nicht. Hingegen ist die Semantik von "anderthalb Monate nach dem zehnten Februar" unklar. Verschiebt man zunächst um einen Monat und sodann um einen weiteren halben, so würde man den halben Monat in Bezug auf den Monat März in Tage umrechnen. Das heißt man käme auf den Tag 15 oder 16 Tage nach dem zehnten März, also auf den 25. oder den 26. März. Zählte man aber zuerst den halben Monat ab – im Februar –, so ergäben sich (für Nicht-Schaltjahre) 14 Tage. Ein Monat nach dem Tag, der 14 Tage nach dem zehnten Februar liegt, ist der 24. März.

Da es sich bei der Angabe eines Bruchteils eines Monats also kaum um eine exakt zu verstehende Angabe handeln kann, scheint eine sofortige Umrechnung in Tage (beziehungsweise in Minuten) zulässig. Hierbei werde ich allgemein eine Länge von 30 Tagen für einen Monat annehmen. Im folgenden genügt also die Betrachtung ganzzahliger Monatsbeträge.

6.1.5 Verschieben von Zeitpunkten um Monate

Aufgrund des Fehlens eines konstanten Zahlenverhältnisses der Dauern von einer Minute und einem Monat kann die weiter oben vorgestellte Addition nicht unmittelbar angewandt werden, um einen Zeitpunkt um eine bestimmte Anzahl von Monaten zu verschieben. Im folgenden sollen also eine Addition sowie eine Subtraktion für Monate definiert werden. Ich beschränke mich hierbei auf die Addition (beziehungsweise Subtraktion) einer ganzen Zahl von Monaten.

Die Semantik eines Ausdrucks wie "drei Monate nach dem 15. Mai" scheint klar zu sein. Es ist der 15. August gemeint. Es muß also nur der neue Monat berechnet werden, gegebenenfalls auch das neue Jahr, der Tag innerhalb des Monats bleibt aber der gleiche. Was aber ist mit dem Tag "einen Monat nach dem 31. Mai" gemeint? Einen 31. Juni gibt es nicht. Bei der Verschiebung aller übrigen Tage im Mai um einen Monat in die Zukunft würde man jeweils 31 Tage addieren. Im Falle des 31. Mai ergäbe ein solches Vorgehen den ersten Juli. Dieses Ergebnis entspricht nach meiner Meinung nicht der intuitiven Semantik eines solchen Ausdrucks. Vielmehr sollten zwei sich im Abstand von einem Monat befindliche Tage in unmittelbar aufeinanderfolgenden Monaten liegen, im Beispiel also im Mai beziehungsweise Juni. Daher setze ich für die Semantik von "ein Monat nach dem 31. Mai" den 30. Juni fest. Entsprechendes gilt auch allgemein, das heißt bei Überschreiten des Monatsendes wird stets der letzte Tag des Monats genommen.

Beachte, daß eine derartige Addition sich nicht wie die Addition von Minuten über einen Nachfolge-Operator definieren läßt. Das liegt daran, daß die Addition von Monaten zu einem Zeitpunkt im Kalender nicht assoziativ ist. Zum Beispiel ist der Tag zwei Monate nach dem 31. Mai der 31. Juli, wohingegen der Tag, der einen Monat nach dem Tag einen Monat nach dem 31. Mai liegt, der 30. Juli ist. Im zweiten Fall wurde zunächst ein Monat addiert; es ergibt sich der 30. Juni. Bei Addition eines weiteren Monats erhält man dann also den 30. Juli.

Definition 6.9 *Für einen Zeitpunkt $t = (y, m, d, h, i) \in \mathcal{C}$ werden die Verschiebung um eine Anzahl von Monaten $n \in \mathbb{N}_0$ in positiver beziehungsweise negativer Richtung, also eine Addition und eine Subtraktion, definiert.*

(i) Seien $y^+ = y + (m + n - 1) \text{ div } 12$ *und* $m^+ = (m + n - 1) \bmod 12 + 1$.

$$t \oplus_{mon} n := \begin{cases} (y^+, m^+, d, h, i) & \text{für } m^+ \in \{1,3,5,7,8,10,12\} \vee \\ & \quad (d \leq 30 \wedge m^+ \in \{4,6,9,11\}) \vee \\ & \quad (d = 29 \wedge m^+ = 2 \wedge \mathrm{leap}(y^+)) \vee \\ & \quad (d \leq 28 \wedge m^+ = 2) \\ (y^+, m^+, 30, h, i) & \text{für } d = 31 \wedge m^+ \in \{4,6,9,11\} \\ (y^+, m^+, 29, h, i) & \text{für } d \in \{30,31\} \wedge m^+ = 2 \wedge \mathrm{leap}(y^+) \\ (y^+, m^+, 28, h, i) & \text{für } d \in \{29,30,31\} \wedge m^+ = 2 \wedge \neg\mathrm{leap}(y^+) \end{cases}$$

(ii) Seien $y^- = y - (12 - m + n) \text{ div } 12$ *sowie* $m^- = (m - n - 1) \bmod 12 + 1$.

$$t \ominus_{mon} n := \begin{cases} (y^-, m^-, d, h, i) & \text{für } m^- \in \{1,3,5,7,8,10,12\} \vee \\ & \quad (d \leq 30 \wedge m^- \in \{4,6,9,11\}) \vee \\ & \quad (d = 29 \wedge m^- = 2 \wedge \mathrm{leap}(y^-)) \vee \\ & \quad (d \leq 28 \wedge m^- = 2) \\ (y^-, m^-, 30, h, i) & \text{für } d = 31 \wedge m^- \in \{4,6,9,11\} \\ (y^-, m^-, 29, h, i) & \text{für } d \in \{30,31\} \wedge m^- = 2 \wedge \mathrm{leap}(y^-) \\ (y^-, m^-, 28, h, i) & \text{für } d \in \{29,30,31\} \wedge m^- = 2 \wedge \neg\mathrm{leap}(y^-) \end{cases}$$

Es ist zum Beispiel $(1998, 5, 12, 14, 0) \ominus_{mon} 6 = (1997, 11, 12, 14, 0)$, in Worten: der Zeitpunkt sechs Monate vor dem zwölften Mai 1998, 14 Uhr, ist der zwölfte November 1997, ebenfalls 14 Uhr.

6.1.6 Verknüpfung von Zeitpunkten und Dauerangaben

Dem Umstand Rechnung tragend, daß sich Minuten und Monate nicht generell ineinander umrechen lassen, werden Dauern als Paare von einer Zahl von Monaten und einer Zahl von Minuten dargestellt. Beides sind natürliche Zahlen (einschließlich 0).

Definition 6.10 *Die Menge* $\mathcal{D} = \mathbb{N}_0 \times \mathbb{N}_0$ *heißt die Menge der* Monats-/Minuten-Paare.

Die in Definition 6.8 und 6.9 eingeführten Verknüpfungen sollen nun kombiniert werden zu einer Addition beziehungsweise Subtraktion für Zeitpunkte und Monats-/Minuten-Paare.

Definition 6.11 *Sei $t \in \mathcal{C}$. Des weiteren sei $(n_1, n_2) \in \mathcal{D}$. Es steht also n_1 für eine Anzahl von Monaten und n_2 für eine Anzahl von Minuten. Zur Addition und Subtraktion werden die Infix-Operatoren $\oplus$ und $\ominus$ definiert. Beide bilden $\mathcal{C} \times \mathcal{D}$ ab auf die Menge $\mathcal{C}$.*

(i) Die Addition wird definiert wie folgt:

$$t \oplus (n_1, n_2) \;:=\; (t \oplus_{mon} n_1) \oplus_{min} n_2$$

(ii) Die Subtraktion wird analog definiert:

$$t \ominus (n_1, n_2) \;:=\; (t \ominus_{mon} n_1) \ominus_{min} n_2$$

Die Addition der Monate erfolgt also stets vor der Addition der Minuten. Dieses Vorgehen entspricht der intuitiven Semantik wohl am besten. Eine Dauer dargestellt als Paar von Monaten und Minuten kann nun also sowohl zu einem Zeitpunkt im Kalender addiert als auch von ihm subtrahiert werden.

6.1.7 Der Abstand zweier Zeitpunkte

Zu zwei gegebenen Zeitpunkten im Kalender kann ihr zeitlicher Abstand in Minuten berechnet werden. Die Definition dieses Abstandsbegriffs ist dual zur Definition der Addition von Minuten zu Zeitpunkten.

Definition 6.12 *Seien $t_1, t_2 \in \mathcal{C}$. Dann heißt $\langle t_1, t_2 \rangle_{\mathcal{C}}$ der* Abstand *der Zeitpunkte t_1 und t_2 in Minuten.*

(i) Für $t_1 \leq_{\mathcal{C}} t_2$ wird er definiert durch folgende Äquivalenz:

$$\langle t_1, t_2 \rangle_{\mathcal{C}} = n \;:\Longleftrightarrow\; \mathrm{succ}^{(n)}(t_1) = t_2$$

(ii) Für $t_2 \leq_{\mathcal{C}} t_1$ wird definiert:

$$\langle t_1, t_2 \rangle_{\mathcal{C}} := -\langle t_2, t_1 \rangle_{\mathcal{C}}$$

Beispielsweise ist der Abstand von 22 Uhr zu 23 Uhr am ersten Januar 1990 genau 60 Minuten: $\langle (1990, 1, 1, 22, 0), (1990, 1, 1, 23, 0) \rangle_{\mathcal{C}} = 60$.

Nachdem nun der Abstand zweier Zeitpunkte definiert wurde, soll auch noch ein Abstandsbegriff für Zeitpunkt-Intervalle eingeführt werden. Hierzu werde ich zunächst definieren, welches der Mittelwert zweier Elemente aus $\mathcal{C}$ ist und diesen Begriff dann auf die zwei Randpunkte eines Intervalls anwenden, um so den Mittelwert eines solchen Intervalls zu bestimmen. Der Abstand zweier Intervalle wird dann als der Abstand ihrer Mittelwerte definiert werden.

Definition 6.13 *Seien $t_1, t_2 \in \mathcal{C}$. Dann heißt $m(t_1, t_2) \in \mathcal{C}$ der* Mittelwert *von t_1 und t_2 und wird wie folgt definiert:*

$$m(t_1, t_2) \;:=\; \begin{cases} t_1 \oplus (0, \lceil \frac{\langle t_1, t_2 \rangle_{\mathcal{C}}}{2} \rceil) & \text{falls } t_1 \leq_{\mathcal{C}} t_2 \\ t_2 \oplus (0, \lceil \frac{\langle t_2, t_1 \rangle_{\mathcal{C}}}{2} \rceil) & \text{sonst} \end{cases}$$

Definition 6.14 *Sei $I = [t_1, t_2] \in Int(\mathcal{C})$ ein Intervall über $\mathcal{C}$. Dann heißt $m(I) \in \mathcal{C}$ der* Mittelwert des Intervalls *I und wird wie folgt definiert:*

$$m(I) \;:=\; m(t_1, t_2)$$

Definition 6.15 *Seien $I_1, I_2 \in Int(\mathcal{C})$ Intervalle über $\mathcal{C}$. Dann heißt die Zahl $\langle I_1, I_2 \rangle_{Int(\mathcal{C})}$ der* Abstand der Zeitpunkt-Intervalle *I_1 und I_2 und wird definiert als:*

$$\langle I_1, I_2 \rangle_{Int(\mathcal{C})} \;:=\; \langle m(I_1), m(I_2) \rangle_{\mathcal{C}}$$

6.1.8 Wochentage

Eine wichtige Aufgabe, die über die Kalendersemantik zu lösen ist, besteht in der Bestimmung des Wochentages für ein gegebenes Datum. Mit anderen Worten, zu einem Zeitpunkt im Kalender muß der zugehörige Wochentag berechnet werden. Dies kann über die Bestimmung des Abstands zu einem anderen Zeitpunkt, für den der Wochentag bereits bekannt ist, erfolgen. Als Referenztag wähle ich den ersten Januar 1995, welcher ein Sonntag war. Es könnte aber ebensogut jeder andere Tag genommen werden.

Es wird eine Funktion weekday : $\mathcal{C} \rightarrow \{0, \ldots, 6\}$ definiert, die für einen Sonntag den Wert 0, für einen Montag den Wert 1, etc. liefert. Hierzu wird zunächst der Abstand zum Referenztag bestimmt. Die Uhrzeit spielt keine Rolle, weshalb sie für beide Argumente zum Beispiel auf zwölf Uhr mittags gesetzt werden kann. Teilt man den Abstand durch 1440 (das ist die Länge eines Tages in Minuten), so erhält man den Abstand in Tagen. (Eine direkte Definition des Abstandes in Tagen ist natürlich weitaus effizienter.) Der Rest schließlich, der beim Teilen dieser Anzahl von Tagen durch sieben bleibt, ist der gewünschte Funktionswert.

Definition 6.16 *Sei* $t = (y, m, d, h, i) \in \mathcal{C}$ *ein Zeitpunkt im Kalender. Dann heißt* $\mathrm{weekday}(t) \in \{0, \ldots, 6\}$ *mit*

$$\mathrm{weekday}(t) := (\langle (1995, 1, 1, 12, 0), (y, m, d, 12, 0) \rangle_{\mathcal{C}} \text{ div } 1440) \text{ mod } 7$$

die Wochentagsnummer *des Zeitpunktes* t.

Beachte, daß diese Funktion aufgrund der Definition negativer Abstände auch für Zeitpunkte vor dem ersten Januar 1995 das richtige Ergebnis liefert.

6.1.9 Julianische Tagesnummern

Die angegebenen Definitionen sind vor allem deklarativ zu lesen. Zwar ist die Definition von Addition und Subtraktion auch konstruktiv, jedoch aufgrund der n-fachen Anwedung des succ- beziehungsweise des pred-Operators wenig effizient in Bezug auf die Berechnungszeit. Der Abstand wurde allerdings rein deklarativ definiert, wenngleich sich natürlich leicht ein primitiver Algorithmus ableiten ließe.

Zur effizienteren Berechnung des Abstandes zweier Zeitpunkte können die sogenannten Julianischen Tagesnummern der entsprechenden Daten herangezogen werden (siehe [Mitton, 1978]). Jedem Tag im Gregorianischen Kalender entspricht eine natürliche Zahl, die Julianische Tagesnummer (kurz JDN für *Julian day number*) dieses Tages. Sie gibt an, wieviele Tage zwischen dem ersten Januar des Jahres 4713 v.Chr. und dem jeweiligen Datum liegen. Die Wahl dieses speziellen Referenzjahres hängt mit bestimmten Eigenheiten dieses Jahres in Bezug auf diverse astronomische Zyklen zusammen und geht auf Joseph Justus Scalinger zurück. Der Abstand zweier Zeitpunkte in Tagen kann also bestimmt werden, indem zunächst die Julianischen Tagesnummern der beiden Zeitpunkte berechnet werden und diese dann subtrahiert werden.

In [Tantzen, 1963] sind Algorithmen sowohl für die Umrechnung eines Datums in die entsprechende Tagesnummer als auch für die Umrechnung von Tagesnummern in Daten angegeben (siehe Tafel 6.1 beziehungsweise 6.2).

Der in Definition 6.12 definierte Abstand zweier Zeitpunkte in Minuten kann nun unter Verwendung des Algorithmus' zur Umrechnung eines Datums in die entsprechende Julianische Tagesnummer berechnet werden. Die Differenz der Tagesnummern liefert zunächst den Abstand in Tagen. Hierzu muß dann noch der sich zusätzlich durch die Uhrzeiten ergebende Abstand addiert werden. Sei

KONVERSION: DATUM → JULIANISCHE TAGESNUMMER

Eingabe: Datum als Tripel (y, m, d) Ausgabe: Julianische Tagesnummer j Es wird die dem durch (y, m, d) beschriebenen Datum entsprechende Julianische Tagesnummer j berechnet. Quelle: [Tantzen, 1963]	$\begin{array}{rcl} y' & := & \begin{cases} y & \text{falls } m > 2 \\ y-1 & \text{sonst} \end{cases} \\ c & := & y' \text{ div } 100 \\ r & := & y' - 100 \cdot c \\ m' & := & \begin{cases} m-3 & \text{falls } m > 2 \\ m+9 & \text{sonst} \end{cases} \\ j & := & (146097 \cdot c) \text{ div } 4 \\ & & + \ (1461 \cdot r) \text{ div } 4 \\ & & + \ (153 \cdot m' + 2) \text{ div } 5 \\ & & + \ d + 1721119 \end{array}$

Tafel 6.1: Algorithmus zur Berechnung der JDN

JDN die durch den in Tafel 6.1 angegebenen Algorithmus implementierte Funktion, die zu gegebenem Datum (y, m, d) die zugehörige Julianische Tagesnummer $j = \text{JDN}(y, m, d)$ liefert. Es gilt für beliebige $t_1 = (y_1, m_1, d_1, h_1, i_1)$ und $t_2 = (y_2, m_2, d_2, h_2, i_2)$ aus $\mathcal{C}$:

$$\langle t_1, t_2 \rangle_{\mathcal{C}} = 1440 \cdot (\text{JDN}(y_2, m_2, d_2) - \text{JDN}(y_1, m_1, d_1)) + 60 \cdot (h_2 - h_1) + i_2 - i_1$$

Definition 6.16 erklärt die Funktion 'weekday' zur Berechnung des Wochentages, an dem ein gegebener Zeitpunkt liegt. Sie kann nun auch ohne Bezug auf den Abstand in Minuten beschrieben werden. Die Julianische Tagesnummer des verwendeten Referenztages, des ersten Januar 1995, ist 2449719. Bei einer Division durch 7 bleibt als Rest 6 (beachte: $6 \equiv -1$ modulo 7). Also gilt für alle $t = (y, m, d, h, i) \in \mathcal{C}$:

$$\begin{array}{rcl} \text{weekday}(t) & = & (\text{JDN}(y, m, d) - \text{JDN}(1995, 1, 1)) \bmod 7 \\ & = & (\text{JDN}(y, m, d) - 2449719) \bmod 7 \\ & = & (\text{JDN}(y, m, d) + 1) \bmod 7 \end{array}$$

Sei JDN^{-1} die durch den Algorithmus in Tafel 6.2 definierte Funktion. Sie bildet natürliche Zahlen auf Tripel aus Jahres-, Monats- und Tagesangabe ab (also zum Beispiel $\text{JDN}^{-1}(j) = (y, m, d)$). Diese Funktion kann nun verwendet werden, um die Addition von Minuten zu Zeitpunkten (siehe Definition 6.8) effizienter zu formulieren. Es werden zunächst volle Tage addiert und danach die verbleibenden Minuten wie bisher durch Mehrfachanwendung des succ-Operators hinzugezählt. Die Addition von Tagen wird mit Hilfe der JDN-Algorithmen bewerkstelligt. Für die Subtraktion gilt entsprechendes.

Für $t = (y, m, d, h, i) \in \mathcal{C}$ und $n \in \mathbb{N}_0$ gilt: [4]

$$\begin{array}{rcl} t \oplus_{min} n & = & \text{succ}^{(n \bmod 1440)}(\text{JDN}^{-1}(\text{JDN}(t) + n \text{ div } 1440) \times (h, i)) \\ t \ominus_{min} n & = & \text{pred}^{(n \bmod 1440)}(\text{JDN}^{-1}(\text{JDN}(t) - n \text{ div } 1440) \times (h, i)) \end{array}$$

[4] Der Kreuz-Operator $\times$ angewandt auf zwei Tupel bezeichne eine Art Verkettung von Tupeln: $(y, m, d) \times (h, i) = (y, m, d, h, i)$.

<table>
<tr><th colspan="2">KONVERSION: JULIANISCHE TAGESNUMMER → DATUM</th></tr>
<tr><td>Eingabe: Julianische Tagesnummer j
Ausgabe: Jahr y, Monat m und Tag d

Es wird das der Julianischen Tagesnummer j entsprechende Datum (y, m, d) berechnet.

Quelle: [Tantzen, 1963]</td>
<td>$$\begin{aligned}
j_1 &:= j - 1721119\\
y_1 &:= (4 \cdot j_1 - 1) \text{ div } 146097\\
j_2 &:= 4 \cdot j_1 - 1 - 146097 \cdot y_1\\
d_1 &:= j_2 \text{ div } 4\\
j_3 &:= (4 \cdot d_1 + 3) \text{ div } 1461\\
d_2 &:= (4 \cdot d_1 - 1461 \cdot j_3 + 7) \text{ div } 4\\
m_1 &:= (5 \cdot d_2 - 3) \text{ div } 153\\
y_2 &:= 100 \cdot y_1 + j_3\\
d &:= (5 \cdot d_2 - 153 \cdot m_1 + 2) \text{ div } 5\\
m &:= \begin{cases} m_1 + 3 & \text{falls } m < 10\\ m_1 - 9 & \text{sonst}\end{cases}\\
y &:= \begin{cases} y_2 & \text{falls } m < 10\\ y_2 + 1 & \text{sonst}\end{cases}
\end{aligned}$$</td></tr>
</table>

Tafel 6.2: Algorithmus zur Gewinnung des Datums aus der JDN

6.1.10 Zusammenfassung: Kalendersemantik

Im ersten Teil dieses Kapitels wurden einige für die Formulierung der Semantik der TEMPORAL EXPRESSION LANGUAGE wichtige Definitionen gegeben. Insbesondere wurden die Menge der zulässigen Zeitpunkte im Kalender $\mathcal{C}$ sowie die Menge der Intervalle über ihr, $Int(\mathcal{C})$, definiert. Es wurde gezeigt, daß Zeitdauern sich *nicht* generell in *einer* Grundeinheit (etwa Minuten) messen lassen. Sie werden auch weiterhin als Paare von Monaten und Minuten angegeben werden. Beachte, daß sich auf der Menge dieser Paare aus den genannten Gründen keine totale Ordnung (im Sinne des Vergleichs der Längen von Zeitspannen) angeben läßt, und daß daher auch keine Intervalle über solchen Paaren gebildet werden können.

Die Verschiebung eines Zeitpunktes in die Zukunft beziehungsweise in die Vergangenheit wird über die Addition beziehungsweise Subtraktion von Elementen aus $\mathcal{C}$ und Monats-/Minuten-Paaren (Elementen aus $\mathcal{D}$) beschrieben. Des weiteren wurde der Abstand zweier Zeitpunkte definiert. Zu jedem Zeitpunkt im Kalender berechnet die Funktion 'weekday' eine Zahl zwischen 0 und 6, die angibt, an welchem Wochentag der jeweilige Zeitpunkt liegt.

Schließlich wurde gezeigt, wie die anfallenden Berechnungen unter Verwendung bekannter Algorithmen zur Julianischen Tagesnummer effizienter gemacht werden können.

6.2 Allgemeines zur Semantik von TEL-Ausdrücken

Im weiteren Verlauf dieses Kapitels wird für jede der drei TEL-Sprachen $L(\texttt{DURATION})$, $L(\texttt{POINT})$ und $L(\texttt{DATE})$ ein semantischer Bereich sowie eine Interpretationsfunktion von der jeweiligen Sprache auf den assoziierten semantischen Bereich festgelegt werden. Hierbei wird die Arbeit zur Semantik von Ausdrücken der Kategorie `DATE` naturgemäß stark auf die Ergebnisse der Beschäftigung mit `DURATION` und `POINT` aufbauen. Insbesondere bei der Interpretation von `POINT`-Ausdrücken wiederum werden die Ergebnisse des ersten Teils dieses Kapitels eine wichtige Rolle spielen.

6.2.1 Interpretationsalternativen

Die meisten zeitlichen Ausdrücke sind unterspezifiziert. Ist etwa von "Ostern" die Rede, so wird dieser Ausdruck in den seltensten Fällen durch eine direkte Angabe des Jahres konkretisiert werden. Solch eine *Unterspezifikation* sollte in der Semantik von einer *Uneindeutigkeit* unterschieden werden. Wird von "drei Uhr" gesprochen, so ist nicht klar, ob drei oder 15 Uhr gemeint ist. Bei der Generierung von TEL-Ausdrücken werden sicherlich auch Fälle auftreten, in denen einem natürlichsprachlichen Ausdruck kein TEL-Ausdruck eindeutig zugeordnet werden kann. Mit Hilfe des in den drei wesentlichen Subsprachen vorhandenen `one_out_of`-Operators können mehrere Alternativen zu einem Ausdruck zusammengefaßt werden, wobei der Sprecher nur genau eine davon tatsächlich gemeint hat. Bei der Definition der semantischen Bereiche für die einzelnen Subsprachen werde ich solche Interpretationsalternativen stets als Elemente einer dem Gesamtausdruck entsprechenden Menge darstellen. Hierbei wird betont, daß der Unterschied zwischen diesen Alternativen gewichtiger ist als der zwischen unterschiedlichen Konkretisierungen eines unterspezifizierten Ausdrucks.

Man könnte diese Uneindeutigkeit natürlich auch aus der Definition einer Interpretationsfunktion herausnehmen, und sie nur für die eindeutigen Fälle definieren. Für die uneindeutigen würden dann lediglich Aussagen über die Zugehörigkeit der Interpretation zu einer Menge von Werten gemacht werden können. Ein derartiger Ansatz würde zwar die Definition der Semantik vereinfachen, ihren Gebrauch jedoch komplizieren. Über Alternativen könnten auch unterschiedliche Auffassungen, welches der erste Tag einer Woche ist (und ähnliches mehr) repräsentiert werden. (Darauf werde ich in der folgenden Darstellung allerdings verzichten.) Außerdem können so verschiedene Kandidaten bei der Anapherauflösung quasi parallel weiterverarbeitet werden.

6.2.2 Zahlenangaben in TEL

TEL-Ausdrücke, die Zahlen beschreiben, tauchen an verschiedenen Stellen in komplexen Ausdrücken auf. Daher sollte die Semantik solcher Zahlen-Ausdrücke vorab festgelegt werden. Ich werde hierzu die Interpretationsfunktion $\mathcal{I}_N$ definieren, die Zahlen bezeichnende TEL-Ausdrücke auf die Menge der rationalen Zahlen abbildet:

$$\mathcal{I}_N : L(\texttt{INTEGER}) \cup L(\texttt{NUMBER}) \cup L(\texttt{NUMBERO}) \cup L(\texttt{FRACTION}) \rightarrow \mathbb{Q}$$

Im einzelnen gibt es Ausdrücke für ganze Zahlen, für natürliche Zahlen mit und ohne 0 sowie für Brüche von natürlichen Zahlen.

Die Sprache $L(\texttt{INTEGER})$ ist gerade die Menge der ganzen Zahlen. $L(\texttt{NUMBER})$ ist gleich der Menge der natürlichen Zahlen, und $L(\texttt{NUMBER0})$ ist die Menge der natürlichen Zahlen einschließlich der 0:

$$L(\texttt{INTEGER}) = \mathbb{Z} \qquad L(\texttt{NUMBER}) = \mathbb{N} \qquad L(\texttt{NUMBER0}) = \mathbb{N}_0$$

Die Interpretationsfunktion $\mathcal{I}_N$ eingeschränkt auf eine dieser drei Subsprachen ist also jeweils die identische Abbildung über der entsprechenden Menge:

$$\mathcal{I}_N|_{L(\texttt{INTEGER})} \equiv \mathrm{id}_{\mathbb{Z}} \qquad \mathcal{I}_N|_{L(\texttt{NUMBER})} \equiv \mathrm{id}_{\mathbb{N}} \qquad \mathcal{I}_N|_{L(\texttt{NUMBER0})} \equiv \mathrm{id}_{\mathbb{N}_0}$$

Elemente der Sprache $L(\texttt{FRACTION})$ sind einfache natürliche Zahlen sowie positive Brüche und gemischte Zahlen. Für Ausdrücke der Kategorie FRACTION wird $\mathcal{I}_N$ wie folgt definiert:

$$\begin{array}{lcll} \mathcal{I}_N(n) & = & n & \text{für } n \in \mathbb{N} \\ \mathcal{I}_N(n_1 : n_2) & = & \frac{n_1}{n_2} & \text{für } n_1, n_2 \in \mathbb{N} \\ \mathcal{I}_N(n_1 : n_2 : n_3) & = & n_1 + \frac{n_2}{n_3} & \text{für } n_1, n_2, n_3 \in \mathbb{N} \end{array}$$

Um die Lesbarkeit der Definitionen in den folgenden Abschnitten zu erhöhen, werde ich für Ausdrücke $n \in L(\texttt{NUMBER})$ statt $\mathcal{I}_N(n)$ stets einfach nur n schreiben. Gleiches gilt für NUMBER0 und INTEGER. Für Ausdrücke $\varphi \in L(\texttt{FRACTION})$ hingegen muß zwischen φ (einem TEL-Ausdruck, zum Beispiel `3:4`) und seiner Interpretation $\mathcal{I}_N(\varphi)$ (einer rationalen Zahl, im Beispiel $0,75$) unterschieden werden.

6.3 Ausdrücke der Kategorie DURATION

6.3.1 Motivation

Zunächst einmal möchte ich die semantische Interpretation für TEL-Ausdrücke der Kategorie DURATION intuitiv motivieren. Eine einfache Dauerangabe wie etwa "fünf Minuten" oder "anderthalb Stunden" kann sehr einfach als die Länge dieser Dauer in Minuten interpretiert werden. Es bietet sich dabei an, gegebenenfalls auf ganze Minuten zu runden. Weiter oben (Abschnitt 6.1.4) wurde bereits darauf hingewiesen, daß sich Dauerangaben, die in einer der Einheiten Monate oder Jahre gemacht werden, nicht als eine Zahl von Minuten interpretieren lassen. Einfache Dauer-Ausdrücke (das sind die der Kategorie BASIC_DUR) werden daher als Paare von Monats- und Minutenangabe interpretiert werden. Werden explizit Mengen von Dauern genannt, zum Beispiel "zwei oder drei Tage", so kann für die Semantik eines solchen Ausdrucks die Menge der Semantiken der einzelnen Ausdrücke definiert werden.

Die Menge der Paare aus Monaten und Minuten kann bezüglich der Längen der ihnen entsprechenden Zeitdauern nicht total geordnet werden. Ein Monat wird dargestellt als $(1, 0)$ und kann mal länger, mal kürzer als 30 Tage (43200 Minuten), das heißt als $(0, 43200)$ sein. Daher können auf dieser Menge von Paaren keine Intervalle definiert werden, da die Angabe zweier Randpunkte nicht ausreicht, um für alle möglichen Paare anzugeben, ob sie zu dem jeweiligen Intervall gehören oder nicht. Es stellt sich also die Frage, ob für TEL-Ausdrücke der Kategorie `RANGE` überhaupt ein adäquater semantischer Bereich definiert werden kann. Aufgrund des bisher gesagten steht sicher fest, daß dies jedenfalls nicht in Form einer expliziten Aufzählung aller möglichen Werte erreicht werden kann. Dies scheint zwar auf den ersten Blick ein schwerwiegender Nachteil zu sein, reflektiert jedoch bei genauerer Betrachtung lediglich die nicht eliminierbare implizite Unschärfe auch der intuitiven Semantik von Dauerangaben. Ist von einem Zeitraum von "drei Wochen bis zu einem Monat" Länge die Rede, so ist keine allgemeingültige Aussage darüber möglich, ob hierzu auch ein Zeitraum von 30 Tagen zählt. Solche Dauerangaben werden erst im Zusammenhang mit bestimmten Zeitpunkten zu exakten und vergleichbaren Werten. Die Semantik von Ausdrücken in $L($`DURATION`$)$ wird also nicht einfach als die Menge der möglichen Dauern beschrieben werden können. Vielmehr wird sie auf einem höheren Abstraktionslevel bleiben müssen, das auch die Interpretation von Ausdrücken des Typs `RANGE` erlaubt. Hiezu werde ich sogenannte *Pseudo-Intervalle* definieren. Sie werden als Paare geschrieben, haben jedoch die Bedeutung von Intervallen. Im Gegensatz zu normalen Intervallen können sie aber nicht in Mengen umgewandelt werden.

Verschiedene mit dem `one_out_of`-Operator beschriebene Alternativen müssen quasi auf einer 'höheren Ebene' als die übrigen Phänomene verwaltet werden. Die Interpretation zweier Alternativen muß vollkommen unabhängig voneinander geschehen, da ja nur durch *eine* der Alternativen die eigentliche, richtige Bedeutung des natürlichsprachlichen Ausdrucks beschrieben wird. Eine einzelne Alternative wird also wie bisher beschrieben behandelt werden. Eine Menge von Alternativen kann dann als die Menge der Interpretationen der einzelnen Alternativen interpretiert werden.

6.3.2 Semantischer Bereich und Interpretationsfunktion $\mathcal{I}_{DUR}$

In Definition 6.10 wurde bereits die Menge der Paare aus Monats- und Minutenangaben $\mathcal{D}$ definiert. Zwar können nicht alle Elemente dieser Menge geordnet werden, wenigstens für einige Fälle ist dies aber doch möglich (fünf Minuten sind sicherlich immer kürzer als ein Monat, etc.).

Definition 6.17 *Über $\mathcal{D}$, der Menge der Monats-/Minuten-Paare, wird für $(n_{11}, n_{12}), (n_{21}, n_{22}) \in \mathcal{D}$ definiert:*

$$\begin{aligned}(n_{11}, n_{12}) \leq_{\mathcal{D}} (n_{21}, n_{22}) \quad :\Longleftarrow \quad & 28 \cdot 1440 \cdot n_{11} + n_{12} \leq 28 \cdot 1440 \cdot n_{21} + n_{22} \ \wedge \\ & 31 \cdot 1440 \cdot n_{11} + n_{12} \leq 31 \cdot 1440 \cdot n_{21} + n_{22}\end{aligned}$$

Beachte, daß diese 'Ordnung' also *nicht* total ist. Die Berechnung der Relation erfolgt durch eine untere (28 Tage) und eine obere Abschätzung (31 Tage).[5]

Grundelement des für $L(\texttt{DURATION})$ zu definierenden semantischen Bereichs wird die Menge der bereits angesprochenen Pseudo-Intervalle sein, deren Definition im folgenden gegeben werden soll.

Definition 6.18 *Über der Menge der Monats-/Minuten-Paare $\mathcal{D}$ wird die Menge der* Pseudo-Intervalle $Int^*(\mathcal{D})$ *definiert:*

$$Int^*(\mathcal{D}) := \{(d_1, d_2) \in \mathcal{D} \times \mathcal{D} \mid d_1 \leq_{\mathcal{D}} d_2\}$$

Ein Pseudo-Intervall $(d_1, d_2) = ((n_{11}, n_{12}), (n_{21}, n_{22}))$ wird assoziiert mit der Menge der Dauern zwischen d_1 und d_2, wenngleich diese Menge nicht explizit angegeben werden kann.

Ein einfacher Ausdruck, in dem kein Bereich, sondern nur eine einzelne Dauerangabe vorkommt, kann natürlich ebenfalls als Pseudo-Intervall repräsentiert werden, nämlich als eines, bei dem Anfangs- und Endpunkt identisch sind. Auf diese Weise ist eine einheitlichere Darstellung möglich. Um die Vereinigung von Ausdrücken darstellen zu können, werden alle Ausdrücke als Mengen von Pseudo-Intervallen interpretiert werden. Für den Fall, daß es mehrere Interpretationalternativen gibt, werden die ihnen entsprechenden Mengen wiederum in einer Menge zusammengefaßt.

Aus dem bisher gesagten geht hervor, daß $\mathcal{P}(\mathcal{P}(Int^*(\mathcal{D})))$, also die Menge der Mengen von Pseudo-Intervallen über Paaren natürlicher Zahlen (mit 0)[6], ein geeigneter semantischer Bereich für die Interpretation von TEL-Ausdrücken der Kategorie DURATION ist. Die Interpretationsfunktion $\mathcal{I}_{DUR}$ bilde diese Ausdrücke auf den semantischen Bereich ab:

$$\mathcal{I}_{DUR} : L(\texttt{DURATION}) \rightarrow \mathcal{P}(\mathcal{P}(Int^*(\mathcal{D})))$$

Angenommen, ein Ausdruck $\delta \in L(\texttt{DURATION})$ würde als $\mathcal{I}_{DUR}(\delta) = \{\{((0,5),(0,5)),((1,0),(2,0))\}\}$ interpretiert. Da die Gesamtmenge einelementig ist, gibt es also nur eine Interpretationalternative (was in der Praxis sicherlich der Normalfall sein wird). Die innere Menge sagt aus, daß es sich um die disjunktive Verbindung zweier Dauerangaben handelt. Die erste dieser Angaben ist "fünf Minuten". Bei der zweiten handelt es sich um ein (echtes) Pseudo-Intervall. Es steht für die Dauerangabe von "ein bis zwei Monaten". $\mathcal{I}_{DUR}(\delta)$ entspricht also der natürlichsprachliche Ausdruck "fünf Minuten oder ein bis zwei Monate".

Beachte, daß durch die Verwendung von Mengen explizite Normalisierung überflüssig wird. Werden etwa bei einer Operation zwei identische Alternativen

[5] Sie ließe sich noch etwas stärker (aber jedenfalls nicht als totale Ordnung) formulieren, wenn man berücksichtigt, daß aufeinanderfolgende Monate meist unterschiedliche Anzahlen von Tagen haben. Zum Beispiel haben drei Monate maximal 92 Tage (nicht 93).

[6] In der Praxis wird man $\mathcal{D}$ selbstverständlich auf einen sinnvollen Bereich einschränken, also insbesondere (∞, ∞) nicht zulassen.

M zusammen in eine Menge gefaßt, so wird durch die Regeln der Mengenlehre eine der beiden automatisch gelöscht. Bei einer Implementierung mit Listen müssen doppelte gegebenenfalls explizit gelöscht werden.

Ausdrücke der Kategorie `FUZZY_DUR` und solche der Form `dur(several,UNIT)` gehören nicht zum Kern von TEL. Selbstverständlich kann die Semantik eines anaphorischen Ausdrucks nur dann berechnet werden, wenn die jeweiligen Anaphern aufgelöst, das heißt, wenn ihnen konkrete Werte zugewiesen werden konnten. Für `ana_dur` selbst kann also keine Interpretation angegeben werden. Für alle übrigen Ausdrücke in $L(\texttt{DURATION})$ wird im folgenden die semantische Interpretationsfunktion $\mathcal{I}_{DUR}$ definiert.

6.3.3 Einfache Ausdrücke (BASIC_DUR)

Ich beginne mit den Ausdrücken der Kategorie `BASIC_DUR`. Durch einem Ausdruck `dur(FRACTION,UNIT)` wird eine Zeitdauer repräsentiert, deren Länge sich als Produkt des Wertes an der Stelle von `FRACTION` mit dem Wert der Einheit `UNIT` in Minuten beziehungsweise in Monaten ergibt. Es wird gegebenfalls auf ganze Minuten gerundet. Bruchteile von Monaten sind *auch im Kontext einer Zeitpunktangabe* unpräzise formulierte Dauern (siehe hierzu die Ausführungen in Abschnitt 6.1.4). Daher werde ich die nicht-ganzzahligen Anteile solcher Dauerangaben in Monaten direkt in Tage (beziehungsweise in Minuten) umrechnen, indem ich von einer Zahl von 30 Tagen (beziehungsweise 43200 Minuten) in einem Monat ausgehe.

Um die Beschreibung der Interpretation von Ausdrücken der Kategorie `BASIC_DUR` zu vereinfachen, werde ich zunächst definieren, wie positive rationale Zahlen und Monats-/Minuten-Paare multipliziert werden. Für rationale Zahlen $q \in \mathbb{Q}_0^+$ bezeichne $\lfloor q \rfloor$ den ganzzahligen Anteil der Zahl q. Den verbleibenden nicht-ganzzahligen Teil werde ich $\mathrm{rest}(q) := q - \lfloor q \rfloor$ nennen. Auf der Menge der rationalen Zahlen $\mathbb{Q}$ sei außerdem ein Rundungsoperator 'round' definiert.

Definition 6.19 *Sei $q \in \mathbb{Q}_0^+$ und sei $(n_1, n_2) \in \mathcal{D}$. Das Produkt dieser Werte wird definiert als:*

$$q \cdot (n_1, n_2) \;:=\; (\lfloor q \cdot n_1 \rfloor, \mathrm{round}(43200 \cdot \mathrm{rest}(q \cdot n_1) + q \cdot n_2))$$

Für natürliche Zahlen $q \in \mathbb{N}_0$ gilt also: $q \cdot (n_1, n_2) = (q \cdot n_1, q \cdot n_2)$.

Für $\varphi \in L(\texttt{FRACTION})$ wird nun definiert:

$$\begin{aligned}
\mathcal{I}_{DUR}(\texttt{dur(}\varphi\texttt{,minutes))} &= \{\{(d,d) \mid d = \mathcal{I}_N(\varphi) \cdot (0,1)\}\} \\
\mathcal{I}_{DUR}(\texttt{dur(}\varphi\texttt{,hours))} &= \{\{(d,d) \mid d = \mathcal{I}_N(\varphi) \cdot (0,60)\}\} \\
\mathcal{I}_{DUR}(\texttt{dur(}\varphi\texttt{,days))} &= \{\{(d,d) \mid d = \mathcal{I}_N(\varphi) \cdot (0,1440)\}\} \\
\mathcal{I}_{DUR}(\texttt{dur(}\varphi\texttt{,weeks))} &= \{\{(d,d) \mid d = \mathcal{I}_N(\varphi) \cdot (0,10080)\}\} \\
\mathcal{I}_{DUR}(\texttt{dur(}\varphi\texttt{,months))} &= \{\{(d,d) \mid d = \mathcal{I}_N(\varphi) \cdot (1,0)\}\} \\
\mathcal{I}_{DUR}(\texttt{dur(}\varphi\texttt{,years))} &= \{\{(d,d) \mid d = \mathcal{I}_N(\varphi) \cdot (12,0)\}\}
\end{aligned}$$

Alle weiteren Ausdrücke haben Argumente, die wiederum Ausdrücke der Kategorie DURATION sind. Es muß also berücksichtigt werden, daß es sich bei der Semantik eines solchen Teilausdrucks stets um eine Menge von Alternativen handelt (oft sicherlich nur einer Alternative, was für die Definition der Interpretationsfunktion aber keine Rolle spielt). Hat ein Ausdruck zwei oder mehr solcher Argumente, so muß für die Semantik eine Art Kreuzprodukt gebildet werden: jede mögliche Kombination von Alternativen der einzelnen Argumente ergibt eine Alternative im Ergebnis.

6.3.4 Bereiche von Zeitdauern (RANGE und OPEN_RANGE)

Als nächstes betrachte ich Bereiche, das heißt die Kategorien RANGE und OPEN_RANGE. Im Falle von RANGE gibt es zwei Argumente. Der tatsächliche Bereich versteht sich als die Menge der Alternativen, die man erhält, wenn man paarweise für jede Alternative des ersten Arguments und jede Alternative des zweiten Arguments den durch diese beiden begrenzten Bereich, ein Pseudo-Intervall, bildet. Dabei ist es möglich, daß solch eine Bereichsgrenze nicht unbedingt aus nur genau einem Wert besteht. Ein Beispiel hierfür wäre etwa der Satz "Treffen wir uns für ein bis drei oder vier Tage". In solch einem Falle muß für jedes Paar von Grenzen ein neues Pseudo-Intervall gebildet werden. Jedes dieser Pseudo-Intervalle übernimmt den linken Randpunkt der linken Grenze und den rechten Randpunkt der rechten Grenze.

Für $\delta_1, \delta_2 \in L(\texttt{DURATION})$ gilt also:

$$\begin{aligned}\mathcal{I}_{DUR}(\texttt{range}(\delta_1, \delta_2)) \quad = \quad & \{\{(d_{11}, d_{22}) \in Int^*(\mathcal{D}) \mid \\ & (d_{11}, d_{12}) \in M_1 \wedge (d_{21}, d_{22}) \in M_2\} \mid \\ & M_1 \in \mathcal{I}_{DUR}(\delta_1) \wedge M_2 \in \mathcal{I}_{DUR}(\delta_2)\}\end{aligned}$$

Werden inkonsistente Angaben gemacht (etwa "zehn bis fünf Minuten"), so werden diese Ausdrücke durch die in $Int^*(\mathcal{D})$ implementierte Ordnungsrelation $\leq_{\mathcal{D}}$ in den meisten Fällen "herausgefiltert". Ob aber ein Ausdruck wie "30 Tage bis ein Monat" inkonsistent ist oder nicht, kann an dieser Stelle nicht bestimmt werden. Sobald solche Dauerangaben aber in Bezug zu einem Zeitpunkt gesetzt werden, wird die Inkonsistenz (falls vorhanden) offensichtlich.

Bei Ausdrücken des Typs OPEN_RANGE wird analog vorgegangen. Für jedes Pseudo-Intevall in jeder der Mengen in jeder Alternativmenge wird ein neues Pseudo-Intervall gebildet. Es wird zwischen den 'inklusiv'-Varianten (`at_least` und `at_most`) und den 'exklusiv'-Varianten (`more` und `less`) unterschieden. Allerdings werde ich darauf verzichten, bei letzteren auch tatsächlich den jeweiligen Randpunkt selbst auch auszuschließen (es geht schließlich nur um eine Minute)[7]. Daraus folgt, daß für ein Pseudo-Intervall, dessen linker Randpunkt mit seinem rechten identisch ist, sich die 'inklusiv'- nicht von der 'exklusiv'-Variante unterscheidet.

[7] Würde man in $\mathcal{D}$ für den Minuten-Wert auch negative Zahlen zulassen, so könnte man leicht eine Subtraktion von einer Minute durchführen und so die 'exklusiv'-Varianten auch streng formulieren.

Es wird also für $\delta \in L(\texttt{DURATION})$ definiert:

$$\begin{array}{lcl}
\mathcal{I}_{DUR}(\texttt{at_least(}\delta\texttt{)}) & = & \{\{(d_1, (\infty, \infty)) \mid (d_1, d_2) \in M\} \mid M \in \mathcal{I}_{DUR}(\delta)\} \\
\mathcal{I}_{DUR}(\texttt{more(}\delta\texttt{)}) & = & \{\{(d_2, (\infty, \infty)) \mid (d_1, d_2) \in M\} \mid M \in \mathcal{I}_{DUR}(\delta)\} \\
\mathcal{I}_{DUR}(\texttt{at_most(}\delta\texttt{)}) & = & \{\{((0,0), d_2) \mid (d_1, d_2) \in M\} \mid M \in \mathcal{I}_{DUR}(\delta)\} \\
\mathcal{I}_{DUR}(\texttt{less(}\delta\texttt{)}) & = & \{\{((0,0), d_1) \mid (d_1, d_2) \in M\} \mid M \in \mathcal{I}_{DUR}(\delta)\}
\end{array}$$

6.3.5 Modifizierte Ausdrücke (MODIFIED_DUR)

Im letzten Kapitel wurde gezeigt, wie Ausdrücke für unscharfe Dauerangaben, also Ausdrücke der Kategorie FUZZY_DUR, in modifizierte Ausdrücke, also solche der Kategorie MODIFIED_DUR, umgewandelt werden können. Die Darstellung in Form von MODIFIED_DUR-Ausdrücken hat den Vorteil, daß sich so eine allgemeingültige Interpretation angeben läßt. Zur Interpretation eines Ausdrucks der Form `modified_dur(FRACTION,FRACTION,DURATION)` wird für jedes Pseudo-Intervall in der Interpretation des DURATION-Ausdrucks die linke Pseudo-Intervallgrenze mit dem ersten und die rechte Grenze mit dem zweiten FRACTION-Wert multipliziert.

Seien $\varphi_1, \varphi_2 \in L(\texttt{FRACTION})$ und $\delta \in L(\texttt{DURATION})$; es gilt:

$$\begin{array}{lcl}
\mathcal{I}_{DUR}(\texttt{modified_dur(}\varphi_1, \varphi_2, \delta\texttt{)}) & = & \{\{(\mathcal{I}_N(\varphi_1) \cdot d_1, \mathcal{I}_N(\varphi_2) \cdot d_2) \in Int^*(\mathcal{D}) \mid \\
& & \quad (d_1, d_2) \in M\} \mid M \in \mathcal{I}_{DUR}(\delta)\}
\end{array}$$

6.3.6 Vereinigung (set)

Mit dem `set`-Operator können mehrere Ausdrücke der Kategorie DURATION auf syntaktischer Ebene disjunktiv zusammengefaßt werden. Auf der semantischen Ebene entspricht dies einer Vereinigung. Beispielsweise wird "zwei oder drei Stunden" als `set([dur(2,hours),dur(3,hours)])` repräsentiert. Das bedeutet, die bezeichnete Dauer kann 60 Minuten oder 120 Minuten sein; alle anderen Möglichkeiten sind ausgeschlossen. Für alle Kombinationen von Alternativen aus der Argumentenliste wird eine Vereinigung von Mengen berechnet.

Da es sich bei dieser semantischen Vereinigung um eine grundsätzliche Eigenschaft des für $L(\texttt{DURATION})$ gewählten semantischen Bereichs $\mathcal{P}(\mathcal{P}(Int^*(\mathcal{D})))$ handelt, werde ich sie zunächst unabhängig von der Interpretation des `set`-Operators definieren.

Definition 6.20 *Seien* $D_1, \ldots, D_n \in \mathcal{P}(\mathcal{P}(Int^*(\mathcal{D})))$. *Die* semantische Vereinigung *dieser Mengen wird wie folgt definiert:*

$$D_1 \sqcup \cdots \sqcup D_n \;:=\; \{M_1 \cup \cdots \cup M_n \mid M_1 \in D_1 \wedge \cdots \wedge M_n \in D_n\}$$

Die Interpretation eines durch `set` eingeleiteten Ausdrucks berechnet sich also als die semantische Vereinigung der Interpretationen der einzelnen Ausdrücke. Seien $\delta_1, \ldots, \delta_n \in L(\texttt{DURATION})$. Dann gilt:

$$\mathcal{I}_{DUR}(\texttt{set([}\delta_1, \ldots, \delta_n\texttt{])}) \;=\; \mathcal{I}(\delta_1) \sqcup \cdots \sqcup \mathcal{I}(\delta_n)$$

6.3.7 Alternativen (`one_out_of`)

Für den Fall, daß bei der Generierung eines TEL-Ausdruckes für eine bestimmte natürlichsprachliche Äußerung Schwierigkeiten auftreten, den korrekten Ausdruck (automatisch) zuzuordenen, da zwischen zwei oder auch mehr Bedeutungsalternativen nicht unterschieden werden kann, wurde der `one_out_of`-Operator in allen drei Subsprachen eingeführt. Es ist allerdings unwahrscheinlich, daß seine Anwendung im Falle der Sprache $L(\texttt{DURATION})$ tatsächlich einmal nötig sein sollte. Uneindeutigkeiten sind sicherlich eher bei den komplexeren Sprachen zu erwarten. Die Semantik des Operators muß aber dennoch definiert werden.

Ein Ausdruck der Kategorie `DURATION` wird als eine Menge von Mengen interpretiert. Die einzelnen Mengen stehen hierbei für die Alternativen, die mit Hilfe eines `one_out_of`-Operators explizit benannt werden können. Wird eine Reihe solcher Ausdrücke nun wiederum als Menge von Alternativen zusammengefaßt, so müssen die Interpretationen der Einzelausdrücke vereinigt werden. Die Alternativmengen der einzelnen Argumente stehen dann also quasi gleichberechtigt nebeneinander. Ich definiere für $\delta_1, \ldots, \delta_n \in L(\texttt{DURATION})$:

$$\mathcal{I}_{DUR}(\texttt{one\_out\_of([}\delta_1, \ldots, \delta_n\texttt{])}) \quad = \quad \mathcal{I}_{DUR}(\delta_1) \cup \cdots \cup \mathcal{I}_{DUR}(\delta_n)$$

Beachte den Unterschied zwischen dieser herkömmlichen Vereingung von Mengen und der im letzten Abschnitt auf $\mathcal{P}(\mathcal{P}(\mathit{Int}^*(\mathcal{D})))$ definierten *semantischen* Vereinigung.

6.4 Ausdrücke der Kategorie POINT

6.4.1 Motivation

Im ersten Teil dieses Kapitels wurde gezeigt, wie sich über ihre Position in unserem Kalender definierte Zeitpunkte als Elemente der Menge $\mathcal{C}$ darstellen lassen.[8] Mit der Sprache $L(\texttt{POINT})$ können solche Zeitpunkte ebenfalls spezifiziert werden. Häufiger jedoch wird man sich dieser Sprache bedienen, um einen bestimmten Zeitpunkt lediglich einzugrenzen. Sinn und Zweck von $L(\texttt{POINT})$ aber ist es, entweder den Anfangs- oder den Endpunkt eines Termins zu umschreiben. Es geht also tatsächlich um eindeutige Zeitpunkte. Insofern können TEL-Ausdrücke, die sich nicht auf einen eindeutig bestimmbaren Zeitpunkt beziehen, als unterspezifiziert bezeichnet werden. Daß die meisten zeitlichen Ausdrücke unterspezifiziert sind, liegt nicht nur an der unpräzisen Ausdrucksweise der Sprecher. Da Termine meist sukzessive vereinbart werden, indem wechselseitig ein zu Anfang gegebener

[8] Während der Entwicklung der Semantik von TEL habe ich lange Zeit eine Darstellung von Zeitpunkten als einfache natürliche Zahlen favorisiert. Hierzu müßte ein Nullpunkt festgelegt werden; die Zahlen entsprächen dann der Zahl der Minuten, die seitdem vergangen sind. Im Prinzip sind beide Darstellungen nartürlich isomorph. Der Ansatz mit lediglich einer Zahl hat aber den schwerwiegenden Nachteil, daß sich Verschiebung um in Monaten angegebene Zeitdauern kaum anschaulich machen läßt. Außerdem ginge die Festlegung auf einen bestimmten Nullpunkt mit einigen formalen Unschönheiten einher.

Ausdruck immer weiter konkretisiert wird, meinen die Sprecher nicht unbedingt nur genau einen Zeitpunkt, sondern gegebenenfalls mehrere Auswahlzeitpunkte, die die gleichen durch den Ausdruck geäußerten Eigenschaften teilen.

Eine Möglichkeit, einen unterspezifizierten Ausdruck semantisch darzustellen, ist, alle mit der Aussage dieses Ausdrucks vereinbaren Zeitpunkte (als Elemente von $\mathcal{C}$) zusammenzufassen. Eine Menge aufeinanderfolgender Zeitpunkte läßt sich auch als Intervall, also als Element aus $Int(\mathcal{C})$ schreiben. Tatsächlich entsprechen den allermeisten TEL-Ausdrücken Mengen solcher Intervalle. Eine einfache Angabe wie "montags" steht für die Menge all jener Zeitpunkt-Intervalle, deren linker Randpunkt an einem Montag um null Uhr und dessen rechter Randpunkt am jeweils gleichen Montag um 23.59 Uhr liegt. Letztlich interessiert nur die Menge der Zeitpunkte, die durch einen Ausdruck beschrieben wird; letztlich sind die Intervalle also auch nichts weiter als eine abkürzende Schreibweise. Dies ist jedoch nur für die fertig berechnete Semantik eines Ausdrucks richtig. Im Verlaufe der Berechnung können Zeitpunkt-Intervalle sehr wohl auch mehr als nur die Menge ihrer Elemente sein. In der natürlichen Sprache, und konsequenterweise auch in der Temporal Expression Language werden nämlich vielfach zusammenhängende Verbände von Zeitpunkten als eigenständige Objekte manipuliert. Um etwa den "dritten Dienstag nach Ostern" abzählen zu können, genügt es nicht, alle Zeitpunkte, die an einem Dienstag liegen, zu kennen, sondern es muß stattdessen die Menge der Dienstage selbst betrachtet werden. Ähnliches gilt bei der Interpretation eines Ausdrucks wie "in der ersten Hälfte des Monats".

Die Definition der Semantik von TEL muß sich selbstverständlich am Zwecke der Entwicklung dieser Sprache orientieren. Es geht, wie bereits erwähnt, um die Eingrenzung eines Zeitpunktes (eventuell auch einer Reihe von Zeitpunkten). Darüber hinausgehende Vielfalt in den unterschiedlichen natürlichen Sprachen kann und soll nicht berücksichtigt werden. Aus diesem Grundsatz folgt etwa, daß (bereits bei der Festlegung der Syntax) nicht zwischen den Angaben "an einem Montag" und "montags" unterschieden wird. Daß im zweiten Ausdruck eine Art Frequenz beschrieben wird, ist für das Ziel der Bestimmung *eines* Zeitpunktes unwesentlich. Es interessiert ausschließlich, daß der gesuchte Zeitpunkt die Eigenschaft hat, an einem Montag zu liegen.

Allerdings erlaubt TEL die Differenzierung zwischen "an *einem* Montag Zeit haben" und beispielsweise "an *zwei* Montagen Zeit haben" (mit Ausdrücken der Kategorie `QUANTIFIED`). Möchte man also explizit ausdrücken, daß der gesuchte Zeitpunkt nur an *einem* Montag oder an einem von *zwei* Montagen (eventuell innerhalb eines vorher eingegrenzten Zeitraumes) liegen kann, so ist dieses möglich. Da bei solch einer Aussage jedoch nicht klar ist, um welchen oder um welche zwei Montag(e) es sich handelt, ist ihr Informationsgehalt auch nicht höher als der von "montags". Die weiter unten vorgestellte Semantik läßt zwar eine Unterscheidung zu, trotzdem scheint in der Praxis der Nutzen der `QUANTIFIED`-Kategorie eher fragwürdig. Bei einer möglichen Erweiterung von TEL dahingehend, mehr als einen Termin mit ein und demselben Ausdruck umschreiben zu können (zum Beispiel für sich regelmäßig wiederholende Treffen und dergleichen), würde ihr aber unter Umständen einer größere Bedeutung zukommen.

In den meisten Fällen wird einem Ausdruck der Sprache $L(\texttt{POINT})$ also eine Menge von Intervallen über $\mathcal{C}$ entsprechen. Die Menge der Zeitpunkte in diesen Intervallen ist die Menge der Zeitpunkte, die durch den entsprechenden Ausdruck umschrieben werden. Außerdem können die Intervalle selbst Objekte von Manipulationen, also etwa Abzählungen oder Kürzungen sein. Um unterschiedliche Interpretationsalternativen darstellen zu können, muß wie auch schon bei $L(\texttt{DURATION})$ noch eien weitere Menge eingeführt werden, die die den verschiedenen Alternativen entsprechenden Mengen enthält. Das Konzept der Alternativmengen wird übrigens auch verwendet werden, um die gerade diskutierten quantifizierten Ausdrücke zu interpretieren. Wenn für den Ausdruck "an zwei Montagen" nicht bekannt ist, welche zwei Montage gemeint sind, so bildet jede in Frage kommende Auswahl zweier Montage eine Alternative.

6.4.2 Semantischer Bereich und Interpretationsfunktion $\mathcal{I}_P$

Wird das bisher über die Semantik von Ausdrücken des Typs `POINT` gesagte formalisiert, so ergibt sich, daß $\mathcal{P}(\mathcal{P}(\mathit{Int}(\mathcal{C})))$ ein geeigneter semantischer Bereich[9] für die Sprache $L(\texttt{POINT})$ ist. (Einzelne Zeitpunkte können als Intervalle mit identischen Randpunkten dargestellt werden.) Nach der TEL-Grammatik sind generische Ausdrücke (also solche der Kategorie `GENERIC`) nicht Teil der Sprache $L(\texttt{POINT})$, sondern sie tauchen lediglich als Argumente von `POINT`-Ausdrücken auf. Trotzdem muß auch für diese Ausdrücke (ähnlich wie für Zahlenangaben, siehe oben) eine Semantik formuliert werden, da bei der Interpretation eines Ausdrucks, der einen generischen Ausdruck als Argument hat, auf die Interpretation des letzteren im Laufe der Berechnung zugegriffen werden können muß. Da sowohl für Ausdrücke der Kategorie `POINT` als auch für solche der Kategorie `GENERIC` der gleiche gerade benannte semantische Bereich geeignet ist, und da auch die Definition einer Interpretationsfunktion in beiden Fällen in analoger Weise geschehen würde, werde ich für beide Sprachen ein und dieselbe Funktion $\mathcal{I}_P$ definieren:

$$\mathcal{I}_P : L(\texttt{POINT}) \cup L(\texttt{GENERIC}) \rightarrow \mathcal{P}(\mathcal{P}(\mathit{Int}(\mathcal{C})))$$

Um die Struktur des gewählten semantischen Bereiches noch einmal zu verdeutlichen, soll ein einfaches Beispiel angeführt werden. Gilt für einen Ausdruck $\pi \in L(\texttt{POINT})$

$$\begin{aligned}\mathcal{I}_P(\pi) \quad = \quad & \{\{[(1998, 3, 3, 14, 0), (1998, 3, 3, 18, 0)], \\ & [(1998, 7, 29, 20, 30), (1998, 7, 29, 20, 30)]\}\},\end{aligned}$$

so bedeutet das, daß π für die Aussage "am dritten März 1998 zwischen 14 und 18 Uhr oder am 29. Juli 1998 abends um halb neun" steht. Hierbei wurde nur eine Interpretationsalternative generiert (das heißt, die Hauptmenge ist einelementig).

Beachte, daß $\mathcal{I}_P$ nach der im letzten Kapitel vorgenommenen Reduktion der TEMPORAL EXPRESSION LANGUAGE nur noch für ihren Kern (siehe Anhang B.3)

[9] In der Praxis wird man natürlich nicht sämtliche Zeitpunkte aus $\mathcal{C}$ zulassen, sondern nur solche, die im Bereich des zu verhandelnden Termins liegen.

definiert werden muß. Da die Interpretation eines anaphorischen Ausdrucks erst möglich wird, sobald die Anapher(n) aufgelöst wurde(n), kann für Ausdrücke der Kategorie `ANA_POINT` keine Interpretation angegeben werden.

6.4.3 Einfache Ausdrücke (`BASIC`)

Zu den meisten Ausdrücke der Kategorie `BASIC` konnten im letzten Kapitel Transformationsregeln angegeben werden, die die Definition von $\mathcal{I}_P$ für diese Ausdrücke überflüssig macht. In der Tat gehören lediglich `now`, `moh:MOH`, `tod:TOD`, `pod:am`, `pod:pm` sowie `year:YEAR` zum Kern von TEL. Mit Hilfe des `not`-Operators wäre es darüber hinaus sogar möglich, `pod:pm` durch `pod:am` (oder andersherum) zu umschreiben. Zunächst werde ich die Interpretationsfunktion also für die genannten Ausdrücke definieren. Darüber hinaus gebe ich auch noch an, wie die Interpretation einiger wichtiger einfacher Ausdrücke direkt berechnet werden könnte, also ohne den Umweg über die Transformationsregeln. Einem solchen Vorgehen würde man in der Praxis aus Gründen der Effizienz möglicherweise den Vorzug geben.

Es sei 'now' eine nullstellige Funktion, die stets den aktuellen Zeitpunkt als Element von $\mathcal{C}$ liefert. Aufgerufen am 31. Dezember 1999 um 23.59 Uhr würde sie also beispielsweise den Funktionswert $(1999, 12, 31, 23, 59)$ liefern. Die Semantik des TEL-Ausdrucks `now` kann dann sehr einfach folgendermaßen beschrieben werden:

$$\mathcal{I}_P(\texttt{now}) \quad = \quad \{\{[t,t] \in \mathit{Int}(\mathcal{C}) \mid t = \mathrm{now}\}\}$$

Ausdrücke wie "um viertel nach" oder "zur vollen Stunde" werden in TEL als `moh:MOH` dargestellt, wobei die Kategorie `MOH` identisch mit `NUMBER0` ist. Bei der Berechnung der Semantik wird die in Definition 6.3 eingeführte Funktion 'minute' verwendet. Sei $i \in L(\texttt{NUMBER0})$; dann gilt:

$$\mathcal{I}_P(\texttt{moh:}i) \quad = \quad \{\{[t,t] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{minute}(t) = i\}\}$$

Beachte, daß durch die Bedingung $[t,t] \in \mathit{Int}(\mathcal{C})$ sichergestellt wird, daß t ein gültiger Zeitpunkt ist, daß i also eine natürliche Zahl zwischen 0 und 59 ist. In diesem Falle könnte diese Bedingung natürlich auch leicht im Rahmen der Überprüfung der Syntax eines Ausdrucks getestet werden. Im allgemeinen (etwa bei `dom:DOM`) ist dies jedoch nicht möglich, weshalb bei der Definition der Interpreationsfunktion stets darauf zu achten ist, daß (semantisch) inkonsistente TEL-Ausdrücke als leere Mengen interpretiert werden.

Für Uhrzeiten beschreibende Ausdrücke wird ähnlich wie für `moh:MOH` vorgegangen. Allerdings ist hier zu berücksichtigen, daß mit einer Angabe wie etwa `tod:5:0` stets zweierlei gemeint sein kann. Entweder geht es um fünf Uhr morgens oder um 17 Uhr abends. Ein Sprecher intendiert sicherlich nur eine dieser beiden Lesarten. Es handelt sich also um zwei Bedeutungsalternativen, die parallel zu verarbeiten sind. $\mathcal{I}_P$ bildet Ausdrücke der Kategorie `tod:TOD` also auf eine Menge von zwei Alternativmengen ab. In der ersten stehen all jene Zeitpunkte (als Intervalle geschrieben), die der jeweiligen Uhrzeit *vor* zwölf Uhr mittags entsprechen, in der zweiten all jene, die dieser Uhrzeit *nach* zwölf Uhr mittags entsprechen. Für $h, i \in L(\texttt{NUMBER0})$ wird definiert:

$$\begin{aligned}\mathcal{I}_P(\texttt{tod:}h\texttt{:}i) &= \{\{[t,t] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t) = h \wedge \mathrm{minute}(t) = i\},\\ &\quad\ \{[t,t] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t) = h + 12 \wedge \mathrm{minute}(t) = i\}\}\end{aligned}$$

Für `TOD` *sollen* ausschließlich Angaben zwischen `0:0` und `11:59` gemacht werden, da nur für sie eine *am*- und eine *pm*-Variante existieren. "Zwölf Uhr mittags" muß in TEL also als "null Uhr pm" kodiert werden. Für Werte zwischen `12:0` und `23:59` liefert $\mathcal{I}_P$ allerdings ebenfalls ein sinnvolles Ergebnis. In diesen Fällen führt die zweite Interpretationsalternative zur Inkonsistenz (da sich bei der Addition von 12 ein Stunden-Wert größer 23 ergibt), also zu einer leeren Menge. Die erste Alternative hingegen wird korrekt berechnet.

Der Teil eines Tages zwischen null Uhr und 11.59 Uhr wird durch den Ausdruck `pod:am` beschrieben, der Rest des Tages durch `pod:pm`. Das sind jeweils zwölf Stunden, also 720 Minuten. Also ist die Semantik dieser Ausdrücke die folgende:

$$\begin{aligned}\mathcal{I}_P(\texttt{pod:am}) &= \{\{[t, t \oplus (0,719)] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t) = 0 \wedge \mathrm{minute}(t) = 0\}\}\\ \mathcal{I}_P(\texttt{pod:pm}) &= \{\{[t, t \oplus (0,719)] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t) = 12 \wedge \mathrm{minute}(t) = 0\}\}\end{aligned}$$

Schließlich gehören noch die Jahresangaben zu den einfachen Ausdrücken der Kernsprache. Ein Jahr hat zwölf Monate. Wenn man also zu einem Zeitpunkt t_1, der der erste Zeitpunkt eines Jahres ist (also erster Januar, null Uhr), zwölf Monate addiert und hiervon wieder eine Minute abzieht, so liegen in dem so beschriebenen Intervall alle Zeitpunkte des jeweiligen Jahres. Es wird also für $y \in L(\texttt{NUMBER})$ definiert:

$$\begin{aligned}\mathcal{I}_P(\texttt{year:}y) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \wedge \mathrm{day}(t_1) = 1\\ &\quad \wedge \mathrm{month}(t_1) = 1 \wedge \mathrm{year}(t_1) = y \wedge t_2 = (t_1 \oplus (12,0)) \ominus (0,1)\}\}\end{aligned}$$

Im folgenden werde ich $\mathcal{I}_P$ noch auf einigen einfachen Ausdrücken, die eigentlich nicht in der Kernsprache sind, definieren. Die Berechnung der Semantik läßt sich auf dem direkten Wege vermutlich effizienter implementieren. Zur Berechnung der Semantik von Ausdrücken für Wochentage wird die in Definition 6.16 eingeführte Funktion 'weekday' benutzt. Über sie wird der Anfangspunkt des jeweiligen Wochentages bestimmt. Ein Tag hat 1440 Minuten; der Endpunkt liegt also 1439 Minuten nach dem Anfangspunkt.

$$\begin{aligned}
\mathcal{I}_P(\texttt{dow:mon}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \\
&\quad \wedge \mathrm{weekday}(t_1) = 1 \wedge t_2 = t_1 \oplus (0,1439)\}\} \\
\mathcal{I}_P(\texttt{dow:tue}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \\
&\quad \wedge \mathrm{weekday}(t_1) = 2 \wedge t_2 = t_1 \oplus (0,1439)\}\} \\
\mathcal{I}_P(\texttt{dow:wed}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \\
&\quad \wedge \mathrm{weekday}(t_1) = 3 \wedge t_2 = t_1 \oplus (0,1439)\}\} \\
\mathcal{I}_P(\texttt{dow:thu}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \\
&\quad \wedge \mathrm{weekday}(t_1) = 4 \wedge t_2 = t_1 \oplus (0,1439)\}\} \\
\mathcal{I}_P(\texttt{dow:fri}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \\
&\quad \wedge \mathrm{weekday}(t_1) = 5 \wedge t_2 = t_1 \oplus (0,1439)\}\} \\
\mathcal{I}_P(\texttt{dow:sat}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \\
&\quad \wedge \mathrm{weekday}(t_1) = 6 \wedge t_2 = t_1 \oplus (0,1439)\}\} \\
\mathcal{I}_P(\texttt{dow:sun}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \\
&\quad \wedge \mathrm{weekday}(t_1) = 0 \wedge t_2 = t_1 \oplus (0,1439)\}\}
\end{aligned}$$

Ein weiterer oft gebrauchter Ausdruck ist `dom:DOM` zur Spezifizierung des Tages innerhalb eines Monats. Seine Semantik wird ähnlich wie für die Wochentage definiert. Sei $d \in L(\texttt{NUMBER})$; es gilt:

$$\begin{aligned}
\mathcal{I}_P(\texttt{dom:}d) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \\
&\quad \wedge \mathrm{day}(t_1) = d \wedge t_2 = t_1 \oplus (0,1439)\}\}
\end{aligned}$$

Auch die Semantik von Monatsangaben ergibt sich bereits indirekt über die im letzten Kapitel genannten Transformationsregeln, kann aber ebensogut direkt formuliert werden. Wie Ausdrücke der Form `month:MONTH` zu interpretieren sind, zeigt die folgende Liste.

$$\begin{aligned}
\mathcal{I}_P(\texttt{month:jan}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \wedge \\
&\quad \mathrm{day}(t_1) = d \wedge \mathrm{month}(t_1) = 1 \wedge t_2 = (t_1 \oplus (1,0)) \ominus (0,1)\}\} \\
\mathcal{I}_P(\texttt{month:feb}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \wedge \\
&\quad \mathrm{day}(t_1) = d \wedge \mathrm{month}(t_1) = 2 \wedge t_2 = (t_1 \oplus (1,0)) \ominus (0,1)\}\} \\
\mathcal{I}_P(\texttt{month:mar}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \wedge \\
&\quad \mathrm{day}(t_1) = d \wedge \mathrm{month}(t_1) = 3 \wedge t_2 = (t_1 \oplus (1,0)) \ominus (0,1)\}\} \\
\mathcal{I}_P(\texttt{month:apr}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \wedge \\
&\quad \mathrm{day}(t_1) = d \wedge \mathrm{month}(t_1) = 4 \wedge t_2 = (t_1 \oplus (1,0)) \ominus (0,1)\}\} \\
\mathcal{I}_P(\texttt{month:may}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \wedge \\
&\quad \mathrm{day}(t_1) = d \wedge \mathrm{month}(t_1) = 5 \wedge t_2 = (t_1 \oplus (1,0)) \ominus (0,1)\}\} \\
\mathcal{I}_P(\texttt{month:jun}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \wedge \\
&\quad \mathrm{day}(t_1) = d \wedge \mathrm{month}(t_1) = 6 \wedge t_2 = (t_1 \oplus (1,0)) \ominus (0,1)\}\}
\end{aligned}$$

$$
\begin{aligned}
\mathcal{I}_P(\texttt{month:jul}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \text{hour}(t_1) = 0 \wedge \text{minute}(t_1) = 0 \wedge \\
&\quad \text{day}(t_1) = d \wedge \text{month}(t_1) = 7 \wedge t_2 = (t_1 \oplus (1,0)) \ominus (0,1)\}\} \\
\mathcal{I}_P(\texttt{month:aug}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \text{hour}(t_1) = 0 \wedge \text{minute}(t_1) = 0 \wedge \\
&\quad \text{day}(t_1) = d \wedge \text{month}(t_1) = 8 \wedge t_2 = (t_1 \oplus (1,0)) \ominus (0,1)\}\} \\
\mathcal{I}_P(\texttt{month:sep}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \text{hour}(t_1) = 0 \wedge \text{minute}(t_1) = 0 \wedge \\
&\quad \text{day}(t_1) = d \wedge \text{month}(t_1) = 9 \wedge t_2 = (t_1 \oplus (1,0)) \ominus (0,1)\}\} \\
\mathcal{I}_P(\texttt{month:oct}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \text{hour}(t_1) = 0 \wedge \text{minute}(t_1) = 0 \wedge \\
&\quad \text{day}(t_1) = d \wedge \text{month}(t_1) = 10 \wedge t_2 = (t_1 \oplus (1,0)) \ominus (0,1)\}\} \\
\mathcal{I}_P(\texttt{month:nov}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \text{hour}(t_1) = 0 \wedge \text{minute}(t_1) = 0 \wedge \\
&\quad \text{day}(t_1) = d \wedge \text{month}(t_1) = 11 \wedge t_2 = (t_1 \oplus (1,0)) \ominus (0,1)\}\} \\
\mathcal{I}_P(\texttt{month:dec}) &= \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \text{hour}(t_1) = 0 \wedge \text{minute}(t_1) = 0 \wedge \\
&\quad \text{day}(t_1) = d \wedge \text{month}(t_1) = 12 \wedge t_2 = (t_1 \oplus (1,0)) \ominus (0,1)\}\}
\end{aligned}
$$

6.4.4 Generische Ausdrücke (GENERIC)

Die generischen Ausdrücke selbst gehören nicht der Kategorie POINT an, da der Informationsgehalt dieser Ausdrücke für sich allein genommen in Hinblick auf die Eingrenzung eines Zeitpunktes gleich null ist. Als Argumente anderer Ausdrücke spielen sie aber sehr wohl eine Rolle. Zum Beispiel ist die Aussage, daß ein Zeitpunkt “in einem Monat” liegt, wenig sinnvoll. Wird hingegen über einen Zeitpunkt geäußert, daß er etwa “am Anfang eines Monats” oder auch “am letzten Sonntag eines Monats” liegt, so kann die Menge der in Frage kommenden Zeitpunkte durchaus eingeschränkt werden. Die Semantik von Ausdrücken der Kategorie GENERIC wird also letztlich nur für die Berechnung anderer Ausdrücke benötigt.

Die Menge aller Zeitpunkte der Semantik eines generischen Ausdrucks wird die Menge aller Zeitpunkte in $\mathcal{C}$ sein. Das folgt schon aus dem gerade diskutierten Nicht-Vorhandensein eines Informationsgehalts solcher Ausdrücke. Enscheidendes Merkmal der Interpretation dieser Ausdrücke ist die Art der Aufteilung der Zeitpunkte in Intervalle. Beispielsweise muß `month` als die Menge der Intervalle, deren linker Randpunkt der erste Zeitpunkt und deren rechter Randpunkt der letzte Zeitpunkt des jeweils gleichen Monats ist. Es wird also jeweils der erste Zeitpunkt t_1 bestimmt und dann der letze Zeitpunkt der jeweiligen Zeitspanne durch Addition der passenden Zahl von Minuten oder Monaten gewonnen. Werden Monate addiert, so muß danach wieder eine Minute abgezogen werden, um nicht schon in das nächste Intervall hineinzuragen (bei der Addition von Miuten geht das in einem Schritt). Beachte, daß für `week` die Funktion ‘weekday’ gebraucht wird, um den Anfangspunkt auf einen Montag legen zu können. Mit `quarter_year` sind Quartale gemeint, also nicht beliebige Zeiträume von dreimonatiger Dauer. Entsprechendes gilt für `half_year`; jedes Jahr hat ein eindeutig definiertes erstes und ein zweites Halbjahr, das im Januar beziehungsweise im Juli beginnt.

$$
\begin{array}{lcl}
\mathcal{I}_P(\mathtt{day}) & = & \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \\
& & \wedge\ t_2 = t_1 \oplus (0, 1439)\}\} \\
\mathcal{I}_P(\mathtt{week}) & = & \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \\
& & \wedge\ \mathrm{weekday}(t_1) = 1 \wedge t_2 = t_1 \oplus (0, 10079)\}\} \\
\mathcal{I}_P(\mathtt{month}) & = & \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \\
& & \wedge\ \mathrm{day}(t_1) = 1 \wedge t_2 = (t_1 \oplus (1,0)) \ominus (0,1)\}\} \\
\mathcal{I}_P(\mathtt{quarter\_year}) & = & \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \\
& & \wedge\ \mathrm{day}(t_1) = 1 \wedge \mathrm{month}(t_1) \in \{1,4,7,10\} \\
& & \wedge\ t_2 = (t_1 \oplus (3,0)) \ominus (0,1)\}\} \\
\mathcal{I}_P(\mathtt{half\_year}) & = & \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \\
& & \wedge\ \mathrm{day}(t_1) = 1 \wedge \mathrm{month}(t_1) \in \{1,7\} \\
& & \wedge\ t_2 = (t_1 \oplus (6,0)) \ominus (0,1)\}\} \\
\mathcal{I}_P(\mathtt{year}) & = & \{\{[t_1,t_2] \in \mathit{Int}(\mathcal{C}) \mid \mathrm{hour}(t_1) = 0 \wedge \mathrm{minute}(t_1) = 0 \\
& & \wedge\ \mathrm{day}(t_1) = 1 \wedge \mathrm{month}(t_1) = 1 \\
& & \wedge\ t_2 = (t_1 \oplus (12,0)) \ominus (0,1)\}\}
\end{array}
$$

6.4.5 Modifizierte Ausdrücke (MODIFIED)

Im letzten Kapitel (Abschnitt 5.2) wurden die durch Lambda-Funktionen modifizierten Ausdrücke als Verallgemeinerung von modifizierten (MODIFIED) und unscharfen (FUZZY) Ausdrücken eingeführt. Zur Berechnung der Semantik eines solchen Ausdrucks muß die an erster Argumentsstelle angegebene Lambda-Funktion auf jedes einzelne Zeitpunkt-Intervall in der Interpretation des an zweiter Stelle angegebenen Ausdrucks angewandt werden.

Hierzu muß zunächst gezeigt werden, wie die als Ausdrücke der Kategorie LAMBDA kodierten Lambda-Funktionen in tatsächliche Funktionen umgewandelt werden können. Es muß also eine Interpretationfunktion $\mathcal{I}_\lambda$ zur Abbildung der Sprache $L(\mathtt{LAMBDA})$ auf die Menge der Lambda-Funktionen, welche Intervalle über $\mathcal{C}$ auf ebensolche abbilden, definiert werden:

$$\mathcal{I}_\lambda : L(\mathtt{LAMBDA}) \to \{\lambda : \mathit{Int}(\mathcal{C}) \to \mathit{Int}(\mathcal{C})\}$$

Für $\vartheta_1, \vartheta_2 \in L(\mathtt{LAMBDA\_ARG})$ wird definiert:

$$
\begin{array}{rcl}
\mathcal{I}_\lambda(\mathtt{lambda(}\vartheta_1, \vartheta_2\mathtt{))} & = & \lambda[t_1,t_2].[f(\vartheta_1), f(\vartheta_2)] \\
\text{mit } f : \vartheta \mapsto & & \left\{ \begin{array}{ll}
t_1 & \text{falls } \vartheta = \mathtt{left} \\
t_2 & \text{falls } \vartheta = \mathtt{right} \\
f(\vartheta') \oplus (0, \lfloor \mathcal{I}_N(\varphi) \cdot \langle t_1, t_2 \rangle_{\mathcal{C}} \rfloor) & \text{falls } \vartheta = \mathtt{plus(}\vartheta', \varphi\mathtt{)} \\
f(\vartheta') \ominus (0, \lceil \mathcal{I}_N(\varphi) \cdot \langle t_1, t_2 \rangle_{\mathcal{C}} \rceil) & \text{falls } \vartheta = \mathtt{minus(}\vartheta', \varphi\mathtt{)}
\end{array} \right.
\end{array}
$$

Wenn also $\lambda \in L(\mathtt{LAMBDA})$ ist, so ist $\mathcal{I}_\lambda(\lambda)$ eine gültige Lanbda-Funktion von $\mathit{Int}(\mathcal{C})$ nach $\mathit{Int}(\mathcal{C})$. Die Semantik des modified-Operators kann nun als die Anwendung von $\mathcal{I}_\lambda(\lambda)$ auf alle Intervalle in allen Alternativmengen der Interpretation eines Ausdrucks definiert werden.

Sei $\pi \in L(\texttt{POINT}) \cup L(\texttt{GENERIC})$ und sei $\lambda \in L(\texttt{LAMBDA})$; dann gilt:

$$\mathcal{I}_P(\texttt{modified}(\lambda, \pi)) \quad = \quad \{\{\mathcal{I}_\lambda(\lambda)(I) \mid I \in M\} \mid M \in \mathcal{I}_P(\pi)\}$$

Zur Illustration folgt ein einfaches Beispiel. Die mit dem TEL-Funktor `firsthalf` assoziierte Lambda-Funktion ist

$$\lambda[t_1, t_2].[t_1, t_1 \oplus (0, \lfloor 0,5 \cdot \langle t_1, t_2 \rangle_\mathcal{C} \rfloor)]$$

Wenn der TEL-Ausdruck $\pi \in L(\texttt{POINT})$ die Semantik

$$\mathcal{I}_P(\pi) = \{\{[(2000, 1, 14, 18, 0), (2000, 1, 14, 19, 0)]\}\}$$

hat, so wird der `MODIFIED`-Ausdruck `firsthalf(`π`)` zunächst mit Hilfe der im letzten Kapitel angegebenen Transformationsregel zu

$$\texttt{modified(lambda(left,plus(left,1:2)),}\pi\texttt{)}$$

reduziert und dann als

$$\{\{[(2000, 1, 14, 18, 0), (2000, 1, 14, 18, 30)]\}\}$$

interpretiert werden.

6.4.6 Intervalle (LIMIT und LIMITS)

Ich betrachte zunächst die einseitig unbegrenzten Intervalle, also die Ausdrücke des Typs `LIMIT`. Ein natürlichsprachlich geäußerter Ausdruck, der einem einseitig unbegrenzten Intervall entspricht, wird sicherlich in den seltensten Fällen auch tatsächlich *unbegrenzt* gemeint sein. Die Äußerung "nach dem 17. Oktober" wird sich kaum auf alle Zeitpunkte, die *irgendwann* – eventuell in fernster Zukunft – nach dem 17. Oktober liegen, beziehen. Bei der Definition der Semantik solcher Ausdrücke kann hierauf aber keine Rücksicht genommen werden, da es sich bei der Begrenzung solcher unbegrenzten Intervalle allenfalls um Heuristiken handeln kann (vergleiche Abschnitt 5.2.4 im letzten Kapitel).

Für $\pi \in L(\texttt{POINT})$ gilt:

$$\begin{aligned}
\mathcal{I}_P(\texttt{in_before}(\pi)) &= \{\{[-\infty_\mathcal{C}, t_2] \mid [t_1, t_2] \in M\} \mid M \in \mathcal{I}_P(\pi)\} \\
\mathcal{I}_P(\texttt{ex_before}(\pi)) &= \{\{[-\infty_\mathcal{C}, t_1 \ominus (0, 1)] \mid [t_1, t_2] \in M\} \mid M \in \mathcal{I}_P(\pi)\} \\
\mathcal{I}_P(\texttt{in_after}(\pi)) &= \{\{[t_1, \infty_\mathcal{C}] \mid [t_1, t_2] \in M\} \mid M \in \mathcal{I}_P(\pi)\} \\
\mathcal{I}_P(\texttt{ex_after}(\pi)) &= \{\{[t_2 \oplus (0, 1), \infty_\mathcal{C}] \mid [t_1, t_2] \in M\} \mid M \in \mathcal{I}_P(\pi)\}
\end{aligned}$$

Ausdrücke der Kategorie `LIMITS` gibt es nur als 'inklusiv'-Varianten. Mit `between` werden *alle* Intervalle, deren linker Randpunkt das an erster Stelle genannte Kriterium erfüllt und deren rechter Randpunkt dem an zweiter Stelle angegebenen Ausdruck entspricht, beschrieben. Die Interpretation eines solchen Ausdrucks ergibt sich, indem für jede Kombination von Alternativmengen der Interpretationen der beiden Argumente wiederum aus jeder Kombination von Zeitpunkt-Intervallen in diesen Alternativmengen ein neues Intervall gebildet wird. Für $\pi_1, \pi_2 \in L(\mathtt{POINT})$ wird also definiert:

$$\begin{array}{rcl}\mathcal{I}_P(\mathtt{between}(\pi_1,\pi_2)) & = & \{\{[t_{11},t_{22}] \in \mathit{Int}(\mathcal{C}) \mid \\ & & \quad [t_{11},t_{12}] \in M_1 \wedge [t_{21},t_{22}] \in M_2\} \mid \\ & & \quad M_1 \in \mathcal{I}_P(\pi_1) \wedge M_2 \in \mathcal{I}_P(\pi_2)\}\end{array}$$

Beachte, daß durch $[t_{11}, t_{22}] \in \mathit{Int}(\mathcal{C})$ sichergestellt wird, daß $t_{11} \leq_{\mathcal{C}} t_{22}$ ist.

Definition 6.21 *Die Funktion* $\mu : \mathcal{P}(\mathcal{P}(\mathit{Int}(\mathcal{C}))) \rightarrow \mathcal{P}(\mathcal{P}(\mathit{Int}(\mathcal{C})))$ *heißt* Minimierungsfunktion *über* $\mathcal{P}(\mathcal{P}(\mathit{Int}(\mathcal{C})))$ *und wird definiert wie folgt:*

$$\mu : P \longmapsto \{\{I \in M \mid \neg\exists\, I' \in M : I \supset I'\} \mid M \in P\}$$

Die Minimierungsfunktion μ löscht also bei der Interpretation $P = \mathcal{I}_P(\pi)$ eines Ausdrucks $\pi \in L(\mathtt{POINT})$ innerhalb jeder der Alternativmengen M all jene Intervalle, die von einem anderen Intervall aus M 'subsumiert' werden. Oder, um es anders zu formulieren, es bleiben jeweils die 'speziellesten' Intervalle übrig, also diejenigen mit dem höchsten Informationsgehalt.

Wird in einem Ausdruck `min_between` statt `between` verwendet, so soll die Interpretation nur die jeweils kürzesten Intervalle liefern. Also zum Beispiel für "von Montag bis Freitag" nicht etwa alle möglichen Intervalle, die an irgendeinem Montag beginnen und an einem beliebigen Freitag enden, sondern vielmehr nur diejenigen, bei denen der Freitag dem jeweiligen Montag unmittelbar folgt. Mit Hilfe der Minimierungsfunktion können alle überlangen Intervalle aussortiert werden. Es sei also für $\pi_1, \pi_2 \in L(\mathtt{POINT})$:

$$\mathcal{I}_P(\mathtt{min\_between}(\pi_1,\pi_2)) \quad = \quad \mu(\mathcal{I}_P(\mathtt{between}(\pi_1,\pi_2)))$$

6.4.7 Verschiebung (SHIFTED)

Mit Ausdrücken der Kategorie `SHIFTED` wird die Addition (beziehungsweise Subtraktion) von Zeitpunkten und Zeitdauern implementiert. Die Intervalle in der Interpretation eines Zeitpunktes $\pi \in L(\mathtt{POINT})$ werden um den durch einen Ausdruck $\delta \in L(\mathtt{DURATION})$ spezifizierten Betrag in die angegebene Richtung verschoben. Falls δ keine einfache Dauer ist, sondern als ein Pseudo-Intervall mit unterschiedlichen Randpunkten interpretiert wird, so wird der bezeichnete Zeitraum entsprechend 'aufgeweicht', das heißt, (bei einer Verschiebung in die Zukunft) linke Intervallgrenzen werden um den kleinsten Betrag verschoben, rechte um den größten (bei einer Verschiebung in die Vergangenheit ist es andersherum).

Sei also $\delta \in L(\mathtt{DURATION})$ und $\pi \in L(\mathtt{POINT})$:

$$\begin{array}{rcl}\mathcal{I}_P(\mathtt{pos\_shift}(\delta,\pi)) & = & \{\{[t_1 \oplus d_1, t_2 \oplus d_2] \in \mathit{Int}(\mathcal{C}) \mid \\ & & \quad (d_1,d_2) \in M_\delta \wedge [t_1,t_2] \in M_\pi\} \mid \\ & & \quad M_\delta \in \mathcal{I}_{DUR}(\delta) \wedge M_\pi \in \mathcal{I}_P(\pi)\} \\ \mathcal{I}_P(\mathtt{neg\_shift}(\delta,\pi)) & = & \{\{[t_1 \ominus d_2, t_2 \ominus d_1] \in \mathit{Int}(\mathcal{C}) \mid \\ & & \quad (d_1,d_2) \in M_\delta \wedge [t_1,t_2] \in M_\pi\} \mid \\ & & \quad M_\delta \in \mathcal{I}_{DUR}(\delta) \wedge M_\pi \in \mathcal{I}_P(\pi)\}\end{array}$$

6.4.8 Ausdrücke mit einer Abzählung (RELATED und ORDINAL)

Bei der Bestimmung der Semantik von Ausdrücken der Kategorien `RELATED` und `ORDINAL` spielt insbesondere das Abzählen von gewissen Objekten eine Rolle. Genauer, innerhalb der Interpretationen der jeweiligen Teilausdrücke der Kategorie `COUNTABLE` wird ein bestimmtes Intervall ausgewählt, das heißt abgezählt werden müssen. Ein solches Abzählen ist nur für bestimmte Mengen von Intervallen möglich, nämlich nur für solche, für deren Elemente eine Reihenfolge angegeben werden kann. Dies ist sicherlich dann der Fall, wenn je zwei Zeitpunkt-Intervalle in einer Menge keinerlei gemeinsame Zeitpunkte haben. Dann erschließt sich die Reihenfolge einfach durch die über $\mathcal{C}$ definierte Ordnung $\leq_{\mathcal{C}}$: es genügt für jedes Intervall eines ihrer Elemente, etwa einen der Randpunkte, als Referenz festzulegen.

Folgende Definition formalisiert, welches die Elemente des für $L(\texttt{POINT})$ gewählten semantischen Bereichs sind, in denen Zeitpunkt-Intervalle auf eindeutige Weise abgezählt werden können. Beachte hierbei, daß sich der Schnitt zweier Elemente aus $Int(\mathcal{C})$ nach den Regeln der Mengenlehre als der Schnitt der den beiden Intervallen entsprechenden Mengen berechnet. Er muß also nicht unbedingt wieder in $Int(\mathcal{C})$ liegen.

Definition 6.22 *$P \in \mathcal{P}(\mathcal{P}(Int(\mathcal{C})))$ heißt* abzählbar, *wenn in jeder der Alternativmengen von P der Schnitt je zweier Zeitpunkt-Intervalle leer ist, wenn also gilt:*

$$\forall M \in P : \forall I_1, I_2 \in M : I_1 \cap I_2 = \emptyset$$

Es kann leicht nachgeprüft werden, daß $\mathcal{I}_P(\gamma)$ für alle $\gamma \in L(\texttt{COUNTABLE1})$ abzählbar ist. Steht an der Stelle eines `COUNTABLE`-Ausdrucks `ana_point`, so ist bei der Anapherresolution darauf zu achten, daß nur ein Ersetzen mit einem anderen Ausdruck des Typs `COUNTABLE` sinnvoll ist. Andernfalls sollte die Interpretation des neuen Ausdrucks wenigstens abzählbar sein. Würde man einen anderen Wert einsetzen, so könnte $\mathcal{I}_P$ zwar immer noch berechnet werden; allerdings könnte hierbei natürlich kaum noch ein sinnvolles Ergebnis erwartet werden.

Im folgenden werden der nach Definition 6.14 bestimmbare Mittelwert von Intervallen über $\mathcal{C}$ sowie der in Definition 6.15 eingeführten Abstandsbegriff über $Int(\mathcal{C})$ eine Rolle spielen. Außerdem wird bei der Beschreibung der Semantik von Ausdrücken, in denen Teilausdrücke der Form `int:DURATION` vorkommen, die in Definition 6.19 definierte Multiplikation von Zahlen mit Elementen aus $\mathcal{D}$ benutzt werden.

Zunächst soll die Semantik von `the_around` beschrieben werden. Es gilt, für jedes Zeitpunkt-Intervall aus der Interpretation des `POINT`-Ausdrucks dasjenige Intervall innerhalb des interpretierten `COUNTABLE1`-Ausdrucks zu bestimmen, das ihm am nächsten liegt, für das also der Betrag des Abstands der Intervalle minimal ist. Argumente der Form `int:DURATION` müssen hier wie auch in den folgenden Fällen getrennt betrachtet werden.

$$\begin{array}{lcccccccccc}
 & & & & & & & & \text{“2. } \texttt{after } [t_{\pi 1}, t_{\pi 2}]\text{”} & & \\
 & & & & & & & & \Downarrow & & \\
M_\gamma: & \cdots & [\quad\quad] & [\quad\quad] & [\quad\quad] & [t'_{\gamma 1}\; t'_{\gamma 2}] & & & [t_{\gamma 1}\; t_{\gamma 2}] & [\quad\quad] & \cdots \\
 & & & [t_{\pi 1} & \quad t_{\pi 2}] & & & & & &
\end{array}$$

Tafel 6.3: Veranschaulichung der Semantik von `the_after`

Es seien $\gamma \in L(\texttt{COUNTABLE1})$ und $\pi \in L(\texttt{POINT})$:

$$\begin{aligned}
\mathcal{I}_P(\texttt{the_around}(\gamma, \pi)) \;=\; & \{\{I_\gamma \in M_\gamma \mid I_\pi \in M_\pi \wedge \\
& \quad \forall I'_\gamma \in M_\gamma : |\langle I_\pi, I_\gamma \rangle_{Int(\mathcal{C})}| \leq |\langle I_\pi, I'_\gamma \rangle_{Int(\mathcal{C})}|\} \mid \\
& \quad M_\gamma \in \mathcal{I}_P(\gamma) \wedge M_\pi \in \mathcal{I}_P(\pi)\}
\end{aligned}$$

Hat das erste Argument die Form `int:DURATION`, so muß für jedes Intervall ein neues Intervall bestimmt werden, dessen Dauer durch den jeweiligen `DURATION`-Ausdruck festgelegt wird und dessen Mittelwert mit dem des ersten Intervalls übereinstimmt. Hierbei gilt das zweite Element eines Pseudo-Intervalls (also die längere Dauer) als maßgeblich. “Die zwei bis drei Tage um den dritten” wird also genauso wie “die drei Tage um den dritten” interpretiert. Würde man zusätzlich für andere Dauern aus diesem Pseudo-Intervall weitere Zeitpunkt-Intervalle aufbauen, so wären diese ohnehin in dem ‘größeren’ enthalten.

Für $\delta \in L(\texttt{DURATION})$ und $\pi \in L(\texttt{POINT})$ wird also definiert:

$$\begin{aligned}
\mathcal{I}_P(\texttt{the_around(int:}\delta, \pi)) \;=\; & \{\{[t, t \oplus d_2 \ominus (0,1)] \in Int(\mathcal{C}) \mid \\
& \quad (d_1, d_2) \in M_\delta \wedge I_\pi \in M_\pi \wedge \\
& \quad t = m(I_\pi) \ominus \tfrac{1}{2} d_2\} \mid \\
& \quad M_\delta \in \mathcal{I}_{DUR}(\delta) \wedge M_\pi \in \mathcal{I}_P(\pi)\}
\end{aligned}$$

Bei der Interpretation von `the_after` beziehungsweise `the_before` müssen von einem bestimmten Zeitpunkt ausgehend Intervalle abgezählt werden. Bei `the_after` ist dieser Referenzzeitpunkt der rechte Randpunkt des jeweiligen Zeitpunkt-Intervalls aus der Interpretation des dritten Arguments; bei `the_before` ist es der linke. Die abzuzählenden Intervalle kommen aus der Interpretation des `COUNTABLE1`-Ausdrucks. Zum besseren Verständnis der folgenden Definition mag Tafel 6.3 hilfreich sein.

Für $n \in L(\texttt{NUMBER})$, $\gamma \in L(\texttt{COUNTABLE})$ und $\pi \in L(\texttt{POINT})$ wird definiert:

$$\begin{aligned}
\mathcal{I}_P(\texttt{the_after}(n, \gamma, \pi)) \;=\; & \{\{[t_{\gamma 1}, t_{\gamma 2}] \mid [t_{\gamma 1}, t_{\gamma 2}] \in M_\gamma \wedge [t_{\pi 1}, t_{\pi 2}] \in M_\pi \wedge \\
& \quad \#\{[t'_{\gamma 1}, t'_{\gamma 2}] \in M_\gamma \mid t_{\pi 2} \leq_{\mathcal{C}} t'_{\gamma 1} \leq_{\mathcal{C}} t_{\gamma 1}\} = n\} \mid \\
& \quad M_\gamma \in \mathcal{I}_P(\gamma) \wedge M_\pi \in \mathcal{I}_P(\pi)\} \\
\mathcal{I}_P(\texttt{the_before}(n, \gamma, \pi)) \;=\; & \{\{[t_{\gamma 1}, t_{\gamma 2}] \mid [t_{\gamma 1}, t_{\gamma 2}] \in M_\gamma \wedge [t_{\pi 1}, t_{\pi 2}] \in M_\pi \wedge \\
& \quad \#\{[t'_{\gamma 1}, t'_{\gamma 2}] \in M_\gamma \mid t_{\gamma 2} \leq_{\mathcal{C}} t'_{\gamma 2} \leq_{\mathcal{C}} t_{\pi 1}\} = n\} \mid \\
& \quad M_\gamma \in \mathcal{I}_P(\gamma) \wedge M_\pi \in \mathcal{I}_P(\pi)\}
\end{aligned}$$

Ist das zweite Argument aus $L(\texttt{int}:\texttt{DURATION})$, so muß im wesentlichen der n-fache Wert der entsprechenden Dauer auf $\mathcal{C}$ abgezählt werden.

Es seien $n \in L(\texttt{NUMBER})$, $\delta \in L(\texttt{DURATION})$ und $\pi \in L(\texttt{POINT})$; dann gilt:

$$\begin{aligned}
\mathcal{I}_P(\texttt{the_after}(n, \texttt{int}{:}\delta, \pi)) &= \{\{[t'_1, t'_2] \in \mathit{Int}(\mathcal{C}) \,|\, (d_1, d_2) \in M_\delta \wedge [t_1, t_2] \in M_\pi \\
&\qquad \wedge\, t'_1 = t_2 \oplus ((n-1) \cdot d_1) \oplus (0,1) \\
&\qquad \wedge\, t'_2 = t_2 \oplus (n \cdot d_2)\} \,| \\
&\qquad M_\delta \in \mathcal{I}_{DUR}(\delta) \wedge M_\pi \in \mathcal{I}_P(\pi)\} \\
\mathcal{I}_P(\texttt{the_before}(n, \texttt{int}{:}\delta, \pi)) &= \{\{[t'_1, t'_2] \in \mathit{Int}(\mathcal{C}) \,|\, (d_1, d_2) \in M_\delta \wedge [t_1, t_2] \in M_\pi \\
&\qquad \wedge\, t'_1 = t_1 \ominus (n \cdot d_2) \\
&\qquad \wedge\, t'_2 = t_1 \ominus ((n-1) \cdot d_1) \ominus (0,1)\} \,| \\
&\qquad M_\delta \in \mathcal{I}_{DUR}(\delta) \wedge M_\pi \in \mathcal{I}_P(\pi)\}
\end{aligned}$$

Bei der Definition der Semantik von Ausdrücken der Kategorie `ORDINAL` muß zusätzlich darauf geachtet werden, daß das jeweils bestimmte Intervall noch innerhalb des 'größeren' Referenzintervalls liegt.

Ein Beispiel für die Kategorie `ORDINAL` sind Ausdrücke, die Kalenderwochen, also Wochen im Jahr, beschreiben. Nach DIN-Norm 1355 aus dem Jahre 1974 (zitiert nach [Bien, 1988]) ist die erste Woche des Jahres die, die den ersten Donnerstag des Jahres enthält. Entsprechendes gilt natürlich auch für die letzte Kalenderwoche: sie ist die Woche, die den letzten Donnerstag des Jahres enthält. In den meisten anderen Fällen ist die Situation ohnehin eindeutig, da es keine 'Phasenverschiebung' wie zwischen Wochen und Jahren gibt. Welches etwa der erste Tag eines Monats sei, bedarf keiner Erklärung. Die Abzählung von 'kleinen' Intervallen in 'größeren' Bezugszeiträumen läßt sich in allgemeiner Form beschreiben, wenn festgelegt wird, daß die Zählung bei dem ersten Intervall beginnt, dessen Mittelwert innerhalb des Bezugszeitraumes liegt. Zum Beispiel ist der Mittelwert einer Woche Donnerstag, zwölf Uhr mittags.

Für $n \in \mathbb{N}$, $\gamma \in L(\texttt{COUNTABLE1})$ und $\pi \in L(\texttt{POINT}) \cup L(\texttt{GENERIC})$ wird definiert:

$$\begin{aligned}
\mathcal{I}_P(\texttt{of}(n, \gamma, \pi)) &= \{\{I_\gamma \in M_\gamma \,|\, [t_{\pi 1}, t_{\pi 2}] \in M_\pi \,\wedge \\
&\qquad \#\{I'_\gamma \in M_\gamma \,|\, t_{\pi 1} \leq_\mathcal{C} m(I'_\gamma) \leq_\mathcal{C} m(I_\gamma) \leq_\mathcal{C} t_{\pi 2}\} = n\} \,| \\
&\qquad M_\gamma \in \mathcal{I}_P(\gamma) \wedge M_\pi \in \mathcal{I}_P(\pi)\} \\
\mathcal{I}_P(\texttt{last_of}(n, \gamma, \pi)) &= \{\{I_\gamma \in M_\gamma \,|\, [t_{\pi 1}, t_{\pi 2}] \in M_\pi \,\wedge \\
&\qquad \#\{I'_\gamma \in M_\gamma \,|\, t_{\pi 1} \leq_\mathcal{C} m(I_\gamma) \leq_\mathcal{C} m(I'_\gamma) \leq_\mathcal{C} t_{\pi 2}\} = n\} \,| \\
&\qquad M_\gamma \in \mathcal{I}_P(\gamma) \wedge M_\pi \in \mathcal{I}_P(\pi)\}
\end{aligned}$$

Für Ausdrücke der Form `int:DURATION` beginnt die Abzählung am linken (beziehungsweise rechten bei `last_of`) Randpunkt eines jeden Bezugsintervalls.

Es seien $n \in L(\texttt{NUMBER})$, $\delta \in L(\texttt{DURATION})$ und $\pi \in L(\texttt{POINT}) \cup L(\texttt{GENERIC})$; dann gilt:

$$\begin{aligned}
\mathcal{I}_P(\texttt{of}(n, \texttt{int}:\delta, \pi)) &= \{\{[t'_1, t'_2] \in \mathit{Int}(\mathcal{C}) \mid (d_1, d_2) \in M_\delta \wedge [t_1, t_2] \in M_\pi \\
&\qquad \wedge\ t'_1 = t_1 \oplus ((n-1) \cdot d_1) \\
&\qquad \wedge\ t'_2 = t_1 \oplus (n \cdot d_2) \ominus (0,1) \leq_{\mathcal{C}} t_2\} \mid \\
&\qquad M_\delta \in \mathcal{I}_{DUR}(\delta) \wedge M_\pi \in \mathcal{I}_P(\pi)\} \\
\mathcal{I}_P(\texttt{last_of}(n, \texttt{int}:\delta, \pi)) &= \{\{[t'_1, t'_2] \in \mathit{Int}(\mathcal{C}) \mid (d_1, d_2) \in M_\delta \wedge [t_1, t_2] \in M_\pi \\
&\qquad \wedge\ t_1 \leq_{\mathcal{C}} t'_1 = t_2 \ominus (n \cdot d_2) \oplus (0,1) \\
&\qquad \wedge\ t'_2 = t_2 \ominus ((n-1) \cdot d_1)\} \mid \\
&\qquad M_\delta \in \mathcal{I}_{DUR}(\delta) \wedge M_\pi \in \mathcal{I}_P(\pi)\}
\end{aligned}$$

6.4.9 Vereinigung und Schnitt (`set` und Listenbildung)

Wie schon innerhalb der Sprache L(`DURATION`) so steht auch in L(`POINT`) der `set`-Operator für die Disjunktion von zeitlichen Ausdrücken, also für eine semantische Vereinigung. Hinzu kommt für Ausdrücke der Kategorie `POINT` die Möglichkeit der Konjunktion von Ausdrücken. Auf der syntaktischen Ebene wird dies durch eine einfache Zusammenfassung der TEL-Ausdrücke in eine Liste dargestellt. Auf der semantischen Ebene entspricht sie dem Schnitt. Da Schnitt und Vereinigung auch über die Interpretation von Ausdrücken der Form `POINT*` beziehungsweise `set(POINT+)` hinaus von Bedeutung sein wird, werde ich beide Operationen zunächst über dem semantischen Bereich $\mathcal{P}(\mathcal{P}(\mathit{Int}(\mathcal{C})))$ definieren. Die Interpretation kann dann über diese semantischen Operationen sehr einfach festgelegt werden.

Die semantische Vereinigung wird in direkter Analogie zur semantischen Vereinigung über dem mit L(`DURATION`) assoziierten semantischen Bereich definiert.

Definition 6.23 *Seien $P_1, \ldots, P_n \in \mathcal{P}(\mathcal{P}(\mathit{Int}(\mathcal{C})))$. Dann wird die* semantische Vereinigung *dieser Mengen folgendermaßen definiert:*

$$P_1 \sqcup \cdots \sqcup P_n \ := \ \{M_1 \cup \cdots \cup M_n \mid M_1 \in P_1 \wedge \cdots \wedge M_n \in P_n\}$$

Zur Berechnung des semantischen Schnitts zweier Elemente aus $\mathcal{P}(\mathcal{P}(\mathit{Int}(\mathcal{C})))$ muß innerhalb jeder möglichen Zuordnung von Alternativmengen für jede Kombination von Intervallen der Schnitt gebildet werden. Die folgende Definition formalisiert dies für n Elemente.

Definition 6.24 *Seien $P_1, \ldots, P_n \in \mathcal{P}(\mathcal{P}(\mathit{Int}(\mathcal{C})))$. Dann wird der* semantische Schnitt *dieser Mengen definiert wie folgt:*

$$\begin{aligned}
P_1 \sqcap \cdots \sqcap P_n \ &:= \ \{\{(I_1 \cap \cdots \cap I_n) \in \mathit{Int}(\mathcal{C}) \mid I_1 \in M_1 \wedge \cdots \wedge I_n \in M_n\} \mid \\
&\qquad M_1 \in P_1 \wedge \cdots \wedge M_n \in P_n\}
\end{aligned}$$

Beachte, daß Intervalle über $\mathcal{C}$ wie Mengen geschnitten werden. Durch die an den Schnitt $I_1 \cap \cdots \cap I_n$ gestellte Bedingung der Zugehörigkeit zu $Int(\mathcal{C})$ werden all jene Ergebnisse der Schnittoperationen, die keine zulässigen Intervalle sind, aussortiert.

Die Interpretation des `set`-Operators entspricht wie gesagt gerade der semantischen Vereinigung über $\mathcal{P}(\mathcal{P}(Int(\mathcal{C})))$. Für $\pi_1, \ldots, \pi_n \in L(\texttt{POINT})$ gilt:

$$\mathcal{I}_P(\texttt{set([}\pi_1, \ldots, \pi_n\texttt{])}) = \mathcal{I}_P(\pi_1) \sqcup \cdots \sqcup \mathcal{I}_P(\pi_n)$$

Eine Liste von Ausdrücken der Kategoie `POINT` wird als der semantische Schnitt der Interpretationen der Einzelausdrücke interpretiert. Seien also $\pi_1, \ldots, \pi_n \in L(\texttt{POINT})$; dann gilt:

$$\mathcal{I}_P(\texttt{[}\pi_1, \ldots, \pi_n\texttt{]}) = \mathcal{I}_P(\pi_1) \sqcap \cdots \sqcap \mathcal{I}_P(\pi_n)$$

Der Fall der leeren Menge muß gesondert betrachtet werden. Werden keine Angaben zu einem Zeitpunkt gemacht, so bedeutet, dies, daß seine Lage in keiner Weise eingeschränkt wird. Also entspricht der TEL-Ausdruck `[]` dem natürlichsprachlichen Ausdruck "immer". Semantisch wird er also als das beidseitig unbegrenzte Intervall über $\mathcal{C}$ (welches laut Definition 6.6 identisch mit der Menge $\mathcal{C}$ ist) interpretiert.

$$\mathcal{I}_P(\texttt{[]}) = \{\{[-\infty_{\mathcal{C}}, \infty_{\mathcal{C}}]\}\}$$

6.4.10 Komplement (`not`)

Zur Berechnung der Negation eines Ausdrucks aus $L(\texttt{POINT})$ muß ein semantisches Komplement gebildet werden. Auch diese Operation werde ich zunächst allgemein definieren und dann zur Beschreibung der Semantik des `not`-Operators benutzen. Für jede Alternativmenge muß das Komplement eines Elements von $P \in \mathcal{P}(\mathcal{P}(Int(\mathcal{C})))$ gesondert berechnet werden. Ein Zeitpunkt liegt im semantischen Komplement einer solchen Alternativmenge, wenn er in keinem ihrer Intervalle, also entweder 'links' oder 'rechts' eines jeden Intervalles dieser Alternativmenge liegt.

Definition 6.25 *Es sei $P \in \mathcal{P}(\mathcal{P}(Int(\mathcal{C})))$. Dann wird das* semantische Komplement[10] *von P definiert wie folgt:*[11]

$$\begin{aligned}\overline{P} \;:=\; \{\{(I_1 \cap \cdots \cap I_n) \in Int(\mathcal{C}) \mid \forall i \in \{1, \ldots, n\} :\\ I_i \in \{[-\infty_{\mathcal{C}}, t_{i1} \ominus (0,1)], [t_{i2} \oplus (0,1), \infty_{\mathcal{C}}]\}\} \mid\\ \{[t_{11}, t_{12}], \ldots, [t_{n1}, t_{n2}]\} \in P\}\end{aligned}$$

Die Interpretation eines durch den `not`-Operator negierten Ausdrucks der Kategorie `POINT` ist gerade das semantische Komplement der Interpretation dieses Ausdrucks. Es ist also für $\pi \in L(\texttt{POINT})$

$$\mathcal{I}_P(\texttt{not(}\pi\texttt{)}) = \overline{\mathcal{I}_P(\pi)}$$

[10] In dieser Arbeit werde ich mit $\overline{P}$ durchgängig das semantische Komplement des Objekts P bezeichnen. Es handelt sich also nicht etwa um das (herkömmliche) Komplement einer Menge.

[11] Beachte, daß sowohl $[-\infty_{\mathcal{C}}, -\infty_{\mathcal{C}} \ominus (0,1)]$ als auch $[\infty_{\mathcal{C}} \oplus (0,1), \infty_{\mathcal{C}}]$ nicht zu $Int(\mathcal{C})$ gehören.

6.4.11 Quantifikation (QUANTIFIED)

Als existentielle Quantoren stehen in der TEMPORAL EXPRESSION LANGUAGE einfache natürliche Zahlen zur Verfügung. Um etwa auszudrücken, daß man "an zwei Montagen im Mai 1998" Zeit hat – ohne sich hierbei festzulegen, um *welche* Tage es sich konkret handelt – kann in L(POINT) der Ausdruck `quantified(2,[dow:mon,month:may,year:1998])` gebildet werden. Da der Mai des Jahres 1998 insgesamt vier Montage hat, gibt es $\binom{4}{2} = 6$ Möglichkeiten, welche zwei Montage tatsächlich gemeint sein könnten. Diese Möglichkeiten werden semantisch als Interpretationsalternativen dargestellt. Zu jeder Interpretationsalternative der Interpretation des zweiten Arguments des `quantified`-Ausdrucks werden also Alternativmengen gebildet, die jeweils Teilmengen der ersten sind und gerade die durch das erste Argument spezifizierte Mächtigkeit haben.

Wird der existentielle Quantor negiert, so bedeutet dies, daß all jene Alternativmengen, die eine geringere als die angegebene Mächtigkeit haben, zu generieren sind. "Nicht an zwei Montagen" wird also als "an einem oder an keinem Montag" interpretiert. Eine andere Möglichkeit bestünde darin, all jene Mengen zu bilden, deren Mächtigkeit von der angegebenen verschieden ist. Das würde bedeuten, daß "nicht an zwei Montagen" mit "entweder an weniger als zwei Montagen oder an mehr als zwei Montagen" gleichgesetzt würde. Dieses Vorgehen erscheint mir jedoch aus pragmatischer Sicht als das weniger sinnvolle.

Es wird also für $n \in L(\mathtt{NUMBER})$ sowie $\pi \in L(\mathtt{POINT}) \cup L(\mathtt{GENERIC})$ definiert:

$$\begin{aligned}\mathcal{I}_P(\mathtt{quantified}(n, \pi)) &= \{M' \subseteq M \mid M \in \mathcal{I}_P(\pi) \wedge \#M' = n\} \\ \mathcal{I}_P(\mathtt{quantified}(\mathtt{not}(n), \pi)) &= \{M' \subseteq M \mid M \in \mathcal{I}_P(\pi) \wedge \#M' < n\}\end{aligned}$$

Hieraus ergibt sich auch unmittelbar die Semantik des universellen Quantors `every`. Es genügt, in obigen Gleichungen n auf der linken Seite durch `every` und auf der rechten Seite durch $\#M$ zu ersetzen. Statt zu sagen, man habe an *allen* Zeitpunkten mit einer bestimmten Eigenschaft Zeit, könnte man schließlich auch angeben, man habe an gerade sovielen dieser Zeitpunkte Zeit, wie es Zeitpunkte mit dieser Eigenschaft gibt. Einige einfache Umformungen ergeben dann die folgende Interpretation.

Seien $n \in L(\mathtt{NUMBER})$ und $\pi \in L(\mathtt{POINT}) \cup L(\mathtt{GENERIC})$; dann gilt (beachte, daß $\subset$ die *echte* Teilmengenrelation bezeichnet):

$$\begin{aligned}\mathcal{I}_P(\mathtt{quantified}(\mathtt{every}, \pi)) &= \mathcal{I}_P(\pi) \\ \mathcal{I}_P(\mathtt{quantified}(\mathtt{not}(\mathtt{every}), \pi)) &= \{M' \subset M \mid M \in \mathcal{I}_P(\pi)\}\end{aligned}$$

Die Semantik von π ist also mit der von `quantified(every,`π`)` identisch. Dieses Ergebnis steht in Übereinstimmung mit der banalen Erkenntnis, daß die *Menge der Montage* identisch mit der *Menge aller Montage* ist.

Schließlich kommen als Quantoren noch Ausdrücke der Form `every:NUMBER` in Frage. In diesem Falle kann der Quantor nicht negiert werden. Durch `quantified(every:2,dom:sun` wird etwa der Ausdruck "jeder zweite Sonntag" dargestellt. Es ist allerdings nicht klar, bei welchem Sonntag die Zählung zu beginnen hat. Also besteht die Interpretation auch dieses Ausdrucks lediglich darin, die Menge der Zeitpunkte, die an einem Sonntag liegen, auf (in diesem Fall) zwei Alternativmengen zu verteilen (es handelt sich also quasi um eine Trennung der 'geraden' und der 'ungeraden' Sonntage'). Es sei daran erinnert, daß als zweites Argument nur Ausdrücke der Kategorie COUNTABLE1, also Ausdrücke mit abzählbarer Semantik (siehe Definition 6.22) zugelassen sind.

Seien also $n \in L(\texttt{NUMBER})$ und $\gamma \in L(\texttt{COUNTABLE1})$; es wird definiert:

$$\begin{aligned}\mathcal{I}_P(\texttt{quantified(every:}n,\gamma\texttt{))} \quad = \quad & \{\{I \in M_\gamma \mid \\ & \#\{I' \in M_\gamma \mid m(I') \leq_C m(I)\} \bmod n = i\} \mid \\ & M_\gamma \in \mathcal{I}_P(\gamma) \wedge i \in \{0,\ldots,n-1\}\}\end{aligned}$$

6.4.12 Alternativen (one_out_of)

Analog zur Definition der Semantik des one_out_of-Operators zur Aufzählung mehrerer Interpretationsalternativen in der Sprache $L(\texttt{DURATION})$ wird auch für Ausdrücke der Kategorie POINT eine einfache Vereinigung der Alternativmengen berechnet.

Sind $\pi_1,\ldots,\pi_n \in L(\texttt{POINT})$, so gilt:

$$\mathcal{I}_P(\texttt{one\_out\_of([}\pi_1,\ldots,\pi_n\texttt{]))} \quad = \quad \mathcal{I}_P(\pi_1) \cup \cdots \cup \mathcal{I}_P(\pi_n)$$

6.5 Ausdrücke der Kategorie DATE

6.5.1 Vorbemerkung zu Konjunktion und Negation bei Dauern

Sowohl $L(\texttt{POINT})$ als auch $L(\texttt{DURATION})$ verfügen über einen set-Operator und einen one_out_of-Operator zu Kodierung von Vereinigung respektive zur Darstellung unterschiedlicher Interpretationsalternativen. In $L(\texttt{POINT})$ kann darüber hinaus durch Listenbildung der Schnitt mehrerer Ausdrücke sowie durch not das Komplement eines Ausdrucks repräsentiert werden. Für Ausdrücke der Kategorie DURATION sind diese Möglichkeiten nicht vorgesehen. Dies läßt sich sowohl von der Warte der Semantik aus als auch über eine Korpusanalyse begründen.

Tatsächlich läßt sich für den für $L(\texttt{DURATION})$ in Abschnitt 6.3.2 gewählten semantischen Bereich überhaupt keine (der intendierten Semantik entsprechnde) Schnittoperation definieren. Das liegt letztlich daran, daß für Elemente aus $\mathcal{D}$, also für Monats-/Minuten-Paare kein allgemeingültiger Äquivalenzbegriff gefunden werden kann. Ob 31 Tage gleich einem Monat sind, ob also $(0, 44640) = (1, 0)$ gilt, kann nicht bestimmt werden. (Aus dem gleichen Grunde konnte übrigens in Definition 6.17 über $\mathcal{D}$ keine totale Ordnung angegeben werden.) Die Berechnung eines semantischen Schnitts läuft aber im wesentlichen auf die mehrfache Auswertung einer Äquivalenzrelation hinaus, kann also für DURATION-Ausdrücke nicht durchgeführt werden. Die Berechnung des Komplements eines einzigen Pseudo-Intervalls wäre noch möglich, wenn man für die Minuten-Werte in den Monats-/Minuten-Paare auch negative Zahlen zuließe, was durch folgende Formel verdeutlicht wird:

$$\overline{\{((n_{11}, n_{12}), (n_{21}, n_{22}))\}} = \{((0, -\infty), (n_{11}, n_{12} - 1)), ((n_{21}, n_{22} + 1), (\infty, \infty))\}$$

Um aber das Komplement einer Menge von Pseudo-Intervallen berechnen zu können, müßte der Schnitt der einzelnen Komplemente berechnet werden können (nach deMorgan), was, wie gesagt, nicht möglich ist. [12]

Das Fehlen von Konjunktion (also Schnitt) und Negation (also Komplement) in der Sprache L(`DURATION`) läßt sich auch über die Korpusanalyse beziehungsweise mit einfachen 'linguistischen' Argumenten begründen. Es wurde tatsächlich kein Beispiel gefunden, bei dem ein Sprecher eine Dauerangabe in Form einer Konjunktion mehrerer einzelner Angaben gemacht hätte. Werden mehr als eine Dauerangabe gemacht, so sind diese immer disjunktiv zu interpretieren. Der Grund hierfür liegt auf der Hand: der Sprecher hat zum Zeitpunkt seiner Äußerung bereits eine feste Vorstellung davon, an welcher Stelle im Kalender die jeweilige Dauer positioniert ist, das heißt, sie wird durch eine einzige Längenangabe (ob in Monaten oder Minuten spielt keine Rolle) aus seiner Sicht vollständig spezifiziert. Es wäre also unsinnig zu versuchen, sie noch weiter zu konkretisieren.[13] Bei der Beschreibung eines Zeitpunktes ist das anders: eine einzelne Äußerung, wie etwa "am Montag" oder "im November" spezifiziert den jeweiligen Zeitpunkt nur unvollständig, weshalb eine Konkretisierung durch die Konjunktion mit weiteren Angaben sinnvoll ist. Die Negation mußte in L(`POINT`) aufgenommen werden, um Äußerungen wie "in der zweiten Maiwoche, aber nicht am Mittwoch" darstellen zu können. Eine Negation *innerhalb* einer Sprache zur Beschreibung einer Dauer ist hingegen nicht nötig, da hier nicht mehrere (eventuell zu verneinende) Merkmale zusammengefaßt werden. Es ist also ausreichend, eine durch einen Ausdruck der Kategorie `DURATION` beschriebene Dauer gegebenenfalls einmal *außerhalb* dieser Sprache, und zwar in L(`DATE`), zu negieren. `DURATION`-Ausdrücke, die in Ausdrücken des Typs `POINT` als Argumente auftauchen, sollten ebenfalls nicht negiert werden können. Verneinung einer Dauer ist also sinnvoller Weise nur in Aussagen wie "Ich möchte mich nicht für drei Stunden treffen." zulässig.

Es müssen also zwei Arten des Gebrauchs der Sprache L(`DURATION`) unterschieden werden. Zum einen kommen `DURATION`-Ausdrücke innerhalb der Sprache L(`POINT`) als Argumente vor. Hier sind Negation und Konjunktion in jedem Falle unerwünscht und eine exakte Auswertung von Dauern beschreibenden Ausdrücken ist wichtig (um zum Beispiel sicherzustellen, daß "einen Monat nach dem dritten März" als "dritter April" interpretiert wird). In diesem Bereich kommen

[12] Eine andere Möglichkeit bestünde zwar in der direkten Berechnung; hierzu müßte aber wiederum eine totale Ordnungsrelation definiert sein.

[13] Eine Ausnahme stellt allerdings der Fall dar, in dem zunächst ein Bereich angegeben wird und danach eine spezielle Dauer aus diesem Bereich genannt wird, also zum Beispiel "für zwei bis drei Stunden, oder sagen wir: für drei Stunden". Solch ein Beispiel taucht im untersuchten Korpus jedoch nicht auf. Außerdem könnte man derartige Phänomene auch als Selbstkorrektur einstufen und den ersten Teil schon vor der Generierung eines TEL-Ausdrucks eliminieren.

insbesondere sowohl Angaben in der Grundeinheit Minuten als auch solche in der Grundeinheit Monate vor. Die Unvergleichbarkeit dieser beiden Einheiten hat letzten Endes dazu geführt, daß die Formulierung einer semantischen Schnittoperation nicht möglich ist.

Zum zweiten können Ausdrücke der Kategorie DURATION durch das Label for eingeleitet in $L(\mathtt{DATE})$ auftreten. Hier sollte die Möglichkeit zur Negation gegeben sein. Die Konjunktion ist allerdings auch hier kaum sinnvoll. Allerdings muß für den Vergleich zweier Terminbeschreibungen der semantische Schnitt berechnet werden können. Um dies auch auf syntaktischer Ebene (durch Verkettung der beiden Ausdrücke) darstellen zu können, bietet es sich an, die Konjunktion trotzdem zuzulassen, zumal ein solches Vorgehen die Definition der Syntax vereinfacht (hat). Es kann davon ausgegangen werden, daß die Dauer eines Termins stets in Minuten, Stunden, Tagen oder höchstens in Wochen, also ausschließlich in in die Grundeinheit Minuten umwandelbaren Einheiten angegeben sein wird. Würde die TEL-Grammatik die Einheiten months und years gar nicht erst zulassen, so ergäben sich auch keinerlei Schwierigkeiten bei der Berechnung eines semantischen Schnitts zweier Dauer-Ausdrücke.

Eine mögliche Lösung der aufgeworfenen Probleme bestünde also darin, eine zweite Sprache zur Beschreibung von Zeitdauern zu definieren, die nur solche Dauer-Ausdrücke, die nach dem for-Label vorkommen können, enthält. Der einzige Unterschied zur tatsächlich definierten Sprache bestünde in der Streichung von months und years aus der Kategorie UNIT. Außerdem könnten Konjunktion und Negation auch innerhalb einer solchen Sprache zugelassen werden. Dieses Vorgehen scheint (formal) recht umständlich. Es würde unter anderem eine vollständige Definition der Interpretation dieser neuen Sprache erforderlich machen. Stattdessen werde ich weiter unten für Ausdrücke der Kategorie DURATION, die nach einem for-Label stehen, lediglich eine etwas vereinfachte Semantik definieren, die sich direkt aus der bereits angegebenen Semantik ableiten läßt. In dieser vereinfachten Semantik werden Längenangaben in Monaten gemäß den genannten Besonderheiten von Dauerangaben bei Terminbeschreibungen keine Rolle spielen, wodurch die Definition von Komplement und Schnitt möglich wird.

6.5.2 Ein Vereinfachter Semantischer Bereich für $L(\mathtt{DURATION})$

Im letzten Abschnitt wurde erläutert, daß für Ausdrücke der Kategorie DURATION, die nach dem Label for stehen, eine einfachere Interpretation als die in Teil 6.3 dieses Kapitels definierte ausreicht. Es werden nur noch Angaben in Minuten gemacht. Statt der Menge der Monats-/Minuten-Paare $\mathcal{D}$ kann nun also mit der Menge der natürlichen Zahlen (einschließlich der 0) gearbeitet werden. Über den natürlichen Zahlen können selbstverständlich Intervalle definiert werden, weshalb Pseudo-Intervalle hier unnötig werden. Mit $Int(\mathbb{N}_0)$ werde die Menge dieser Intervalle bezeichnet. Der (vereinfachte) semantische Bereich für in diesem speziellen Kontext auftretende Ausdrücke aus $L(\mathtt{DURATION})$ ist also $\mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))$.

Im folgenden soll die Funktion σ zur Abbildung des einen Bereichs auf den anderen definiert werden. Die einfachste Möglichkeit wäre sicherlich, die erste Stelle aus den Paaren aus $\mathcal{D}$ einfach zu streichen (sie sollte ohnehin stets 0 sein). Eine weitere Möglichkeit besteht darin, Monats-/Minuten-Paare in Minuten umzurechnen soweit dies eben möglich ist. Eine Pseudo-Intervall wird dabei auf ein echtes Intervall abgebildet. Hierbei wird die linke Intervallgrenze nach unten abgeschätzt, die rechte nach oben. Diese Abschätzung erfolgt analog zu der der in Definition 6.17 angegebenen Ordungsrelation über $\mathcal{D}$. Da der Wert für die Anzahl der Monate stets 0 sein sollte, handelt es sich in der Praxis jedoch *nicht* um eine Abschätzung, sondern führt zu dem gleichen Ergebnis wie das Streichen der ersten Stelle.

Definition 6.26 *Die Funktion* $\sigma : \mathcal{P}(\mathcal{P}(Int^*(\mathcal{D}))) \to \mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))$ *dient zur Abbildung des allgemeinen semantischen Bereichs von* $L(\texttt{DURATION})$ *auf den speziellen semantischen Bereich von* $L(\texttt{DURATION})$ *und wird definiert wie folgt:*

$$\begin{aligned} \sigma : D \quad \longmapsto \quad & \{\{[28 \cdot 1440 \cdot n_{11} + n_{12}, 31 \cdot 1440 \cdot n_{21} + n_{22}] \mid \\ & ((n_{11}, n_{12}), (n_{21}, n_{22})) \in M\} \mid M \in D\} \end{aligned}$$

Für den vereinfachten semantischen Bereich $\mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))$ können nun neben der Vereinigung auch Schnitt und Komplement berechnet werden. Die Definitionen dieser Operationen werden im folgenden angegeben. Vergleiche hierzu auch die entsprechenden Definitionen für die Sprache $L(\texttt{POINT})$. Die einfache Vereinigung von Interpretationsalternativen (`one_out_of`) ist auch hier natürlich wieder eine herkömmliche Vereinigung von Mengen.

Definition 6.27 *Seien* $D_1, \ldots, D_n \in \mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))$. *Dann wird die* semantische Vereinigung *dieser Mengen definiert wie folgt:*

$$D_1 \sqcup \cdots \sqcup D_n \quad := \quad \{M_1 \cup \cdots \cup M_n \mid M_1 \in D_1 \wedge \cdots \wedge M_n \in D_n\}$$

Definition 6.28 *Seien* $D_1, \ldots, D_n \in \mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))$. *Dann wird der* semantische Schnitt *dieser Mengen definiert wie folgt:*

$$\begin{aligned} D_1 \sqcap \cdots \sqcap D_n \quad := \quad & \{\{(I_1 \cap \cdots \cap I_n) \in Int(\mathbb{N}_0) \mid I_1 \in M_1 \wedge \cdots \wedge I_n \in M_n\} \mid \\ & M_1 \in D_1 \wedge \cdots \wedge M_n \in D_n\} \end{aligned}$$

Mit $I_1 \cap \cdots \cap I_n$ wird der Schnitt der den Intervallen I_1 bis I_n entsprechenden Mengen bezeichnet. Nur wenn dieser ebenfalls ein gültiges Intervall ist, wird er Teil des definierten semantischen Schnitts.

Definition 6.29 *Sei* $D \in \mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))$. *Dann wird das* semantische Komplement *von* D *definiert als folgende Menge:*

$$\begin{aligned} \overline{D} \quad := \quad & \{\{(I_1 \cap \cdots \cap I_n) \in Int(\mathbb{N}_0) \mid \forall i \in \{1, \ldots, n\} : \\ & I_i \in \{[0, \max\{0, d_{i1} - 1\}], [d_{i2} + 1, \infty]\}\} \mid \\ & \{[d_{11}, d_{12}], \ldots, [d_{n1}, d_{n2}]\} \in D\} \end{aligned}$$

6.5.3 Motivation

Ein Termin kann durch die Angabe zweier seiner Merkmale Anfangspunkt, Endpunkt und Dauer eindeutig spezifiziert werden. Einen eindeutig bestimmbaren Termin würde man also am einfachsten als ein Paar aus einem Zeitpunkt für den Anfang und einen für das Ende repräsentiern. In der Praxis wird man jedoch in den meisten Fällen unterspezifizierte Termine darzustellen haben. Für die Sprachen L(`DURATION`) und L(`POINT`) wurden unterspezifizierte Ausdrücke als die Menge aller Möglichkeiten interpretiert. Schränkt man dabei vorab $\mathcal{D}$ beziehungsweise $\mathcal{C}$ sinnvoll ein, so bleibt die Größe der interpretierten Ausdrücke in einem begrenzten Rahmen, insbesondere da durch die Verwendung von Intervallen eine recht kompakte Darstellung möglich ist. Sollen hingegen Termine mit Anfangs- und Endpunkt dargestellt werden, so kann dies nicht über Intervalle geschehen. Es können lediglich je ein Intervall für den Anfangspunkt und eines für den Endpunkt angegeben werden. Wird jedoch darüber hinaus eine bestimmte Dauer spezifiziert, so sind nicht alle Kombinationen aus den Zeitpunkten aus diesen Intervallen gültig. Eine Semantik, aus der die Menge der möglichen Termine direkt ablesbar ist, wäre also nur ohne die Verwendung von Intervallen machbar und daher von enormer Komplexität. Es scheint aber angemessen, von der Interpretation von Ausdrücken der Kategorie `DATE` zu fordern, daß sie von vergleichbarer Komplexität wie die Interpretationen ihrer Teilausdrücke aus L(`DURATION`) beziehungsweise L(`POINT`) ist. Somit scheidet die einfache Aufzählung aller möglicherweise gemeinten Termine als semantischer Bereich aus.

Die andere sich anbietende Möglichkeit besteht darin, sich unmittelbar an der Syntax von L(`DATE`) zu orientieren, und jeweils eine Gruppe von Terminen über die sie spezifizierenden Merkmale wie Anfangs- und Endpunkt beziehungsweise Dauer einzugrenzen. Ob ein bestimmter Termin dann tatsächlich zu einer solchen Gruppe gehört, muß dann immer noch im Einzelfall überprüft werden. Ein eindeutig spezifizierter Termin wird sich aber auch aus einer solchen semantischen Darstellung direkt ablesen lassen. Ich werde diesem Ansatz folgen.

Die Informationen zur Eingrenzung eines Termins stehen in L(`DATE`) hinter den Labeln `from`, `to`, `during` und `for`. Durch `for` wird die Dauer des Termins bestimmt, durch `from` sein Anfangspunkt und durch `to` sein Endpunkt. Außerdem kann durch einen `POINT`-Ausdruck nach dem Label `during` eine Aussage über die zeitliche Lage des gesamten Termins, also aller seiner Zeitpunkte inklusive des Anfangs- und des Endpunktes gemacht werden. Beachte, daß in diesem Falle mit der Interpretation eines solchen Ausdrucks der Kategorie `POINT` nicht mehr *ein* unterspezifizierter Zeitpunkt sondern vielmehr eine Menge zusammenhängender (möglicherweise ebenfalls unterspezifizierter) Zeitpunkte gemeint ist. Die Berechnungen sind natürlich die gleichen, nur die jeweiligen Auslegungen differieren leicht.

Im letzten Kapitel wurde gezeigt, wie durch das `interval`-Label eingeleitete Ausdrücke auf Ausdrücke mit den Labeln `from`, `to` beziehungsweise `during` reduziert werden können. Eine entsprechende Transformation von durch `during` eingeleiteten Ausdrücken auf solche mit den Labeln `from` und `to` ist übrigens nicht möglich.

Die Information, daß alle Zeitpunkte, die zwischen Anfangs- und Endpunkt liegen einer bestimmten Menge von Zeitpunkten angehören müssen, läßt sich nicht vollständig durch separate Angaben zu Anfangs- und Endpunkt selbst kodieren. Der zu definierende semantische Bereich wird also drei Teilbereiche zur Lage des Termins (entsprend den Labeln `from`, `to` und `during`) sowie einen Teilbereich zur Dauer des Termins (`for`) enthalten.

Angaben, die nach dem Label `statement` gemacht werden, beziehen sich nicht direkt auf den zu verhandelnden Termin, müssen also auch nicht als `DATE`-Ausdrücke interpretiert werden. Sie werden allerdings unter Umständen für die Auflösung von Anaphern benötigt. Ein Statement ist *wahr*, wenn die Interpretation des entsprechenden `POINT`-Ausdruckes wenigstens einen Zeitpunkt enthält (also nicht bloß leere Mengen).

6.5.4 Semantischer Bereich und die Interpretationsfunktion $\mathcal{I}$

Der semantische Bereich zur Interpretation von Ausdrücken der Sprache $L(\mathtt{DATE})$ wird im wesentlichen aus Quadrupeln bestehen, durch die Angaben zum Terminbeginn, zu dessen Ende, zur Lage des gesamten Termins sowie zu seiner Dauer zusammengefaßt werden. An den ersten drei Stellen werden Objekte aus dem für $L(\mathtt{POINT})$ definierten Bereich stehen; die letzte Stelle wird die (vereinfachte) Interpretation eines `DURATION`-Ausdrucks enthalten. Um die über $L(\mathtt{DATE})$ definierten Operatoren `set` und `one_out_of` interpretieren zu können, wird auch dieser semantische Bereich wieder eine Menge von Mengen solcher Quadrupel sein. Die zugehörige Interpretationsfunktion heißt $\mathcal{I}$.

$$\mathcal{I} : L(\mathtt{DATE}) \to \mathcal{P}(\mathcal{P}(\ \mathcal{P}(\mathcal{P}(\mathit{Int}(\mathcal{C})))^3 \times \mathcal{P}(\mathcal{P}(\mathit{Int}(\mathbb{N}_0)))\ ))$$

Sei $A \in \mathcal{P}(\mathcal{P}(\mathcal{P}(\mathcal{P}(\mathit{Int}(\mathcal{C})))^3 \times \mathcal{P}(\mathcal{P}(\mathit{Int}(\mathbb{N}_0)))))$. Außerdem seien $t_1, t_2 \in \mathcal{C}$ und $d \in \mathbb{N}_0$. Wenn der Abstand zwischen t_1 und t_2 gerade d ist, so gibt es eine Termin, der zum Zeitpunkt t_1 beginnt, zum Zeitpunkt t_2 endet und der eine Dauer von d Minuten hat. Das Tripel (t_1, t_2, d) stellt dann eine zulässige, explizite Terminangabe dar. Dieser Termin gehört der durch A beschriebenen Menge von Terminen an, wenn es in A eine Alternativmenge gibt, in der es wiederum ein Quadrupel $Q = (P_f, P_t, P_d, D)$ gibt, für das die folgenden vier Bedingungen erfüllt sind:

- In P_f existiert eine Alternativmenge, die ein Zeitpunkt-Intervall enthält, das wiederum t_1 enthält.
- In P_t existiert eine Alternativmenge, die ein Zeitpunkt-Intervall enthält, das wiederum t_2 enthält.
- In P_d existiert eine Alternativmenge, deren Zeitpunkte-Intervalle zusammen alle Zeitpunkte von t_1 bis t_2 (einschließlich) enthalten.
- In D existiert eine Alternativmenge, die ein Intervall enthält, das wiederum d enthält.

Diese Beziehung zwischen Quadrupeln und den durch sie umschriebenen Terminen wird durch die folgende Definition formalisiert.

Definition 6.30 *Sei* $Q = (P_f, P_t, P_d, D) \in \mathcal{P}(\mathcal{P}(Int(\mathcal{C})))^3 \times \mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))$ *ein Quadrupel zur Darstellung einer Terminbeschreibung und sei* $(t_1, t_2, d) \in \mathcal{C}^2 \times \mathbb{N}_0$ *ein durch seinen Anfangspunkt* t_1*, seinen Endpunkt* t_2 *und seine Dauer* d *in Minuten spezifizierter Termin. Durch* $Q \vdash_{DATE} (t_1, t_2, d)$ *werde ausgedrückt, daß der Termin* (t_1, t_2, d) *zulässig ist und durch* Q *umschrieben wird. Diese Relation wird definiert wie folgt:*

$$
\begin{aligned}
Q \vdash_{DATE} (t_1, t_2, d) \quad :\Longleftrightarrow \quad & \langle t_1, t_2 \rangle_{\mathcal{C}} = d \;\wedge \\
& \exists\, M_f \in P_f : \exists\, I_f \in M_f : t_1 \in I_f \;\wedge \\
& \exists\, M_t \in P_t : \exists\, I_t \in M_t : t_2 \in I_t \;\wedge \\
& \exists\, M_d \in P_d : \forall\, t \in [t_1, t_2] : \exists\, I_d \in M_d : t \in I_d \;\wedge \\
& \exists\, M_D \in D : \exists\, I_D \in M_D : d \in I_D
\end{aligned}
$$

Es sei darauf hingewiesen, daß die $\vdash_{DATE}$ entsprechende Relation für den Bereich der Beschreibung von Zeitpunkten die simple Inklusion von Elementen aus $\mathcal{C}$ in Intervallen über $\mathcal{C}$ ist. Entsprechendes gilt auch für den vereinfachten semantischen Bereich für Dauerangaben.

6.5.5 Lage eines Termins (Label `from`, `to` und `during`)

Über die Label `from`, `to` und `during` kann die Lage eines Termins beschrieben werden. Die Interpretation solcher Ausdrücke besteht im wesentlichen darin, die Interpretation des jeweiligen Ausdrucks der Kategorie `POINT` an die richtige Stelle in einem Quadrupel zu schreiben. An die übrigen drei Positionen des Quadrupels muß die Interpretation eines Ausdrucks geschrieben werden, der keinerlei Einschränkungen macht. Für $L(\texttt{POINT})$ ist dies die leere Liste. Für $L(\texttt{DURATION})$ existiert ein solcher Ausdruck auf syntaktischer Ebene zwar nicht, im vereinfachten semantischen Bereich für Dauern gibt es hierfür aber den Wert $\{\{[0, \infty]\}\}$.

Für $\pi \in L(\texttt{POINT})$ wird definiert:

$$
\begin{aligned}
\mathcal{I}(\texttt{from:}\pi) &= \{\{(\mathcal{I}_P(\pi), \mathcal{I}_P(\texttt{[]}), \mathcal{I}_P(\texttt{[]}), \{\{[0, \infty]\}\})\}\} \\
\mathcal{I}(\texttt{to:}\pi) &= \{\{(\mathcal{I}_P(\texttt{[]}), \mathcal{I}_P(\pi), \mathcal{I}_P(\texttt{[]}), \{\{[0, \infty]\}\})\}\} \\
\mathcal{I}(\texttt{during:}\pi) &= \{\{(\mathcal{I}_P(\texttt{[]}), \mathcal{I}_P(\texttt{[]}), \mathcal{I}_P(\pi), \{\{[0, \infty]\}\})\}\}
\end{aligned}
$$

Statt $\mathcal{I}_P(\texttt{[]})$ könnte man natürlich auch direkt $\{\{[-\infty_{\mathcal{C}}, \infty_{\mathcal{C}}]\}\}$ schreiben.

6.5.6 Dauer eines Termins (Label `for`)

über das Label `for` wird die Dauer eines Termins beschrieben. Die Interpretation des jeweiligen Ausdrucks aus $L(\texttt{DURATION})$ muß zunächst mit Hilfe der Funktion σ vereinfacht, das heißt auf den vereinfachten semantischen Bereich für Zeitdauern abgebildet werden. Der Funktionswert wird dann an die letzte Stelle eines

Quadrupels geschrieben, deren übrige Stellen wieder mit dem allgemeinen Wert $\mathcal{I}_P(\texttt{[]})$ belegt werden.

Sei $\delta \in L(\texttt{DURATION})$; dann gilt:

$$\mathcal{I}(\texttt{for:}\delta) \quad = \quad \{\{(\mathcal{I}_P(\texttt{[]}), \mathcal{I}_P(\texttt{[]}), \mathcal{I}_P(\texttt{[]}), \sigma(\mathcal{I}_{DUR}(\delta)))\}\}$$

6.5.7 Vereinigung und Schnitt (set und Listenbildung)

Mit Hilfe des `set`-Operators können Ausdrück der Kategorie `DATE` disjunktiv verbunden werden. Auf semantischer Ebene entspricht dies einer Vereinigung. Werden mehrere Ausdrücke in einer Liste zusammengefaßt, so steht dies für eine Konjunktion, beziehungsweise für einen semantischen Schnitt. Diese beiden Operationen werden im folgenden auf dem für $L(\texttt{DATE})$ festgelegten semantischen Bereich definiert.

Definition 6.31 *Seien* $A_1 \ldots, A_n \in \mathcal{P}(\mathcal{P}(\mathcal{P}(\mathcal{P}(Int(\mathcal{C})))^3 \times \mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))))$. *Dann wird die* semantische Vereinigung *dieser Mengen definiert als folgende Menge:*

$$A_1 \sqcup \cdots \sqcup A_n \quad := \quad \{M_1 \cup \cdots \cup M_n \mid M_1 \in A_1 \wedge \cdots \wedge M_n \in A_n\}$$

Definition 6.32 *Seien* $A_1 \ldots, A_n \in \mathcal{P}(\mathcal{P}(\mathcal{P}(\mathcal{P}(Int(\mathcal{C})))^3 \times \mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))))$. *Dann wird der* semantische Schnitt *dieser Mengen folgendermaßen definiert:*

$$\begin{aligned} A_1 \sqcap \cdots \sqcap A_n \ := \ & \{\{(P_1^f \sqcap \cdots \sqcap P_n^f, P_1^t \sqcap \cdots \sqcap P_n^t, P_1^d \sqcap \cdots \sqcap P_n^d, D_1 \sqcap \cdots \sqcap D_n) \mid \\ & (P_1^f, P_1^t, P_1^d, D_1) \in M_1 \wedge \cdots \wedge (P_n^f, P_n^t, P_n^d, D_n) \in M_n\} \mid \\ & M_1 \in A_1 \wedge \cdots \wedge M_n \in A_n\} \end{aligned}$$

Diese Definitionen können nun unmittelbar zur Beschreibung der Interpretationen von Disjunktion und Konjunktion über $L(\texttt{DATE})$ angewendet werden.

Seien $\alpha_1, \ldots, \alpha_n \in L(\texttt{DATE})$; dann gilt:

$$\begin{aligned} \mathcal{I}(\texttt{set([}\alpha_1, \ldots, \alpha_n\texttt{])}) &= \mathcal{I}(\alpha_1) \sqcup \cdots \sqcup \mathcal{I}(\alpha_n) \\ \mathcal{I}(\texttt{[}\alpha_1, \ldots, \alpha_n\texttt{]}) &= \mathcal{I}(\alpha_1) \sqcap \cdots \sqcap \mathcal{I}(\alpha_n) \end{aligned}$$

Die Semantik der leeren Menge muß noch gesondert definiert werden.

$$\mathcal{I}(\texttt{[]}) \quad = \quad \{\{(\mathcal{I}_P(\texttt{[]}), \mathcal{I}_P(\texttt{[]}), \mathcal{I}_P(\texttt{[]}), \{\{[0, \infty]\}\})\}\}$$

6.5.8 Komplement (not)

Bei der Interpretation negierter DATE-Ausdrücke werden aus jedem Quadrupel vier neue Quadrupel. Da ein Quadrupel für die Konjunktion vierer Aussagen steht, ist seine Negation die Disjunktion der Negationen der vier Aussagen. Es wird also jeweils an einer Position das semantische Komplement gebildet und an die anderen Stellen wird das beidseitig unbegrenzte Intervall über dem jeweiligen Bereich geschrieben.

Definition 6.33 *Es sei* $A \in \mathcal{P}(\mathcal{P}(\mathcal{P}(\mathcal{P}(Int(\mathcal{C})))^3 \times \mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))))$. *Dann wird das* semantische Komplement *von* A *definiert wie folgt:*

$$\begin{aligned}\overline{A} \;:=\; \{\{ & (\overline{P_f}, \mathcal{I}_P(\texttt{[]}), \mathcal{I}_P(\texttt{[]}), \{\{[0,\infty]\}\}),\\ & (\mathcal{I}_P(\texttt{[]}), \overline{P_t}, \mathcal{I}_P(\texttt{[]}), \{\{[0,\infty]\}\}),\\ & (\mathcal{I}_P(\texttt{[]}), \mathcal{I}_P(\texttt{[]}), \overline{P_d}, \{\{[0,\infty]\}\}),\\ & (\mathcal{I}_P(\texttt{[]}), \mathcal{I}_P(\texttt{[]}), \mathcal{I}_P(\texttt{[]}), \overline{D}) \mid\\ & (P_f, P_t, P_d, D) \in M\} \mid M \in A\}\end{aligned}$$

Für $\alpha \in L(\texttt{DATE})$ gilt:

$$\mathcal{I}(\texttt{not(}\alpha\texttt{)}) \;=\; \overline{\mathcal{I}(\alpha)}$$

Beachte, daß das Komplement von $\mathcal{I}_P(\texttt{[]})$ ("immer") $\{\{\}\}$ ("nie") ist. Entsprechendes gilt auch für $\{\{[0,\infty]\}\}$. Macht ein Ausdruck also zu einem der Merkmale Terminbeginn, Terminende, Terminverlauf oder Termindauer keine Aussage, so steht im entsprechenden Quadrupel bei der Komplementbildung an der jeweiligen Position $\{\{\}\}$. Es gibt also keinen gültigen Termin, der durch solch ein Quadrupel beschrieben würde. Diese Beobachtung belegt, daß beispielsweise für $\pi \in L(\texttt{POINT})$ die Interpretationen der Ausdrücke `not(from:`π`)` und `from:not(`π`)` semantisch äquivalent sind.

6.5.9 Alternativen (one_out_of) und Label-Listen

Um eine Menge durch den one_out_of-Operator zusammengefaßter Alternativen zu interpretieren, müssen wieder die Interpretationen der Einzelausdrücke vereinigt werden.

Seien also $\alpha_1, \ldots, \alpha_n \in L(\texttt{DATE})$. Es wird definiert:

$$\mathcal{I}(\texttt{one\_out\_of([}\alpha_1, \ldots, \alpha_n\texttt{])}) \;=\; \mathcal{I}(\alpha_1) \cup \cdots \cup \mathcal{I}(\alpha_n)$$

Die TEL-Grammatik für die Sprache $L(\texttt{DATE})$ gestattet das Zusammenfassen mehrerer Interpretationsalternativen, die sich nur durch das gewählte Label unterscheiden, durch die Verwendung von Label-Listen. Dies gilt nur für Label zur Eingrenzung der Lage eines Termins. Durch das statement-Label eingeleitete Ausdrücke werden bei der Interpretation gegebenenfalls ignoriert. Deklarativ besteht kein Unterschied zu der gerade definierten Semantik des one_out_of-Operators. Bei der algorithmischen Verwirklichung der Auswertung der Interpretationsfunktion $\mathcal{I}$ erlaubt diese syntaktische Variante allerdings ein effizienteres

Vorgehen, bei dem für den jeweiligen POINT-Ausdruck π der Funktionswert $\mathcal{I}_P(\pi)$ nur einmal berechnet werden muß.

Sind $\ell_1, \ldots, \ell_n \in \{\texttt{from}, \texttt{to}, \texttt{during}\}$ und ist $\pi \in L(\texttt{POINT})$, so gilt:

$$\mathcal{I}(\texttt{[}\ell_1, \ldots, \ell_n\texttt{]:}\pi) \quad = \quad \mathcal{I}(\texttt{one\_out\_of([}\ell_1 : \pi, \ldots, \ell_n : \pi\texttt{]))}$$

6.6 Zusammenfassung

In diesem Kapitel wurde die formale Semantik der in Kapitel 4 eingeführten TEMPORAL EXPRESSION LANGUAGE beschrieben. Nach der im letzten Kapitel vorgenommenen Reduktion der Sprache genügte es, die Semantik lediglich für die sogenannte Kernsprache, wie sie in Anhang B.3 wiedergegeben ist, anzugeben. Im ersten Teil des Kapitels wurden zunächst unabhängig vom eigentlichen Ziel der Interpretation von TEL-Ausdrücken einige grundsätzliche Anmerkungen zur Semantik des Kalenders gemacht. Danach wurde für jede der drei Subsprachen L(DURATION), L(POINT) und L(DATE) zunächst ein geeigneter semantischer Bereich und sodann eine passende Interpretationsfunktion zur Abbildung der jeweiligen Sprache auf diesen Breich definiert.

Allen drei Sprachen ist gemein, daß möglicherweise zu berücksichtigende unterschiedliche Interpretationsalternativen in einer Menge zusammengefaßt werden. Im Falle der Ausdrücke der Kategorie DURATION besteht jede dieser Mengen aus einer Reihe von Pseudo-Intervallen über der Menge der Paare aus Monats- und Minutenangabe (siehe Definition 6.10 und Definition 6.18). Jedes Pseudo-Intervall gibt hierbei einen Bereich von in Frage kommenden Zeitdauern an; die Menge dieser Pseudo-Intervalle steht für die semantische Vereinigung der assoziierten Bereiche. Die Interpretation eines DURATION-Ausdrucks beinhaltet also die Aussage, daß eine der Alternativmengen der vom Sprecher intendierten Semantik entspricht und daß die durch seine Äußerung eingegrenzte Menge der Zeitdauern gerade die Menge der Zeitdauern ist, die durch diese Alternativmenge beschrieben wird. Entsprechendes gilt auch für Ausdrücke der Kategorie POINT. Ein solcher Ausdruck wird als eine Menge von Alternativmengen von Intervallen über der Menge der Zeitpunkte im Kalender (siehe Definition 6.2 und Definition 6.6) interpretiert. Auch hier entspricht eine der Interpretationsalternativen der tatsächlichen Semantik. In dieser Menge wiederum entspricht die Vereinigung der Intervalle gerade der Menge von Zeitpunkten, die durch die Äußerung des Sprechers eingegrenzt wird. Ausdrücke der Kategorie DATE werden interpretiert, indem die nach den Labeln from, to, during sowie for gemachten Angaben zu Quadrupeln zusammengefaßt werden. Diese Quadrupel sind wiederum Elemente von Alternativmengen. Aus Komplexitätsgründen handelt es sich bei der hier gewählten Art der Interpretation nicht um eine explizite Aufzählung aller möglicherweise gemeinten Termine. Trotzdem können Termine wenigstens indirekt abgeleitet werden (siehe Definition 6.30). Für die Dauerangabe eines Termins konnte ein vereinfachter semantischer Bereich der Sprache L(DURATION) definiert werden (siehe Abschnitt 6.5.2).

Im folgenden Kapitel werden über den gerade eingeführten semantischen Bereichen verschiedenen Inferenzoperationen erklärt werden, mit deren Hilfe interpretierte TEL-Ausdrücke miteinander verglichen werden können. Hierbei werden insbesondere semantischer Schnitt und semantisches Komplement, wie sie in diesem Kapitel für die unterschiedlichen semantischen Bereiche definiert wurden, eine zentrale Rolle spielen.

Kapitel 7

Inferenz

Die Zeit entlarvt den Bösen.
Euripides

Nachdem im letzten Kapitel die Semantik von Ausdrücken der Temporal Expression Language beschrieben wurde, soll nun gezeigt werden, wie anhand dieser Semantik einige Inferenzen auf zeitlichen Ausdrücken definiert werden können.

Zu diesen Inferenzen gehören das Erkennen von inkonsistenten Ausdrücken und die Definition von Äquivalenzrelationen über den diversen semantischen Bereichen. Außerdem wird definiert werden, was es bedeutet, daß ein Ausdruck 'spezieller' oder 'genereller' als ein anderer ist. Diese Definitionen beziehen sich zunächst einmal auf die semantischen Abbildungen von TEL-Ausdrücken, die entsprechenden Begriffe können aber natürlich auch auf die Ausdrücke selbst angewandt werden. Des weiteren wird der Begriff der semantischen Differenz eingeführt werden. Auch dieses Kapitel schließt mit einer kurzen Zusammenfassung.

7.1 Inkonsistenz

Nicht jeder TEL-Ausdruck, der syntaktisch korrekt ist, ist auch semantisch konsistent. Einfache Beispiele hierfür sind `[dom:31,month:apr]` oder `tod:3:60`. Im folgenden wird definiert werden, wann ein Ausdruck aus $L(\texttt{DURATION})$, $L(\texttt{POINT})$ oder $L(\texttt{DATE})$ inkonsistent ist.

Das Erkennen einer Inkonsistenz kann etwa dazu dienen, Fehler, die im System schon vor der Generierung eines TEL-Ausdrucks aufgetreten sind, als solche zu identifizieren. Beim Versuch, einen Dialogverlauf nachzuvollziehen, wird es außerdem mitunter notwendig sein zu überprüfen, ob zwei Zeitausdrücke miteinander vereinbar sind. Dies ist genau dann der Fall, wenn ihr semantischer Schnitt nicht inkonsistent ist.

7.1.1 Dauer

Ob ein Ausdruck der Kategorie `DURATION` inkonsistent ist, läßt sich unmittelbar an seiner semantischen Interpretation ablesen. Ein solcher Ausdruck wird stets als eine Menge von Alternativmengen von Pseudo-Intervallen über $\mathcal{D}$, der Menge der Paare aus Monats- und Minutenangaben, interpretiert. Jedes dieser Pseudo-Intervalle steht für einen Bereich von Dauerangaben. Wenn also jede der Alternativmengen leer ist, beziehungsweise wenn gar die Gesamtmenge leer ist, dann repräsentiert diese Menge keine einzige Dauerangabe, stellt also eine inkonsistente Beschreibung einer Dauer dar. Dies ist etwa für einen Ausdruck der Kategorie `RANGE` der Fall, bei dem das erste Argument größer als das zweite ist, wie beispielsweise bei `range(dur(5,days),dur(4,days))` (siehe hierzu auch die Anmerkungen in Abschnitt 6.3.4 des letzten Kapitels).

Definition 7.1 *Das Objekt $D \in \mathcal{P}(\mathcal{P}(Int^{*}(\mathcal{D})))$ aus dem semantischen Bereich zur Interpretation von Ausdrücken zur Beschreibung einer Dauer heißt* inkonsistent *genau dann, wenn gilt:*

$$D = \emptyset \quad oder \quad \forall M \in D : M = \emptyset$$

Entsprechend wird auch ein Ausdruck $\delta \in L(\mathtt{DURATION})$, für den $\mathcal{I}_{DUR}(\delta)$ inkonsistent ist, als inkonsistent bezeichnet.

In analoger Weise wird auch die Inkonsistenz über dem vereinfachten semantischen Bereich für $L(\mathtt{DURATION})$ definiert.

Definition 7.2 *Das Objekt $D \in \mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))$ aus dem vereinfachten semantischen Bereich zur Interpretation von Ausdrücken zur Beschreibung einer Dauer heißt* inkonsistent *genau dann, wenn gilt:*

$$D = \emptyset \quad oder \quad \forall M \in D : M = \emptyset$$

Entsprechend wird auch ein Ausdruck $\delta \in L(\mathtt{DURATION})$, für den $\sigma(\mathcal{I}_{DUR}(\delta))$ inkonsistent ist, als inkonsistent bezeichnet.

7.1.2 Zeitpunkte

Die Sprache $L(\mathtt{POINT})$ ist syntaktisch stark übergeneralisierend. In den meisten Fällen führt dies zu einer inkonsistenten Semantik. Auch die Konjunktion unvereinbarer Ausdrücke, wie zum Beispiel `[dow:tue,pow:weekend]`, führen zu semantisch inkonsistenten Interpretationen. Das Erkennen der Inkonsistenz funktioniert genauso wie für $L(\mathtt{DURATION})$.

Definition 7.3 *Das Objekt $P \in \mathcal{P}(\mathcal{P}(Int(\mathcal{C})))$ aus dem semantischen Bereich zur Interpretation von Ausdrücken zur Beschreibung eines Zeitpunktes heißt* inkonsistent *genau dann, wenn gilt:*

$$P = \emptyset \quad oder \quad \forall M \in P : M = \emptyset$$

Entsprechend wird auch ein Ausdruck $\pi \in L(\mathtt{POINT})$, für den $\mathcal{I}_P(\pi)$ inkonsistent ist, als inkonsistent bezeichnet.

7.1.3 Termine

Bevor Inkonsistenz über dem semantischen Bereich von $L(\texttt{DATE})$ erklärt werden kann, muß definiert werden, was es bedeutet, daß ein Quadrupel aus Terminbeginn, -ende, -verlauf und -dauer inkonsistent ist. Das ist sicherlich immer dann der Fall, wenn wenigstens eines dieser vier Elemente inkonsistent ist. Hierbei handelt es sich jedoch nur um eine hinreichende Bedingung, keine notwendige. Ein Quadrupel ist beispielsweise auch dann inkonsistent, wenn zwar jedes seiner vier Elemente für sich genommen konsistent ist, die durch das erste Element kodierten möglichen Terminbeginne aber alle *nach* den durch das zweite Element dargestellten Terminenden liegen. Ein Termin wird durch einen Anfangspunkt $t_1 \in \mathcal{C}$, einen Endpunkt $t_2 \in \mathcal{C}$ und eine Dauer $d \in \mathbb{N}_0$ mit $\langle t_1, t_2 \rangle_{\mathcal{C}} = d$ definiert. Allgemein gilt, daß ein Quadrupel (P_f, P_t, P_d, D) genau dann inkonsistent ist, wenn es keinen solchen Termin gibt, der durch dieses Quadrupel umschrieben wird, das heißt, wenn es nicht der Fall ist, daß es erstens in P_f ein Intervall gibt, das t_1 enthält, daß es zweitens in P_t ein Intervall gibt, das t_2 enthält, daß es drittens in D ein Intervall gibt, welches d enthält, und daß es viertens für jeden Zeitpunkt t von t_1 bis t_2 in P_d ein Intervall gibt, das t enthält. Dies wird nun unter Verwendung der in Definition 6.30 eingeführten Relation $\vdash_{DATE}$ formalisiert.

Definition 7.4 *Ein Quadrupel* $(P_f, P_t, P_d, D) \in \mathcal{P}(\mathcal{P}(Int(\mathcal{C})))^3 \times \mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))$ *heißt* inkonsistent *genau dann, wenn gilt:*

$$\neg\exists\, (t_1, t_2, d) \in \mathcal{C}^2 \times \mathbb{N}_0 : (P_f, P_t, P_d, D) \vdash_{DATE} (t_1, t_2, d)$$

Eine Überprüfung dieser Bedingung wird in vielen Fällen schwierig sein. Die bereits angesprochene hinreichende Bedingung für die Inkonsistenz eines Quadrupels, nämlich die Inkonsistenz wenigstens eines seiner Elemente, kann natürlich einfacher getestet werden.

Nun kann die Inkonsistenz einer Interpretation eines TEL-Ausdrucks zur Beschreibung eines Termins durch die Inkonsistenz von Quadrupeln definiert werden.

Definition 7.5 *Das Objekt* $A \in \mathcal{P}(\mathcal{P}(\mathcal{P}(\mathcal{P}(Int(\mathcal{C})))^3 \times \mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))))$ *aus dem semantischen Bereich zur Interpretation von Ausdrücken zur Beschreibung eines Termins heißt* inkonsistent *genau dann, wenn gilt:*

$$A = \emptyset \quad \textit{oder} \quad \forall\, M \in A : M = \emptyset \vee \forall\, Q \in M : Q \ \textit{ist inkonsistent}$$

Entsprechend wird auch ein Ausdruck $\alpha \in L(\texttt{DATE})$*, für den* $\mathcal{I}(\alpha)$ *inkonsistent ist, als inkonsistent bezeichnet.*

7.2 Äquivalenz

Neben der syntaktischen Gleichheit von Ausdrücken der TEMPORAL EXPRESSION LANGUAGE und der unmittelbaren, elementweisen Gleichheit ihrer semantischen Interpretationen sollte auch eine semantische Äquivalenzrelation angegeben werden, über die bestimmt werden kann, ob zwei Dauer-, Zeitpunkt- oder Terminbeschreibungen tatsächlich die gleichen Mengen von Dauern, Zeitpunkten oder Terminen beinhalten. Hierfür muß zunächst grundsätzlich geklärt werden, wie Interpretationsalternativen zu behandeln sind. Wie die semantische Äquivalenz etwa zweier Mengen von Zeitpunkt-Intervallen zu definieren ist, ist klar. Es müssen lediglich die Mengen der durch die beiden Gruppen von Intervallen beschriebenen Zeitpunkte verglichen werden. Für eine entsprechende Äquivalenzrelation über Mengen solcher Alternativmengen gibt es jedoch wenigstens zwei Möglichkeiten. Entweder könnten zwei solche Mengen als äquivalent bezeichnet werden, wenn jede Alternativmenge der ersten Menge ein Pendant in der zweiten hat, zu dem es äquivalent ist, oder sie könnten als äquivalent bezeichnet werden, wenn es in ihnen wenigstens ein Paar von Alternativmengen gibt, die semantisch äquivalent sind. Ich werde die zweiten Variante verfolgen. Dies scheint mir angemessener, da normalerweise nur je eine Alternativmenge der 'wahren' Bedeutung eines Ausdrucks entspricht.[1] An dieser Stelle wird übrigens der Unterschied zwischen Interpretationsalternativen und Disjunktionen nochmals verdeutlicht: würde man beide Phänomene gleichrangig behandeln, als Interpretationen also einfache Mengen wählen, so wäre diese Art der Definition semantischer Äquivalenz nicht möglich.

7.2.1 Dauer

Aus den im letzten Kapitel zu Genüge diskutierten Gründen kann über $\mathcal{D}$ keine 'exakte' Äquivalenzrelation definiert werden. Über dem semantischen Bereich von Ausdrücken der Kategorie `DURATION` wird dies also erst recht nicht möglich sein. Für den vereinfachten semantischen Bereich hingegen ist es sehr wohl möglich, wie die folgende Definition zeigt.

Definition 7.6 *Die zwei Objekte $D_1, D_2 \in \mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))$ aus dem (vereinfachten) semantischen Bereich für Ausdrücke zur Beschreibung der Länge einer zeitlichen Dauer heißen* semantisch äquivalent *genau dann, wenn es in D_1 eine Alternativmenge M_1 und in D_2 eine Alternativmenge M_2 gibt, für die die durch sie beschriebenen Mengen von Minutenangaben identisch sind, das heißt:*

$$\begin{aligned} D_1 \equiv D_2 \quad :\Longleftrightarrow \quad & \exists\, M_1 \in D_1 : \exists\, M_2 \in D_2 : \\ & \forall\, d \in \mathbb{N}_0 : ((\exists\, I_1 \in M_1 : d \in I_1) \leftrightarrow (\exists\, I_2 \in M_2 : d \in I_2)) \end{aligned}$$

[1] Allerdings kann es so unter Umständen passieren, daß zwei 'falsche' Alternativmengen semantisch äquivalent sind und daraufhin auf die Äquivalenz der beiden semantischen Objekte geschlossen wird. Insofern handelt es sich um eine 'optimistische' Auffassung von Äquivalenz.

7.2.2 Zeitpunkte

Über dem semantischen Bereich von $L(\mathtt{POINT})$ wird eine entsprechende Äquivalenzrelation definiert.

Definition 7.7 *Die zwei Objekte* $P_1, P_2 \in \mathcal{P}(\mathcal{P}(Int(\mathcal{C})))$ *aus dem semantischen Bereich für Ausdrücke zur Beschreibung der Lage eines Zeitpunktes heißen* semantisch äquivalent *genau dann, wenn es in* P_1 *eine Alternativmenge* M_1 *und in* P_2 *eine Alternativmenge* M_2 *gibt, für die die durch sie beschriebenen Mengen von Zeitpunkten identisch sind, das heißt:*

$$\begin{aligned} P_1 \equiv P_2 \quad :\Longleftrightarrow \quad & \exists\, M_1 \in P_1 : \exists\, M_2 \in P_2 : \\ & \forall\, t \in \mathcal{C} : ((\exists\, I_1 \in M_1 : t \in I_1) \leftrightarrow (\exists\, I_2 \in M_2 : t \in I_2)) \end{aligned}$$

7.2.3 Termine

Für Ausdrücke der Sprache $L(\mathtt{DATE})$ ist die Situation deutlich schwieriger. Unter Verwendung der Relation $\vdash_{DATE}$ ist zwar eine recht elegante Formulierung möglich, wie die folgende Definition zeigt, eine reale Auswertung der Äquivalenzbedingung wird in vielen Fällen jedoch sehr rechenintensiv sein. Dieses Manko hat letztlich genauso wie die Schwierigkeiten bei der Formulierung einer geeigneten Inkonsistenzbedingung seinen Grund darin, daß im semantischen Bereich für Terminbeschreibungen die verschiedenen Termine nicht explizit gelistet werden, sondern implizit über die Werte in den Quadrupeln zusammen mit gewissen Nebenbedingungen angegeben werden. Meine Beweggründe für dieses Vorgehen habe ich bereits im letzten Kapitel dargelegt. Die Menge der Termine im Rahmen der semantischen Interpretationen direkt zu berechnen hieße lediglich, die bei der Definition von Inkonsistenz und Äquivalenz bestehenden Probleme nach vorne zu verlagern, und dieses *immer*, auch dann, wenn die Frage nach Inkonsistenz beziehungsweise Äquivalenz überhaupt nicht gestellt wird.

Definition 7.8 *Die Objekte* $A_1, A_2 \in \mathcal{P}(\mathcal{P}(\mathcal{P}(\mathcal{P}(Int(\mathcal{C})))^3 \times \mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))))$ *aus dem semantischen Bereich für Ausdrücke zur Beschreibung eines Termins heißen* semantisch äquivalent *genau dann, wenn es in* A_1 *eine Alternativmenge* M_1 *und in* A_2 *eine Alternativmenge* M_2 *gibt, für die die Mengen der durch die in ihnen enthaltenen Quadrupel* Q *definierten Termine* a *identisch sind, das heißt:*

$$\begin{aligned} A_1 \equiv A_2 \quad :\Longleftrightarrow \quad & \exists\, M_1 \in A_1 : \exists\, M_2 \in A_2 : \\ & \{a \mid Q \in M_1 \wedge Q \vdash_{DATE} a\} = \{a \mid Q \in M_2 \wedge Q \vdash_{DATE} a\} \end{aligned}$$

7.3 Generalisierung und Spezialisierung

Nachdem nun ein semantischer Äquivalenzbegriff eingeführt ist, kann definiert werden, wann ein zeitlicher Ausdruck spezieller oder genereller als ein anderer ist. Bei der Spezialisierung handelt es sich um eine Art semantischer Teilmengenrelation, bei der Generalisierung um das entsprechende Gegenstück.

7.3.1 Dauer

Für Ausdrücke der Kategorie `DURATION` können Spezialisierung und Generalisierung nur für den Fall einer Abbildung auf den vereinfachten semantischen Bereich definiert werden. Das liegt daran, daß andernfalls kein semantischer Schnitt erklärt wäre. Für den vereinfachten Bereich wurde er durch Definition 6.28 eingeführt.

Definition 7.9 *Seien $D_1, D_2 \in \mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))$ Objekte zur (vereinfachten) Beschreibung der Länge einer zeitlichen Dauer. Es wird definiert:*

(i) D_1 heißt spezieller *als D_2 genau dann, wenn D_1 und der semantische Schnitt von D_1 und D_2 semantisch äquivalent sind, das heißt:*

$$D_1 \sqsubseteq D_2 \;:\Longleftrightarrow\; D_1 \equiv D_1 \sqcap D_2$$

(ii) D_1 heißt genereller *als D_2 genau dann, wenn D_2 spezieller als D_1 ist, das heißt:*

$$D_1 \sqsupseteq D_2 \;:\Longleftrightarrow\; D_2 \sqsubseteq D_1$$

7.3.2 Zeitpunkte

Die Definitionen von Spezialisierung beziehungsweise Generalisierung über dem semantischen Abbild von $L($`POINT`$)$ lauten entsprechend. Die Schnittoperation über dem semantischen Bereich für Ausdrücke der Kategorie `POINT` wurde in Definition 6.24 definiert.

Definition 7.10 *Seien $P_1, P_2 \in \mathcal{P}(\mathcal{P}(Int(\mathcal{C})))$ Objekte zur Beschreibung der Lage eines Zeitpunktes. Es wird definiert:*

(i) P_1 heißt spezieller *als P_2 genau dann, wenn P_1 und der semantische Schnitt von P_1 und P_2 semantisch äquivalent sind, das heißt:*

$$P_1 \sqsubseteq P_2 \;:\Longleftrightarrow\; P_1 \equiv P_1 \sqcap P_2$$

(ii) P_1 heißt genereller *als P_2 genau dann, wenn P_2 spezieller als P_1 ist, das heißt:*

$$P_1 \sqsupseteq P_2 \;:\Longleftrightarrow\; P_2 \sqsubseteq P_1$$

7.3.3 Termine

Schließlich wird die Definition noch auf die Semantik von Terminbeschreibungen ausgedehnt. Definition 6.32 im letzten Kapitel erklärt den semantischen Schnitt für diesen Fall.

Definition 7.11 *Seien* $A_1, A_2 \in \mathcal{P}(\mathcal{P}(\mathcal{P}(\mathcal{P}(\mathit{Int}(\mathcal{C})))^3 \times \mathcal{P}(\mathcal{P}(\mathit{Int}(\mathbb{N}_0)))))$ *Objekte zur Beschreibung eines Termins. Es wird definiert:*

(i) A_1 *heißt* spezieller *als* A_2 *genau dann, wenn* A_1 *und der semantische Schnitt von* A_1 *und* A_2 *semantisch äquivalent sind, das heißt:*

$$A_1 \sqsubseteq A_2 \;:\Longleftrightarrow\; A_1 \equiv A_1 \sqcap A_2$$

(ii) A_1 *heißt* genereller *als* A_2 *genau dann, wenn* A_2 *spezieller als* A_1 *ist, das heißt:*

$$A_1 \sqsupseteq A_2 \;:\Longleftrightarrow\; A_2 \sqsubseteq A_1$$

7.4 Semantische Differenz

Analog zur Differenz von Mengen werde ich in diesem Abschitt definieren, was unter der semantischen Differenz zweier Interpretationen von Zeitausdrücken zu verstehen ist. Eine solche Operation könnte zum Beispiel dazu dienen zu bestimmen, worin genau der zusätzliche Informationsgehalt einer im Verlaufe eines Dialogs verfeinerten Terminbeschreibung besteht.

7.4.1 Dauer

Für zeitliche Ausdrücke zur Beschreibung einer Dauer muß (und kann) der Begriff der semantischen Differenz nur über dem vereinfachten semantischen Bereich $\mathcal{P}(\mathcal{P}(\mathit{Int}(\mathbb{N}_0)))$ der Sprache $L(\texttt{DURATION})$ erklärt werden. Dazu werden die in Definition 6.28 beziehungsweise 6.29 eingeführten Operationen des semantischen Schnitts und der Komplementbildung über diesem Bereich gebraucht.

Definition 7.12 *Seien* $D_1, D_2 \in \mathcal{P}(\mathcal{P}(\mathit{Int}(\mathbb{N}_0)))$ *Objekte zur (vereinfachten) Beschreibung der Länge einer zeitlichen Dauer. Dann heißt die Menge*

$$D_1 \setminus\!\!\setminus D_2 \;:=\; D_1 \sqcap \overline{D_2}$$

die semantische Differenz *von* D_1 *und* D_2*.*

7.4.2 Zeitpunkte

Für Ausdrücke zur Beschreibung der Lage eines Zeitpunktes wird die semantische Differenz entsprechend definiert. Schnitt und Komplement über dem semantischen Bereich für $L(\mathtt{POINT})$ wurden im letzten Kapitel in den Definitionen 6.24 und 6.25 erklärt.

Definition 7.13 *Seien $P_1, P_2 \in \mathcal{P}(\mathcal{P}(Int(\mathcal{C})))$ Objekte zur Beschreibung der Lage eines Zeitpunktes. Dann heißt die Menge*

$$P_1 \setminus\!\!\setminus P_2 \ := \ P_1 \sqcap \overline{P_2}$$

die semantische Differenz *von P_1 und P_2.*

7.4.3 Termine

Nun muß noch die semantische Differenz der Interpretationen zweier Terminbeschreibungen definiert werden. Für den semantischen Bereich von $L(\mathtt{DATE})$ wurden Schnitt und Komplement in den Definitionen 6.32 beziehungsweise 6.33 beschrieben.

Definition 7.14 *Seien $A_1, A_2 \in \mathcal{P}(\mathcal{P}(\mathcal{P}(\mathcal{P}(Int(\mathcal{C})))^3 \times \mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))))$ Objekte zur Beschreibung eines Termins. Dann heißt die Menge*

$$A_1 \setminus\!\!\setminus A_2 \ := \ A_1 \sqcap \overline{A_2}$$

die semantische Differenz *von A_1 und A_2.*

7.5 Zusammenfassung

In diesem Kapitel wurden einige grundlegende Inferenzen über den semantischen Bereichen für TEL-Ausdrücke eingeführt, die zur semantischen Analyse eines Dialogverlaufs nützlich sein können. Insbesondere wurde erklärt, wie inkonsistente Ausdrücke erkannt werden können, und eine semantische Äquivalenzrelation wurde eingeführt. Zu zwei gegebenen Ausdrücken kann nun bestimmt werden, ob einer der Ausdrücke genereller oder spezieller als der andere ist, beziehungsweise, ob sie überhaupt miteinander vereinbar sind. Außerdem wurde gezeigt, wie die semantische Differenz zweier Ausdrücke zu berechnen ist.

Damit schließt der theoretische Teil dieser Arbeit. Das nächste Kapitel beschreibt einen Ansatz zur Implementierung der eingeführten Funktionen.

Kapitel 8

Implementierung

'Das ist wohl gesprochen', versetzte Candide,
'aber wir müssen unseren Garten bestellen.'
Voltaire: Candide oder der Optimismus.

Nachdem in dieser Arbeit die Syntax einer Repräsentationssprache für zeitliche Ausdrücke angegeben, ihre formale Semantik definiert und diverse Inferenzen mit Ausdrücken dieser Sprache beschrieben worden sind, soll nun eine konkrete Implementierung des Systems besprochen werden. Beim Entwurf dieses Prototyps ging es mir vor allem um die Festlegung einer Grundstruktur eines solchen Systems und nicht um die Entwicklung eines unmittelbar einsatzfähigen VERBMOBIL-Moduls. Die vorliegende Implementierung erfolgte vollständig in PROLOG (siehe beispielsweise [Bratko, 1990]).

Das Kapitel ist wie folgt gegliedert. Im ersten Teil wird gezeigt, wie die syntaktische Korrektheit eines TEL-Ausdrucks überprüft werden kann. Dann wird die Implementierung der Transformationsregeln aus Kapitel 5 beschrieben. Der dritte Teil beschäftigt sich mit den semantischen Bereichen und den über ihnen erklärten Funktionen (das schließt die im letzten Kapitel definierten Inferenzen mit ein). Im vierten Teil wird die Implementierung der Interpretationsfunktionen dokumentiert. Vor der abschießenden Zusammenfassung werden im fünften Teil einige Verbesserungsmöglichkeiten des bestehenden Prototyps diskutiert.

8.1 Kategorien und Syntaktische Korrektheit

An verschiedenen Stellen des Systems wird es notwendig sein zu erkennen, ob ein bestimmter Ausdruck einer gewissen TEL-Kategorie angehört oder nicht. Hierzu wird das zweistellige Prädiktat `category/2` definiert; es ist genau dann erfolgreich, wenn der an erster Stelle übergebene Ausdruck zu der an zweiter Stelle übergebenen Kategorie gehört. Die Syntax von Ausdrücken ist genau die gleiche wie in diesem Text, lediglich die groß geschriebenen Kategorienamen müssen in Quotes gefaßt werden, damit sie von PROLOG nicht als Variablen interpretiert werden.

Die Implementierung dieses Prädikats ist denkbar einfach, es muß allerdings berücksichtigt werden, daß auch die verschiedenen Formen von Listen eigenständige Kategorien darstellen, wie etwa `DATE*` und `DATE+`. Angenommen, `category/2` wurde für `DATE` bereits geschrieben, dann wird durch folgende zwei Klauseln definiert, welche Terme zur Kategorie `DATE+` gehören, also nichtleere Listen von Ausdrücken der Kategorie `DATE` sind.

```
category( [Term], 'DATE+') :-
  category( Term, 'DATE').

category( [Term|Terms], 'DATE+') :-
  category( Term, 'DATE'),
  category( Terms, 'DATE+').
```

Für Listen, die auch leer sein können, beziehungsweise für die anderen Typen werden entsprechende Klauseln geschrieben. Am einfachsten ist natürlich das Erkennen von Terminalsymbolen. Ein Beispiel hierfür ist `day` aus der Kategorie `GENERIC`, die zugehörige Klausel lautet:

```
category( day, 'GENERIC').
```

Bei den meisten übrigen Termen läßt sich die Kategorie schon am jeweiligen Funktor ablesen. Nur einige wenige Funktoren, wie etwa `set`, sind für mehr als eine Kategorie zulässig. Um die Kategorie eines zusammengesetzten Terms zu überprüfen, müssen seine Argumente extrahiert und auf Zugehörigkeit zu den zulässigen Kategorien getestet werden. Dies wird durch das folgende Programmfragment illustriert. Es gibt die Definition der Kategorie `DATE` erschöpfend wieder.

```
category( Term, 'DATE') :-
  category( Term, 'DATE*').

category( set(Term), 'DATE') :-
  category( Term, 'DATE+').

category( one_out_of(Term), 'DATE') :-
  category( Term, 'DATE+').

category( not(Term), 'DATE') :-
  category( Term, 'DATE').

category( for:Term, 'DATE') :-
  category( Term, 'DURATION').

category( Term1:Term2, 'DATE') :-
  category( Term1, 'POINTLABEL'),
  category( Term2, 'POINT').

category( Term1:Term2, 'DATE') :-
  category( Term1, 'POINTLABEL+'),
  category( Term2, 'POINT').
```

Auf entsprechende Weise werden auch die Klauseln für alle übrigen TEL-Kategorien implementiert. Ihre Reihenfolge orientiert sich an der Definition der Grammatik (siehe Anhang B.2). Zum Schluß soll noch auf zwei Sonderfälle hingewiesen werden. Zum einen wurde bisher nicht näher definiert, welche Ausdrücke zur Kategorie `NAME` gehören. Diese Festlegung (beziehungsweise die Entscheidung, ob der `tempex`-Operator überhaupt in die Repräsentationssprache mitaufgenommen werden sollte) hängt natürlich stark von der konkreten Realisierung der Dialogverarbeitung, in die eine TEL-Komponente eingebaut werden soll, ab. Durch folgende Klausel wird `NAME` die Menge aller PROLOG-Atome zugeschrieben:

```
category( Term, 'NAME') :-
  atom( Term).
```

Den zweiten Sonderfall stellen die Ausdrücke zur Beschreibung ganzer beziehungsweise natürlicher Zahlen, also die Kategorien `INTEGER`, `NUMBER` und `NUMBER0`, dar. Beispielhaft sei hier die Klausel für `NUMBER` herausgegriffen:

```
category( Term, 'NUMBER') :-
  integer( Term),
  Term > 0.
```

Natürlich kann `category/2` auch dazu verwendet werden, die syntaktische Korrektheit eines TEL-Ausdrucks zu überprüfen. In einem fertigen System sollte sie zwar ohnehin gewährleistet sein, im Laufe seiner Entwicklung werden solche Überprüfungen jedoch mitunter vonnöten sein.

Wird bei einem Aufruf von `category/2` das zweite Argument uninstanziiert gelassen, so liefert Backtracking nacheinander alle möglichen Kategorien dieses Ausdrucks. Aufgrund der Reihenfolge der Klauseln im Programm wird dabei die allgemeinste Kategorie zuerst ausgegeben (darüber hinaus hat die Reihenfolge keine Bedeutung). Folgt dem Aufruf also ein Cut, so kann `category/2` auch zum Finden der (allgemeinsten) Kategorie eines Ausdrucks (also normalerweise `DURATION`, `POINT` oder `DATE`) benutzt werden. Im folgenden Beispiel würde die Variable `Category` dann ausschließlich mit `'POINT'` unifiziert werden.

```
| ?- category( after([dom:6,month:oct]), Category).
Category = 'POINT' ;

Category = 'COMPLEX' ;

Category = 'LIMIT' ;

no
```

8.2 Reduktion auf die Kernsprache

In Kapitel 5 wurde die Reduktion der Temporal Expression Language auf ihren Kern durch die Angabe von Transformationsregeln beschrieben. Das Prädikat `transform/2` implementiert diese Transformationen. Wird an erster Stelle ein TEL-Ausdruck übergeben, so wird die an zweiter Stelle angegebene Variable mit dem vollständig reduzierten Ausdruck der Kernsprache unifiziert. Jedem Aufruf von `transform/2` sollte ein Cut folgen, da weiteres Backtacking falsche (nämlich nicht dem Kern angehörige) Lösungen liefern würde. Daher wird das Prädikat `reduce/2` definiert. Außerhalb des Reduktionsmoduls sollte statt `transform/2` stets `reduce/2` aufgerufen werden.

```
reduce( Term, ReducedTerm) :-
  transform( Term, ReducedTerm), !.
```

Bei der Implementierung von `transform/2` ist darauf zu achten, daß auch Ausdrücke, für die selbst keine Transformationsregeln greifen, die aber Teilausdrücke haben könnten, die reduziert werden müssen, richtig behandelt werden. Für Ausdrücke, die sicher nicht mehr weiter reduziert werden können, also zum Beispiel einfache Terminalsymbole, wird als *letzte* Klausel die folgende definiert:

```
transform( Term, Term).
```

Die Reihenfolge der übrigen Klauseln ist unerheblich. Unter ihnen können im wesentlichen vier Klassen unterschieden werde. Erstens gibt es Ausdrücke, die selbst zwar nicht transformierbar sind, für deren Teilausdrücke `transform/2` aber trotzdem aufgerufen werden muß. Ein Beispiel hierfür sind Ausdrücke mit dem `the_around`-Funktor. Für diese Klasse wurde keine Transformationsregel aufgestellt, es ist aber möglich, daß eines seiner Argumente reduzierbar ist. In der entsprechenden Klausel werden zunächst die Kategorien der Teilausdrücke überprüft. Diese Überprüfung ist in den meisten Fällen nicht nötig, insofern sichergestellt ist, daß ausschließlich syntaktisch korrekte Ausdrücke an `transform/2` übergeben werden. Zur Effizienzsteigerung kann die Implementierung dahingehend leicht verkürzt werden. Nach dieser Überprüfung wird `tranform/2` für die Argumente aufgerufen (`TTerm1` beziehungsweise `TTerm2` stehen für *transformed term*).

```
transform( the_around(Term1,Term2), the_around(TTerm1,TTerm2)) :-
  category( Term1, 'COUNTABLE'),
  category( Term2, 'POINT'),
  transform( Term1, TTerm1),
  transform( Term2, TTerm2).
```

Bei der zweiten Klasse von Klauseln handelt es sich um tatsächliche Transformationsregeln, bei denen allerdings sichergestellt ist, daß keines der Argumente des Zielausdrucks weiter transformiert werden muß. Hierzu zählt etwa die Regel zur Behandlung von Ausdrücken der Form `woy:WOY`. In solch einem Fall wird lediglich die Kategorie des Arguments überprüft (was, wie gesagt, nicht unbedingt erforderlich wäre).

```
transform( woy:Term, of(Term,week,year)) :-
  category( Term, 'NUMBER').
```

Zur nächste Gruppe von Transformationsregeln gehören die, bei denen einerseits Teilausdrücke möglicherweise noch weiter umgeformt werden müssen, bei denen andererseits aber feststeht, daß sich der Hauptfunktor des Zielausdrucks nicht mehr ändern wird. Das gilt etwa für die Eliminierung des `interval`-Labels vor einem Ausdruck mit `between`.[1] Die Klausel hat eine ähnliche Gestalt wie im vorletzten Beispiel. Der Unterschied besteht darin, daß hier tatsächlich eine Transformation stattfindet; das Ergebnis steht im zweiten Argument.

```
transform( interval:between(Term1,Term2), [from:TTerm1,to:TTerm2]) :-
  category( Term1, 'POINT'),
  category( Term2, 'POINT'),
  transform( Term1, TTerm1),
  transform( Term2, TTerm2).
```

Die vierte Klasse von Klauseln schließlich beschreibt all diejenigen Transformationsregeln, bei denen auf den gesamten Zielausdruck eine weitere Regel angewandt werden kann. Zum Beispiel wird `tomorrow` zunächst auf `deictic_day(1)` abgebildet. Dieser Ausdruck kann dann wiederum in `deictic(1,day)` umgeformt werden, welcher selbst ebenfalls noch weiter reduzierbar ist. Jede Klausel vollzieht jedoch nur einen solchen Schritt, weshalb mit dem Ergebnis der ersten Transformation `transform/2` ein weiteres Mal aufgerufen wird.

```
transform( tomorrow, TTerm) :-
  transform( deictic_day(1), TTerm).
```

Die Implementierung von `transform/2` ist damit im wesentlichen erklärt. Um die Reduktion von Ausdrücken der Formen `interval:POINT*`, `interval:set(POINT+)` beziehungsweise `interval:one_out_of(POINT+)` zu realisieren, mußte allerdings zusätzliche eine Funktion `map_interval/2` implementiert werden, die in einer gegebenen Liste alle Elemente mit `interval` labelt. Des weiteren wurde `kill_interval/2` definiert, um aus einer Liste des Typs `POINTLABEL+` alle Auftreten von `interval` löschen zu können.

Analog zu `category/2` wird das Prädikat `ccategory/2` (für *core-category*) definiert, welches die Erkennung der Kategorien von Ausdrücken der Kernsprache entsprechend Anhang B.3 leistet.

[1] Hier ist der Zielausdruck ausnahmsweise eine Liste, weshalb das Wort *Hauptfunktor* vielleicht etwas irreführend sein mag; aber auch eine Liste hat einen gewissermaßen 'unsichtbaren' Funktor.

8.3 Semantische Grundfunktionen und Inferenzen

Zu den semantischen Grundfunktionen, die als nächstes beschrieben werden sollen, rechne ich insbesondere die im ersten Teil von Kapitel 6 unter der Überschrift *Kalendersemantik* definierten Funktionen. Außerdem zählen hierzu die auf den diversen semantischen Bereichen erklärten Operationen wie Schnitt oder Vereinigung. Im folgenden werde ich nur die interessantesten Funktionen im Detail beschreiben, einige weniger wichtige auch ganz überspringen.

8.3.1 Darstellung von Mengen

Wichtigste Strukur bei der (theoretischen) Definition der Semantik zeitlicher Ausdrück war und ist die Menge. In den meisten Fällen wurden Mengen nicht explizit durch die unmittelbare Angabe aller Elemente beschrieben, sondern implizit durch eine bestimmte (generierende) Bedingung definiert. Grundsätzlich hat solch eine Mengenbeschreibung die folgende Form:

$$M = \{x \in D \mid cond(x)\}$$

Das heißt, M wird als die Menge aller Elemente x aus dem Bereich D, die die Bedingung $cond(x)$ erfüllen, definiert. Zunächst einmal gilt es, grundsätzlich zu klären, wie eine solche Struktur in PROLOG umzusetzen ist. In der Theorie haben die bezeichneten Mengen oft unendlich viele Elemente. In der Praxis lassen sich die in Frage kommenden Bereiche jedoch einschränken, so daß sichergestellt ist, daß unendlichelementige Mengen nicht auftreten werden. Damit wird eine explizite Darstellung in Form einer Liste aller mit Hilfe der jeweiligen Bedingung generierbaren Elemente möglich. Ich werde diesem Ansatz folgen. Da allerdings mitunter Listen beträchtlicher (wenn auch nicht unendlicher) Länge generiert werden können, ist dies sicherlich nicht die optimale Lösung. Andere Vorgehensweisen beziehungsweise mögliche Verbesserungen des vorgestellten Ansatzes werden in Abschnitt 8.5 angesprochen werden.

8.3.2 Semantischer Bereich für Dauerangaben

Eine Dauer wird als ein Paar aus Monats- und Minutenangabe, also als ein Element der Menge $\mathcal{D}$, repräsentiert. Beide Angaben werden in Form von natürlichen Zahlen (inklusive 0) gemacht. Die Zugehörigkeit eines Objekts zu $\mathcal{D}$ kann mit Hilfe des PROLOG-Prädikats `inD/1` überprüft werden.

```
inD( (N1,N2)) :-
  integer( N1), N1 >= 0,
  integer( N2), N2 >= 0.
```

In Definition 6.19 wurde eine Multiplikation von rationalen Zahlen mit Elementen aus $\mathcal{D}$ angegeben. Das entsprechende Prädikat lautet `multiply/3`. Die in

Definition 6.17 definierte 'Ordnung' über $\mathcal{D}$ wird durch das Prädikat `leqD/2` implementiert.

Pseudo-Intervalle werden als zweielementige Terme mit dem Funktor `pi` dargestellt. Beispielsweise wird "fünf bis zehn Minuten" durch den PROLOG-Term `pi((0,5),(0,10))` repräsentiert. Mit `inPseudoIntD/1` kann die Zugehörigkeit zur Menge der Pseudo-Intervalle über $\mathcal{D}$ nachgeprüft werden.

```
inPseudoIntD( pi(D1,D2)) :-
  inD( D1),
  inD( D2),
  leqD( D1, D2).
```

Als semantischer Bereich für Ausdrücke der Kategorie `DURATION` wurde in Kapitel 6 $\mathcal{P}(\mathcal{P}(Int^*(\mathcal{D})))$ definiert. Durch einen Aufruf des Prädikats `inSemanticDomainDUR/1` kann nachgeprüft werden, ob ein Objekt zu diesem Bereich gehört oder nicht.[2]

```
inSemanticDomainDUR( Lists) :-
  forall( member( List, Lists),
          forall( member( PseudoInt, List), inPseudoIntD( PseudoInt))).
```

8.3.3 Vereinfachter Semantischer Bereich für Dauerangaben

Durch Anwendung der Funktion σ werden Objekte des semantischen Bereichs von $L($`DURATION`$)$ auf den vereinfachten semantischen Bereich für Dauerangaben abgebildet. Das entsprechende PROLOG-Prädikat ist `simplify/2`. Es wird wie folgt implementiert:

```
simplify( D, S) :-
  findall( Set,
  (
    member( M, D),
    findall( i(N1,N2),
    (
      member( pi((N11,N12),(N21,N22)), M),
      N1 is 40320 * N11 + N12,
      N2 is 44640 * N21 + N22
    ),
    Set)
  ),
  S).
```

[2] Das Prädikat `forall/2` ist erfolgreich, wenn für jede Lösung des an erster Stelle angegebenen Generators der an zweiter Stelle gegebene Test erfolgreich ist. Steht es in einem PROLOG-Dialekt nicht zur Verfügung, so kann es durch `forall( A, B) :- \+ ( A, \+ B).` implementiert werden.

Hierbei steht `D` für das gegebene semantische Objekt zur Beschreibung einer Dauer und `S` für das Ergebnis der Berechnung, also für das entsprechende Objekt aus dem vereinfachten semantischen Bereich. Über das erste `findall/3` wird `S` als die Liste der Elemente `Set` definiert, die wiederum für alle Alternativmengen `M` aus `D` durch das zweite `findall/3` wie folgt gebildet werden. (Da es sich im Ergebnis um echte Intervalle – im Gegensatz zu Pseudo-Intervallen – handelt, wird zur Darstellung nun der Funktor `i` verwendet.) Durch (das zweite) `member/2` werden alle Pseudo-Intervalle in `M` generiert. Zu jedem von ihnen werden dann die neuen Randpunkte `N1` und `N2` berechnet. `Set` ergibt sich als die Liste aller so erzeugten `i(N1,N2)`, also aller Intervalle von `N1` bis `N2`. Eine solche Schachtelung zweier `findall/3`-Prädikate zur Generierung einer Liste von Listen wird im folgenden noch häufig gebraucht werden.

Die Zugehörigkeit zur Menge der Intervalle über $\mathbb{N}_0$ kann durch das Prädikat `inIntN0/1` überprüft werden. Um testen zu können, ob eine bestimmte Zahl Element eines solchen Intervalls ist, wird `elementIntN0/2` wie folgt implementiert:

```
elementIntN0( N, i(N1,N2)) :-
  N1 =< N,
  N =< N2.
```

Dieses Prädikat kann nur zur Überprüfung benutzt werden. Um (in Analogie zu `member/2`) auch alle Elemente eines Intervalls generieren zu können, wird zusätzlich `get_elementIntN0/2` eingeführt:

```
get_elementIntN0( N1, i(N1,_)).

get_elementIntN0( N, i(N1,N2)) :-
  N0 is N1 + 1,
  N0 =< N2,
  get_elementIntN0( N, i(N0,N2)).
```

Die erste Klausel bewirkt die Rückgabe des linken Randpunktes. Bei Backtracking wird im nächsten Schritt die zweite Klausel erreicht. In ihr wird das Prädikat erneut, diesmal jedoch mit um 1 erhöhtem linkem Randpunkt aufgerufen. So werden nacheinander alle Elemente des Intervalls generiert. Zu reinen Überprüfungszwecken (also mit einem festen Wert als erstem Argument) könnte `get_elementIntN0/2` natürlich auch benutzt werden; es ist aber wesentlich ineffizienter als `elementIntN0/2`.

Des weiteren wurde das Prädikat `intN0_coverage/2` implementiert. Es ist erfolgreich, wenn das an erster Stelle übergebene Intervall durch die Intervalle in der an zweiter Stelle übergebenen Liste vollständig überdeckt wird. Weiter unten wird es zur Implementierung der Äquivalenzrelation über dem vereinfachten semantischen Bereich für Dauerangaben verwendet werden.

Durch `intN0_intersection/3` wird der Schnitt zweier Intervalle über $\mathbb{N}_0$ verwirklicht. Ist das Ergebnis des Schnitts (der den Intervallen entsprechenden Mengen) die leere Menge, so schlägt der Aufruf des Prädikats fehl.

```
intN0_intersection( i(N11,N12), i(N21,N22), i(N1,N2)) :-
  max( N11, N21, N1),
  min( N12, N22, N2), !,
  N1 =< N2.
```

Die Verwendung von `intN0_intersection/3` wird durch die folgenden beiden Beispiele demonstriert. Im ersten Falle gelingt die Berechnung des Schnitts: $[2, 10] \cap [5, 15] = [5, 10]$. Im zweiten Falle ergäbe sich als Schnitt die leere Menge, der kein Intervall entspricht, daher gibt es keine Lösung für `Result`.

```
| ?- intN0_intersection( i(2,10), i(5,15), Result).
Result = i(5,10)

| ?- intN0_intersection( i(5,10), i(15,20), Result).
no
```

Um den semantischen Schnitt mehrerer interpretierter `DURATION`-Ausdrücke bestimmen zu können, wird der Schnitt einer ganzen Liste von Intervallen über $\mathbb{N}_0$ berechnet werden müssen. Hierzu wird das zweistellige Prädikat `multiple_intN0_intersection/2` durch Mehrfachanwendung des einfachen Schnittprädikats implementiert.

In Anlehnung an das entsprechende Prädikat für den allgemeinen semantischen Bereich für `DURATION`-Ausdrücke wird `inSemanticDomainSimpleDUR/1` definiert. Das Prädikat `semantic_intersectionSimpleDUR/2` zur Berechnung des semantischen Schnitts einer Liste von Objekten aus dem vereinfachten semantischen Bereich von L(`DURATION`) wird folgendermaßen implementiert:

```
semantic_intersectionSimpleDUR( ListOfSemantics, Intersection) :-
  findall( Set,
  (
    members( Ms, ListOfSemantics),
    findall( I,
    (
      members( Is, Ms),
      multiple_intN0_intersection( Is, I)
    ),
    Set)
  ),
  Intersection).
```

Hierbei werden durch `members( Ms, ListOfSemantics)` nacheinander alle möglichen Kombinationen von Alternativmengen `Ms` aus den Interpretationen der zu schneidenden Ausdrücke gewonnen. (Das Prädikat `members/2` generiert bei Angabe einer Liste von Listen an zweiter Position nacheinander an erster Position alle Möglichkeiten einer Liste, die aus jeder der übergebenen Listen je ein Element enthält.) Durch `members( Is, Ms)` werden hieraus dann wiederum alle Kombinationen von Intervallen generiert und für jede der Kombinationen kann dann ihr Schnitt berechnet werden.

Die Implementierung des Prädikats `semantic_complementSimpleDUR/2` zur Gewinnung des semantischen Komplements über $\mathcal{P}(\mathcal{P}(Int(\mathbb{N}_0)))$ erfolgt entsprechend. Über eine Hilfsfunktion müssen hier zunächst für jedes Intervall die sich links und rechts anschließenden einseitig unbegrenzten Intervalle gewonnen werden.

8.3.4 Funktionen aus der Kalendersemantik

Zur Kalendersemantik gehört zunächst einmal die Beschreibung der Menge $\mathcal{C}$ der Zeitpunkte sowie die Definition der Nachfolger- und der Vorgängerrelation über dieser Menge. Die entsprechenden Definitionen aus Abschnitt 6.1.2 lassen sich sehr einfach in PROLOG-Prädikate umsetzen. Zur Illustration wird die Implementierung von `inC/1` zum Testen auf Zugehörigkeit zu $\mathcal{C}$ entsprechend Definition 6.2 angegeben:

```
inC( (Y,M,D,H,I)) :-
  integer( Y), integer( M), integer( D), integer( H), integer( I),
  H >= 0, H < 24,
  I >= 0, I < 60,
  (
    ( member( M, [1,3,5,7,8,10,12]), D >= 1, D =< 31 ) ;
    ( member( M, [4,6,9,11]), D >= 1, D =< 30 ) ;
    ( M = 2, D >= 1, D =< 28 ) ;
    ( leap( Y), M = 2, D = 29)
  ).
```

Hierbei implementiere `leap/1` die in Definition 6.1 beschriebene Funktion, ist also gerade dann erfolgreich, wenn die im Argument übergebene Jahreszahl für ein Schaltjahr steht.

Die Hilfsfunktionen aus Definition 6.3 zum Auslesen der einzelnen Stellen eines einen Zeitpunkt repräsentierenden Quintupels werden durch Prädikate `year/2`, `month/2`, `day/2`, `hour/2` und `minute/2` realisiert. Die Ordnungsrelation $\leq_{\mathcal{C}}$ kann ebenfalls unmittelbar Definition 6.4 folgend als `leqC/2` implementiert werden. Maximum und Minimum zweier Zeitpunkte können durch `maxC/3` beziehungsweise `minC/3` bestimmt werden. Auch die Implementierung von `succ/2` und `pred/2` ergibt sich in einfachster Weise aus Definition 6.7. Über Mehrfachanwendung dieser Prädikate können die primitive Addition und Subtraktion von Minuten zu beziehungsweise von Zeitpunkten beschrieben werden.

Bei der Implementierung der tatsächlichen Additions- und Subtraktionsprädikate folge ich nicht ganz der Darstellung aus Kapitel 6. Dort wurde nur die Addition/Subtraktion von Tagen mit Hilfe der Julianischen Tagesnummern effizienter gemacht. Darüber hinaus lassen sich aber auch die Additition/Subtraktion der verbleibenden Stunden und Minuten schneller direkt als über Mehrfachanwendung von `succ/2` beziehungsweise `pred/2` verwirklichen. Zur Addition von Minuten wird `add_min/3` verwendet:

```
add_min( (Y,M,D,H,I), N, T) :-
  date2jdn( (Y,M,D), JDN),
  T1 is JDN + N//1440,
  jdn2date( T1, (Y1,M1,D1)),
  N1 is N mod 1440,
  Hours is N1 // 60,
  Minutes is N1 mod 60,
  add_hours_0_23( (Y1,M1,D1,H,I), Hours, T2),
  add_minutes_0_59( T2, Minutes, T).
```

Hierbei ist `T` das Ergebnis der Addition von `N` Minuten zu einem Zeitpunkt `(Y,M,D,H,I)`. Das Prädikat `date2jdn/2` liefert in zweiter Position die einem an erster Position gegebenen Datum entsprechende Julianische Tagesnummer. Die Konversion in die entgegengesetzte Richtung wird durch `jdn2date/2` bewerkstelligt. Mit `add_hours_0_23/3` kann eine Zahl von Stunden zwischen 0 und 23 zu einem Zeitpunkt addiert werden. In der Situation, in der diese Funktion gebraucht wird, steht fest, daß die zu addierenden Stunden sich stets im Bereich von 0 bis 23 bewegen, und durch diese Vorbedingung wird die Implementierung einfacher und effizienter als für die allgemeingültige primitive Addition. Entsprechendes gilt für `add_minutes_0_59/3`.

Das Prädikat `subtract_min/3` zur Subtraktion einer Zahl von Minuten von einem Zeitpunkt wird auf analoge Weise wie `add_min/3` implementiert.

Entsprechend Definition 6.9 werden durch `add_mon/3` und `subtract_mon/3` Addition und Subtraktion von Monaten berechnet. Nun kan die Addition eines Monats-/Minuten-Paares zu einem Zeitpunkt aus $\mathcal{C}$ durch `add/3` beschrieben werden (`subtract/3` wird entsprechend implementiert).

```
add( T, (N1,N2), T2) :-
  add_mon( T, N1, T1),
  add_min( T1, N2, T2).
```

Der Abstand zweier Zeitpunkte in Minuten kann unter Verwendung der Prädikate zur Julianischen Tagesnummer folgendermaßen bestimmt werden:

```
difference( (Y1,M1,D1,H1,I1), (Y2,M2,D2,H2,I2), N) :-
  date2jdn( (Y1,M1,D1), JDN1),
  date2jdn( (Y2,M2,D2), JDN2),
  N is 1440 * (JDN2 - JDN1) + 60 * (H2 - H1) + I2 - I1.
```

Mit Hilfe von `mean/3` wird der Mittelwert von zwei Zeitpunkten berechnet. Durch `weekday/2` kann die Wochentagsnummer eines Zeitpunktes errechnet werden.

Damit sind die Funktionen, die sich auf einzelne Zeitpunkte aus $\mathcal{C}$ beziehen, abgehandelt. Als nächstes sollen Funktionen über der Menge $Int(\mathcal{C})$, der Menge der Zeitpunkt-Intervalle, behandelt werden. Die Zugehörigkeit zu dieser Menge wird mit Hilfe von `inIntC/1` überprüft.

```
inIntC( i(T1,T2)) :-
  inC( T1),
  inC( T2),
  leqC( T1, T2).
```

Die Implementierung von `meanIntC/2` zur Bestimmung des Mittelwerts eines Intervalls über $\mathcal{C}$ basiert auf einer einfachen Anwendung von `mean/3`. Entsprechendes gilt für `differenceIntC/3` zur Bestimmung des Abstands zweier Intervalle (als des Abstands ihrer Mittelwerte).

In Analogie zu den in Abschnitt 8.3.3 beschriebenen Prädikaten für Intervalle über $\mathbb{N}_0$ wurden für Intervalle über $\mathcal{C}$ `elementIntC/2`, `get_elementIntC/2` sowie `intC_coverage/2` implementiert.

Schließlich sollen noch einige 'Mengenoperationen' über den Zeitpunkt-Intervallen erklärt werden. Hierzu zählen die Vereinigung (`intC_union/3`) und der Schnitt (`intC_intersection/3`) zweier Intervalle sowie die Inklusionsrelation (`intC_inclusion/2`), also die Abfrage, ob ein Intervall vollständig in einem zweiten enthalten ist. Beispielhaft sei hier nur die Implementierung von `intC_union/3` angegeben. Ist die Vereinigung der als Mengen interpretierten Intervalle nicht zusammenhängend, so schlägt das Prädikat fehl.

```
intC_union( i(T11,T12), i(T21,T22), i(T1,T4)) :-
  minC( T11, T21, T1),
  maxC( T11, T21, T2),
  minC( T12, T22, T3),
  maxC( T12, T22, T4), !,
  ( leqC( T2, T3) ; succ( T3, T2) ).
```

Da später auch die Berechnung des Schnitt aller Intervalle einer Liste erforderlich sein wird, wird `multiple_intC_intersection/2` über eine Mehrfachanwendung von `intC_intersection/3` implementiert.

8.3.5 Semantischer Bereich für Zeitpunktangaben

Die Zugehörigkeit zum semantischen Bereich für $L(\texttt{POINT}) \cup L(\texttt{GENERIC})$, also zu der Menge $\mathcal{P}(\mathcal{P}(\mathit{Int}(\mathcal{C})))$ kann durch eine Anwendung des Prädikats `inSemanticDomainP/1` überprüft werden.

```
inSemanticDomainP( Lists) :-
  forall( member( List, Lists),
          forall( member( Int, List), inIntC( Int))).
```

Über diesem Bereich ist die Minimierungsfunktion μ (siehe Definition 6.21) erklärt. Sie wird durch das zweistellige Prädikat `minimize/2` umgesetzt.

```
minimize( P, Result) :-
  findall( Set,
  (
    member( M, P),
    findall( Interval,
    (
      member( Interval, M),
      forall( member(I, M),
              ( intC_inclusion( I, Interval) -> I = Interval ; true ))
    ),
    Set)
  ),
  Result).
```

Des weiteren müssen semantischer Schnitt und Komplementbildung über diesem Bereich angegeben werden. Hierzu werden `semantic_intersectionP/2` und `semantic_complementP/2` analog zu den entsprechenden Prädikaten für den vereinfachten semantischen Bereich von L(`DURATION`) implementiert.

8.3.6 Semantischer Bereich für Terminbeschreibungen

Ob ein PROLOG-Term ein gültiges Quadrupel zur Terminspezifikation darstellt, kann durch `inQuadruples/1` geprüft werden.

```
inQuadruples( (Pf,Pt,Pd,D)) :-
  inSemanticDomainP( Pf),
  inSemanticDomainP( Pt),
  inSemanticDomainP( Pd),
  inSemanticDomainSimpleDUR( D).
```

Die in Definition 6.30 eingeführte Relation $\vdash_{DATE}$ wird durch das Prädikat `quadruple_describes_date/2` realisiert. Auf ähnliche Weise kann auch `quadruple_generates_date/2` implementiert werden, das für ein nicht instanziiertes zweites Argument alle Tripel *generiert*, die zulässige und durch das an erster Stelle angegebene Quadrupel umschriebene Termine darstellen.[3]

```
quadruple_describes_date( (Pf,Pt,Pd,D), (T1,T2,N)) :-
  difference( T1, T2, N), !,
  member( Mf, Pf),
  member( If, Mf),
  elementIntC( T1, If), !,
  member( Mt, Pt),
  member( It, Mt),
  elementIntC( T2, It), !,
  member( Md, Pd),
  intC_coverage( i(T1,T2), Md), !,
  member( MD, D),
  member( ID, MD),
  elementIntN0( N, ID).
```

[3] Vergleiche hierzu auch den Unterschied zwischen `elementIntN0/2` und `get_elementIntN0/2` (Abschnitt 8.3.3).

Die Zugehörigkeit zu dem mit L(DATE) assoziierten semantischen Bereich kann unter Verwendung von `inQuadruples/1` ebenso einfach wie für die übrigen Bereiche getestet werden.

```
inSemanticDomainDATE( Lists) :-
  forall( member( List, Lists),
          forall( member( Quadruple, List), inQuadruples( Quadruple))).
```

Bevor Schnitt und Komplement über dem semantischen Bereich für Ausdrücke der Kategorie DATE implementiert werden können, muß zunächst einmal der Schnitt zweier Quadrupel beschrieben werden. Hierbei wird für jede der vier Stellen ein semantischer Schnitt berechnet.

```
quadruple_intersection((Pf1,Pt1,Pd1,D1),(Pf2,Pt2,Pd2,D2),(Pf,Pt,Pd,D)) :-
  semantic_intersectionP( [Pf1,Pf2], Pf),
  semantic_intersectionP( [Pt1,Pt2], Pt),
  semantic_intersectionP( [Pd1,Pd2], Pd),
  semantic_intersectionSimpleDUR( [D1,D2], D).
```

Das Prädikat `multiple_quadruple_intersection/2` verallgemeinert dies auf eine Liste von zu schneidenden Quadrupeln. Die Implementierung des semantischen Schnitts von Terminbeschreibungen als `semantic_intersectionDATE/2` kann nun ähnlich wie für `semantic_intersectionSimpleDUR/2` realisiert werden. Die Komplementbildung ist etwas komplizierter; sie wird durch folgendes Prädikat umgesetzt:

```
semantic_complementDATE( Semantics, Complement) :-
  get_minimal_timepoint( MinC),
  get_maximal_timepoint( MaxC),
  EntireP = [[i(MinC,MaxC)]],
  get_infty_minutes( MaxN),
  EntireD = [[i(0,MaxN)]],
  findall( Set,
  (
    member( M, Semantics),
    findall( Q,
    (
      member( (Pf,Pt,Pd,D), M),
      (
        ( semantic_complementP( Pf, ComplPf),
          Q = (ComplPf,EntireP,EntireP,EntireD) ) ;
        ( semantic_complementP( Pt, ComplPt),
          Q = (EntireP,ComplPt,EntireP,EntireD) ) ;
        ( semantic_complementP( Pd, ComplPd),
          Q = (EntireP,EntireP,ComplPd,EntireD) ) ;
        ( semantic_complementSimpleDUR( D, ComplD),
          Q = (EntireP,EntireP,EntireP,ComplD) )
      )
    ),
    Set)
  ),
  Complement).
```

Hierbei werden durch `get_minimal_timepoint/1` und `get_maximal_timepoint/1` Werte für einen frühesten und einen spätesten Zeitpunkt (in der Theorie: $-\infty_{\mathcal{C}}$ beziehungsweise $\infty_{\mathcal{C}}$) gewonnen. Entsprechendes gilt für das Prädikat `get_infty_minutes/1` zur Bestimmung der maximalen Länge einer Dauer in Minuten.

Die Berechnung der semantischen Vereinigung einer Liste von Objekten erfolgt in jedem der vier in diesem Teil des Kapitels angesprochenen semantischen Bereiche auf die gleiche Weise, weshalb das allgemeine Prädikat `semantic_union/2` angegeben wird.

```
semantic_union( ListOfSemantics, Union) :-
  findall( Set,
  (
    members( Ms, ListOfSemantics),
    multiple_append( Ms, Set)
  ),
  Union).
```

Durch `multiple_append/2` werden dabei alle Listen einer Liste miteinander verkettet.

8.3.7 Inferenzen

Unabhängig von der semantischen Interpretation, deren Implementierung Abschnitt 8.4 gewidmet sein wird, können die im letzten Kapitel beschriebenen Inferenzen auf der Grundlage der gerade behandelten semantischen Bereiche implementiert werden.

Das Testen auf Inkonsistenz funktioniert bei Dauerangaben (allgemeiner oder vereinfachter Bereich) und bei Zeitpunktangaben auf die gleiche Weise. Beispielhaft sei nur das Prädikat für letztere angegeben:

```
inconsistentP( []).

inconsistentP( Lists) :-
  forall( member( List, Lists), List = []).
```

Bevor die Inkonsistenz von Interpretation von Ausdrücken der Kategorie `DATE` behandelt werden kann, muß zunächst bestimmt werden, wann ein Quadrupel zur Terminspezifikation inkonsistent ist. Das folgende Prädikat ist gerade dann erfolgreich, wenn mit `quadruple_generates_date/2` keine Termin-Tripel generiert werden können.

```
inconsistentQuadruple( Q) :-
  \+ quadruple_generates_date( Q, _).
```

Mit Hilfe von `inconsistentQuadruple/1` kann die Inkonsistenzbedingung über dem semantischen Bereich von L(`DATE`) dann recht einfach formuliert werden:

```
inconsistentDATE( []).

inconsistentDATE( Lists) :-
  forall( member( List, Lists),
  (
    List = [] ;
    forall( member( Q, List), inconsistentQuadruple( Q))
  )).
```

Kann ein semantischer Schnitt berechnet werden (das ist nur für den allgemeinen semantischen Bereich von L(`DURATION`) *nicht* der Fall), so kann unmittelbar ein Prädikat angegeben werden, welches testet, ob zwei semantische Objekt miteinander vereinbar sind. Für die Zeitpunktangaben ist dies beispielsweise `unifiableP/2`.

```
unifiableP( P1, P2) :-
  semantic_intersectionP( [P1,P2], P),
  \+ inconsistentP( P).
```

Spezialisierung und Generalisierung wurden im letzten Kapitel unter Verwendung der Äquivalenzrelationen definiert. In der Praxis ist eine direkte Auswertung der Spezialisierungsrelation jedoch effizienter. Für den semantischen Bereich von L(`POINT`) werden sowohl Äquivalenz als auch Spezialisierung mit Hilfe von `intC_coverage/2` implementiert. Ein Ausdruck (genauer: seine Interpretation) `P1` ist spezieller als ein Ausdruck `P2`, wenn es in `P1` eine Liste `M1` und in `P2` eine Liste `M2` gibt, so daß es wiederum in `M1` ein Intervall `I1` gibt, welches von den Intervallen in `M2` vollständig überdeckt wird. Für die Äquivalenzrelation muß zusätzlich getestet werden, ob es auch in `M2` ein `I2` gibt, das von den Intervallen in `M1` überdeckt wird. Das Generalisierungsprädikat muß lediglich das Spezialisierungsprädikat mit vertauschten Argumenten aufrufen.

```
equalP( P1, P2) :-
  member( M1, P1),
  member( M2, P2),
  forall( member( I1, M1), intC_coverage( I1, M2)),
  forall( member( I2, M2), intC_coverage( I2, M1)).

specializationP( P1, P2) :-
  member( M1, P1),
  member( M2, P2),
  forall( member( I1, M1), intC_coverage( I1, M2)).

generalizationP( P1, P2) :-
  specializationP( P2, P1).
```

Für den vereinfachten semantischen Bereich für Dauerangaben sowie für Terminbeschreibungen werden entsprechende Prädikate eingeführt. Für die Terminbeschreibungen unterscheidet sich die Implementierung ein wenig; beispielhaft sei hier nur das Prädikat für die Spezialisierungsrelation angegeben.

```
specializationDATE( A1, A2) :-
  member( M1, A1),
  member( M2, A2),
  forall( ( member( Q1, M1), quadruple_generates_date( Q1, D)),
          ( member( Q2, M2), quadruple_describes_date( Q2, D))).
```

Schließlich wurde im letzten Kapitel noch der Begriff der semantischen Differenz eingeführt. Die Implementierung der entsprechenden Prädikate ist trivial, wenn Schnitt und Komplement bereits gegeben sind. Über dem vereinfachten semantischen Bereich für Dauerangaben etwa wird es folgendermaßen implementiert:

```
semantic_differenceSimpleDUR( S1, S2, D) :-
  semantic_complementSimpleDUR( S2, S2c),
  semantic_intersectionSimpleDUR( [S1,S2c], D).
```

8.4 Semantische Interpretation

Zur semantischen Interpretation von Ausdrücken der TEMPORAL EXPRESSION LANGUAGE, also zur Abbildung auf den jeweiligen semantischen Bereich durch Anwendung einer der Funktionen $\mathcal{I}_N$, $\mathcal{I}_{DUR}$, $\mathcal{I}_P$ und $\mathcal{I}$ werden im folgenden die PROLOG-Prädikate `interpretationN/2`, `interpretationDUR/2`, `interpretationP/2` und `interpretationDATE/2` angegeben. Diese Funktionen beziehungsweise Prädikate beziehen sich auf Ausdrücke der TEL-Kernsprache. Ob ein bestimmter Ausdruck einer bestimmten Kernkategorie angehört, kann durch Anwendung von `ccategory/2` (siehe Abschnitt 8.2) abgefragt werden. In den meisten Fällen ergibt sich die Kategorie eines Ausdrucks (mitunter auch die seiner Teilausdrücke) bereits aus dem Hauptfunktor.

Wird `ccategory/2` an zweiter Position mit einer Variablen aufgerufen, so wird diese mit der (allgemeinsten) Kategorie des an erster Stelle übergebenen Ausdrucks instanziiert. Somit ist es möglich, das allgemeine Prädikat `interpretation/2` zu implementieren, welches zunächst mit Hilfe von `ccategory/2` die Kategorie des an erster Position gegebenen Ausdrucks bestimmt und dann das geeignete Interpretationsprädikat auswählt, welches das entsprechende semantische Objekt liefert, das dann über die zweite Position als Ergebnis 'ausgegeben' werden kann.

8.4.1 Interpretation von Zahlenangaben

Zahlenangaben werden mit Hilfe des Prädikats `interpretationN/2` interpretiert, das wie folgt implementiert ist:

```
interpretationN( N1:N2:N3, Q) :-
  ccategory( N1, 'NUMBER'),
  ccategory( N2, 'NUMBER'),
  ccategory( N3, 'NUMBER'),
  Q is N1 + N2 / N3.
```

```
interpretationN( N1:N2, Q) :-
  ccategory( N1, 'NUMBER'),
  ccategory( N2, 'NUMBER'),
  Q is N1 / N2.

interpretationN( N, N):-
  ccategory( N, 'INTEGER').
```

Die letzte Klausel beinhaltet hier auch die Kategorien `NUMBER`, `NUMBER0` sowie den ersten (beziehungsweise hier letzten) Fall der Kategorie `FRACTION`.

8.4.2 Interpretation von Dauerangaben

Ausdrücke der Kategorie `DURATION` können unter Verwendung des Prädikats `interpretationDUR/2` interpretiert werden. Ich werde die Ausführung nur für einige der Ausdrucksformen angeben, da sie sich ohnehin oft kaum unterscheiden.

Im einfachsten Falle wird eine Dauerangabe in Form eines Ausdrucks der Kategorie `BASIC_DUR` gemacht. Die folgende Klausel zeigt, wie solch ein Ausdruck für den Fall der Zeiteinheit `days` zu verarbeiten ist (für die anderen Einheiten aus L(`UNIT`) gilt entsprechendes):

```
interpretationDUR( dur(Term,days), [[pi(D,D)]]) :-
  interpretationN( Term, Q),
  multiply( Q, (0,1440), D).
```

Bei der Interpretation von Ausdrücken der Kategorie `RANGE` muß für jedes auftretende Paar von Pseudo-Intervallen ein neues Pseudo-Intervall gebildet werden, *falls* dieses überhaupt möglich ist:

```
interpretationDUR( range(Term1,Term2), Semantics) :-
  interpretationDUR( Term1, S1),
  interpretationDUR( Term2, S2),
  findall( Set,
  (
    member( M1, S1),
    member( M2, S2),
    findall( pi(D11,D22),
    (
      member( pi(D11,_), M1),
      member( pi(_,D22), M2),
      inPseudoIntD( pi(D11,D22))
    ),
    Set)
  ),
  Semantics).
```

Als Beispiel für Ausdrücke der Kategorie `OPEN_RANGE` sei die Klausel für linksseitig offene 'exklusiv'-Bereiche gegeben:

```
interpretationDUR( less(Term), Semantics) :-
  interpretationDUR( Term, S),
  findall( Set,
  (
    member( M, S),
    findall( pi((0,0),D1),
    (
      member( pi(D1,_), M)
    ),
    Set)
  ),
  Semantics).
```

Für die rechtsseitig offenen Bereiche müssen Maximalwerte für Monats- und Minutenangabe festgelegt werden.

Die nächste Klausel zeigt, wie modifizierte Dauerangaben interpretiert werden. Es sei daran erinnert, daß die Kategorie `MODIFIED_DUR` ausschließlich in der Kernsprache definiert wurde (siehe hierzu Kapitel 5).

```
interpretationDUR( modified_dur(Term1,Term2,Term3), Semantics) :-
  interpretationN( Term1, Q1),
  interpretationN( Term2, Q2),
  interpretationN( Term3, S),
  findall( Set,
  (
    member( M, S),
    findall( pi(ModD1,ModD2),
    (
      member( pi(D1,D2), M),
      multiply( Q1, D1, ModD1),
      multiply( Q2, D2, ModD2),
      inPseudoIntD( pi(ModD1,ModD2))
    ),
    Set)
  ),
  Semantics).
```

Durch den `set`-Operator wird eine Disjunktion von Dauerausdrücken angedeutet. Wenn sichergestellt ist, daß `interpretationDUR/2` ausschließlich mit Ausdrücken der Kategorie `DURATION` aufgerufen wird, so steht fest, daß der `set` folgende Teilausdruck vom Typ `DURATION+` ist. Bei der Interpretation muß im wesentlichen das Prädikat `semantic_union/2` zur Berechnung der semantischen Vereinigung einer Liste von interpretierten Ausdrücken aufgerufen werden.

```
interpretationDUR( set(Terms), Semantics) :-
  findall( S,
  (
    member( Term, Terms),
    interpretationDUR( Term, S)
  ),
  Ss),
  semantic_union( Ss, Semantics).
```

Bei der Interpretation des `one_out_of`-Operators ist die Situation noch etwas einfacher. Hier müssen die interpretierten Alternativmengen lediglich zu einer Liste verkettet werden (durch `multiple_append/2`).

```
interpretationDUR( one_out_of(Terms), Semantics) :-
  findall( S,
  (
    member( Term, Terms),
    interpretationDUR( Term, S)
  ),
  Ss),
  multiple_append( Ss, Semantics).
```

8.4.3 Interpretation von Zeitpunktangaben

Ausdrücke der Kategorien `POINT` und `GENERIC` werden durch das Prädikat `interpretationP/2` interpretiert. Der einfachste dieser Ausdrücke ist `now`. Durch die Hilfsfunktion `get_now/1` kann stets der aktuelle Zeitpunkt in Form eines Quintupels generiert werden. Alternativ besteht die Möglichkeit, einen vom Benutzer vorher festgesetzten Zeitpunkt zu generieren, um so auch in der Vergangenheit dokumentierte Dialoge im Nachhinein analysieren zu können. Die Interpretation von `now` besteht also im wesentlichen aus einem Aufruf von `get_now/1`:

```
interpretationP( now, [[i(T,T)]]) :-
  get_now( T).
```

Durch alle weiteren einfachen Ausdrücke (abgesehen von `year:YEAR`) sowie die generischen Ausdrücke werden bestimmte Merkmale eines Zeitpunktes festgeschrieben, während die übrigen Merkmale undefiniert bleiben, also alle Möglichkeiten durchlaufen müssen. Hierzu werden die Funktionen `get_a_year/1`, `get_a_month/1`, `get_a_day/1`, etc. implementiert, die bei Backtracking alle zulässigen Werte liefern. So generiert beispielsweise `get_a_day/1` die Folge der Zahlen von 1 bis 31. Ob solch eine Zahl auch tatsächlich eine gültige Tagesangabe darstellt, muß im jeweiligen Falle noch zusätzlich überprüft werden (zum Beispiel mit `inC/1`). Diese Zahlenmengen sollten aus Effizienzgründen natürlich so gering wie möglich gehalten werden (siehe hierzu auch die Bemerkungen in Abschnitt 8.5).

Die Implementierung für Ausdrücke der Kategorien `BASIC` und `GENERIC` folgt immer dem gleichen Muster, weshalb hier nur zwei Beispiele angegeben werden. Folgende Klausel realisiert die Interpretation von Ausdrücken der Form `tod:TOD`.

```
interpretationP( tod:H:I, [Set0,Set12]) :-
  H12 is H + 12,
  findall( i((Y,M,D,H,I),(Y,M,D,H,I)),
  (
    get_a_year( Y),
    get_a_month( M),
    get_a_day( D),
    inC( (Y,M,D,H,I))
  ),
  Set0),
  findall( i((Y,M,D,H12,I),(Y,M,D,H12,I)),
  (
    get_a_year( Y),
    get_a_month( M),
    get_a_day( D),
    inC( (Y,M,D,H12,I))
  ),
  Set12).
```

Unter den generischen Ausdrücken ist die Interpretation von `week` wohl am kompliziertesten. In folgender Klausel werden zunächst alle Zeitpunkte `T`, die an einem Donnerstag um 0 Uhr liegen, generiert, und sodann durch Addition und Subtraktion die zugehörigen Werte für Montag, 0 Uhr und Sonntag, 23.59 Uhr gewonnen.

```
interpretationP( week, [Set]) :-
  findall( i(T1,T2),
  (
    get_a_year( Y),
    get_a_month( M),
    get_a_day( D),
    T = (Y,M,D,0,0),
    inC( T),
    weekday( T, 4),
    subtract( T, (0,4320), T1),
    add( T, (0,5759), T2)
  ),
  Set).
```

Auch für die zusammengesetzten Ausdrücke der Kategorie `POINT` werde ich nur eine Auswahl von Klauseln näher beschreiben, da sie sich untereinander teilweise stark ähneln. In einigen Fällen werden die bereits in Abschnitt 8.3.6 erwähnten Hilfsfunktionen `get_maximal_timepoint/1` und `get_minimal_timepoint/1` verwendet werden.

Durch Lambda-Funktionen modifizierte Ausdrücke werden durch folgende Klausel interpretiert:

```
interpretationP( modified(lambda(Term1,Term2),Term3), Semantics) :-
  interpretationP( Term3, S),
```

```
  findall( Set,
  (
    member( M, S),
    findall( i(T1,T2),
    (
      member( I, M),
      f( I, Term1, T1),
      f( I, Term2, T2)
    ),
    Set)
  ),
  Semantics).
```

Hierbei implementiert `f/3` die in Kapitel 6 bei der Definition der Interpretationsfunktion $\mathcal{I}_P$ über $L($`MODIFIED`$)$ einführte Hilfsfunktion f zur Berechnung der neuen Intervallgrenzen. In der zweiten Zeile wird die Interpretation des Teilausdrucks `Term3` mit `S` unifiziert. Dann werden alle Alternativmengen in `S` und darin wiederum alle Intervalle durchlaufen, um jeweils ein modifiziertes Intervall generieren zu können.

Als Beispiel für die einseitig offenen Intervalle seien Ausdrücke mit dem Funktor `in_after` herausgegriffen. (Es sei daran erinnert, daß in Kapitel 5 angedeutet wurde, daß Ausdrücke der Kategorie `LIMIT` nicht unbedingt in die Kernsprache mitaufgenommen werden müßten.) Um die Intervalle aufbauen zu können muß mit Hilfe von `get_maximal_timepoint/1` zunächst ein maximaler Zeitpunkt (statt $\infty_{\mathcal{C}}$) festgelegt werden.

```
interpretationP( in_after(Term), Semantics) :-
  interpretationP( Term, S),
  get_maximal_timepoint( MaxC),
  findall( Set,
  (
    member( M, S),
    findall( i(T1,MaxC),
    (
      member( i(T1,_), M)
    ),
    Set)
  ),
  Semantics).
```

Die Klausel für Ausdrücke mit `between` ist der letzten sehr ähnlich. Statt eines festen Maximalpunktes werden als rechte Randpunkte hier die Werte aus dem zweiten Teilausdruck ausgelesen. Das Backtracking innerhalb der `findall/3`-Strukturen macht es überflüssig, zusätzliche Schleifenkonstrukte zu implementieren; es reicht aus, `member/2` zweimal unmittelbar nacheinander für die verschiedenen Listen aufzurufen.

```
interpretationP( between(Term1,Term2), Semantics) :-
  interpretationP( Term1, S1),
```

```
  interpretationP( Term2, S2),
  findall( Set,
  (
    member( M1, S1),
    member( M2, S2),
    findall( i(T11,T22),
    (
      member( i(T11,_), M1),
      member( i(_,T22), M2),
      inIntC( i(T11,T22))
    ),
    Set)
  ),
  Semantics).
```

Ersetzt man `between` durch `min_between`, so muß auf das Ergebnis noch die Minimierungsfunktion μ in Form des Prädikats `minimize/2` angewandt werden.

Bei Verschiebungen müssen zunächst der `DURATION`-Teilausdruck und der `POINT`-Teilausdruck interpretiert werden. Dann können die Randpunkte der neuen Zeitpunkt-Intervalle mit Hilfe der definierten Prädikate zur Addition (für `neg_shift` Subtraktion) bestimmt werden. Ob die so bestimmten Randpunkte ein gültiges Intervall definieren, wird mit `inIntC/1` überprüft.

```
interpretationP( pos_shift(Term1,Term2), Semantics) :-
  interpretationDUR( Term1, S1),
  interpretationP( Term2, S2),
  findall( Set,
  (
    member( M1, S1),
    member( M2, S2),
    findall( i(NewT1,NewT2),
    (
      member( pi(D1,D2), M1),
      member( i(T1,T2), M2),
      add( T1, D1, NewT1),
      add( T2, D2, NewT2),
      inIntC( i(NewT1,NewT2))
    ),
    Set)
  ),
  Semantics).
```

Als Beispiel für Ausdrücke der Kategorie `RELATED` mit einem Argument der Form `int:DURATION` gebe ich die Klausel für den Funktor `the_around` an. Zunächst werden die beiden Teilausdrücke interpretiert. Dann werden alle Zeitpunkt-Intervalle der Interpretation des zweiten Terms durchlaufen und für jedes von ihnen wird ein neues generiert. Hierzu wird mit `meanIntC/2` zunächst der Mittelwert des ursprünglichen Intervalls berechnet. Aus einem ebenfalls gegebenen Pseudo-Intervall wird die größere Dauer ausgewählt und mit Hilfe von `multiply/3` halbiert. Beachte, daß hierbei eine rationale Zahl mit einem Dauer-Paar multipliziert wird. Von dem berechneten Mittelpunkt wird diese halbe Länge subtrahiert, um den linken Randpunkt des neuen Intervalls zu erhalten; zur Berechnung den rechten Randpunkt muß sie addiert werden.

```
interpretationP( the_around(int:Term1,Term2), Semantics) :-
  interpretationDUR( Term1, S1),
  interpretationP( Term2, S2),
  findall( Set,
  (
    member( M1, S1),
    member( M2, S2),
    findall( i(T1,T2),
    (
      member( pi(_,D2), M1),
      member( I, M2),
      meanIntC( I, MeanI),
      multiply( 0.5, D2, HalfD2),
      subtract( MeanI, HalfD2, T1),
      add( T1, D2, T),
      subtract( T, (0,1), T2)
    ),
    Set)
  ),
  Semantics).
```

Die nächste Klausel zeigt die Interpretation eines `RELATED`-Ausdrucks mit einem `COUNTABLE1`-Argument anhand von Ausdrücken mit dem Funktor `the_after`.

```
interpretationP( the_after(N,Term1,Term2), Semantics) :-
  ccategory( Term1, 'COUNTABLE1'),
  interpretationP( Term1, S1),
  interpretationP( Term2, S2),
  findall( Set,
  (
    member( M1, S1),
    member( M2, S2),
    findall( I1,
    (
      member( i(_,T), M2),
      get_Nth_after( N, T, M1, I1)
    ),
    Set)
  ),
  Semantics).
```

Durch die Bedingung `ccategory( Term1, 'COUNTABLE1')` wird sichergestellt, daß das zweite Argument in der Tat nicht die Form `int:DURATION` hat. Sodann werden die Interpretation dieses Arguments und die des Referenzpunktes (also des dritten Arguments) bestimmt. Bei der weiteren Berechnung kann vorausgesetzt werden, daß die Intervalle in der Interpretation des `COUNTABLE1`-Ausdrucks aufsteigend geordnet sind. Dies vereinfacht die Implementierung des Prädikats `get_Nth_after/4`, welches dazu dient, zu einer gegebenen Zahl `N` und einem Referenzzeitpunkt `T` aus einer Liste `M1` von Intervallen das *`N`te* Intervall `I1` auszuwählen, dessen linker Randpunkt größer oder gleich `T` ist.

```
get_Nth_after( 1, T, [i(T1,T2)|_], i(T1,T2)) :-
  leqC( T, T1).

get_Nth_after( N, T, [i(T1,_)|Rest], I) :-
  leqC( T, T1), !,
  N > 1,
  N1 is N - 1,
  get_Nth_after( N1, T, Rest, I).

get_Nth_after( N, T, [_|Rest], I) :-
  get_Nth_after( N, T, Rest, I).
```

Auf die Beschreibung der Klauseln für Ausdrücke aus L(`ORDINAL`) kann hier verzichtet werden, da sie denen für Ausdrücke der Kategorie `RELATED` sehr ähnlich sind. Was bleibt, sind die Top-Level-Ausdrücke der Kategorie `POINT`. Disjunktion und die Verkettung von Alternativen werden wie auch schon für L(`DURATION`) umgesetzt. Auch bei Konjunktion und Negation müssen lediglich die bereits implementierten Funktionen für semantischen Schnitt und Komplement angewendet werden. Nur der Sonderfall der leeren Liste muß gesondert behandelt werden.

```
interpretationP( [], [[i(MinC,MaxC)]]) :-
  get_minimal_timepoint( MinC),
  get_maximal_timepoint( MaxC).

interpretationP( Terms, Semantics) :-
  ccategory( Terms, 'POINT*'),
  findall( S,
  (
    member( Term, Terms),
    interpretationP( Term, S)
  ),
  Ss),
  semantic_intersectionP( Ss, Semantics).

interpretationP( not(Term), Semantics) :-
  interpretationP( Term, S),
  semantic_complementP( S, Semantics).
```

Als Beispiel für die Implemetierung der Interpretation von quantifizierten Ausdrücken gebe ich die Klausel für den Fall an, daß es sich bei dem Quantor um eine Zahl `N` handelt. Hier müssen für jede Alternativmenge `M` all deren Teilmengen mit `N` Elementen ausgewählt werden.

```
interpretationP( quantified(N,Term), Semantics) :-
  ccategory( N, 'NUMBER'),
  interpretationP( Term, S),
  findall( Set,
  (
    member( M, S),
    choose_n_elements( N, M, Set)
  ),
  Semantics).
```

Das Hilfsprädikat `choose_n_elements/3` liefert nacheinander alle Möglichkeiten, aus der übergebenen Liste die spezifizierte Anzahl von Elementen auszuwählen.

```
choose_n_elements( N, [Element|List], [Element|Elements]) :-
  N > 1,
  N1 is N - 1,
  choose_n_elements( N1, List, Elements).

choose_n_elements( 1, [Element|_], [Element]).

choose_n_elements( N, [_|List], Elements) :-
  choose_n_elements( N, List, Elements).
```

Jede Auswahl wird hierbei nur genau einmal generiert, was durch folgendes Beispiel demonstriert wird:

```
| ?- choose_n_elements( 2, [1,2,3,4], SubList).
SubList = [1,2] ;

SubList = [1,3] ;

SubList = [1,4] ;

SubList = [2,3] ;

SubList = [2,4] ;

SubList = [3,4] ;

no
```

8.4.4 Interpretation von Terminbeschreibungen

Ausdrücke der Sprache L(`DATE`) werden unter Verwendung des Prädikats `interpretationDATE/2` interpretiert. Die grundlegenden Ausdrücke dieser Sprache sind die der Formen `from:POINT`, `to:POINT`, `during:POINT` und `for:DURATION`. Wie diese Ausdrücke interpretiert werden, zeigen bespielhaft die Klauseln für die Label `from` und `for`:[4]

[4]Das Prädikat `simplify/2` implementiert die Funktion σ und wurde in Abschnitt 8.3.3 angegeben.

```
interpretationDATE( from:Term, [[(P,EntireP,EntireP,[[i(0,MaxN)]])]]) :-
  interpretationP( Term, P),
  get_minimal_timepoint( MinC),
  get_maximal_timepoint( MaxC),
  EntireP = [[i(MinC,MaxC)]],
  get_infty_minutes( MaxN).

interpretationDATE( for:Term, [[(EntireP,EntireP,EntireP,SimpleD)]]) :-
  interpretationDUR( Term, D),
  simplify( D, SimpleD),
  get_minimal_timepoint( MinC),
  get_maximal_timepoint( MaxC),
  EntireP = [[i(MinC,MaxC)]].
```

Die Implementierung von Klauseln für die Konjunktion, die Disjunktion, die Negation und den `one_out_of`-Operator erfolgt in strenger Analogie zu `interpretationP/2`. Die benötigten Prädikate, etwa zur Berechnung eines semantischen Schnitts über L(`DATE`) wurden bereits in Abschnitt 8.3.6 besprochen.

Schließlich muß noch die Verarbeitung von Label-Listen behandelt werden. In der folgenden Klausel wird in der ersten Zeile sichergestellt, daß es sich tatsächlich um eine Label-Liste handelt. Dann wird der angegebene `POINT`-Ausdruck *einmal* interpretiert. Je nachdem, welche Label vorkommen, werden nun Quadrupel mit dieser Interpretation an der jeweiligen Position erzeugt.

```
interpretationDATE( Labels:Term, Semantics) :-
  ccategory( Labels, 'POINTLABEL+'),
  interpretationP( Term, P),
  get_maximal_timepoint( MaxC),
  get_minimal_timepoint( MinC),
  EntireP = [[i(MinC,MaxC)]],
  get_infty_minutes( MaxN),
  EntireD = [[i(0,MaxN)]],
  ( member( from, Labels)
    -> S1 = [[(P,EntireP,EntireP,EntireD)]] ; S1 = [] ),
  ( member( to, Labels)
    -> S2 = [[(EntireP,P,EntireP,EntireD)]] ; S2 = [] ),
  ( member( during, Labels)
    -> S3 = [[(EntireP,EntireP,P,EntireD)]] ; S3 = [] ),
  multiple_append( [S1,S2,S3], Semantics).
```

8.4.5 Rückgewinnung von TEL-Ausdrücken

Obschon ich im theoretischen Teil nicht darauf eingegangen bin, mag es in der Praxis notwendig werden, zu einem semantischen Objekt den (oder vielmehr *einen*) entsprechenden TEL-Ausdruck zurückzugewinnen. Da die auf L(`DURATION`), L(`POINT`) und L(`DATE`) definierten Intepretationsfunktionen offensichtlich nicht injektiv sind, kann aber keine Umkehrfunktion angegeben werden. Es ist jedoch möglich, für ein semantisches Objekt einen ihm entsprechenden 'kanonischen' Ausdruck zu berechnen. Hierzu wird das PROLOG-Prädikat `sem2syn/2` implementiert. Je nach Art des semantischen Bereiches wird wird ein entsprechendes Subprädikat aufgerufen. Im Falle des mit Ausdrücken der Kategorie `POINT` assoziierten Bereiches (Überprüfung mit `inSemanticDomainP/1`) ist dies `sem2synP/2`. An erster Stelle wird ein semantisches Objekt zur Umschreibung eines Zeitpunktes, also eine Liste von Listen von Zeitpunkt-Intervallen, angegeben; die an zweiter Stelle übergebene Variable wird dann mit einem diesem semantischen Objekt entsprechenden TEL-Ausdruck unifiziert.

Hat die Hauptliste nur ein Element, so erübrigt sich im Zielausdruck der `one_out_of`-Operator. Das Prädikat zur Behandlung einer Alternativmenge (beziehungsweise -liste) heißt `sem2synP_set/2`.

```
sem2synP( [Set], Term) :- !,
  sem2synP_set( Set, Term).

sem2synP( Semantics, one_out_of(Terms)) :-
  findall( Term,
  (
    member( Set, Semantics),
    sem2synP_set( Set, Term)
  ),
  Terms).
```

Die an `sem2synP_set/2` übergebenen Listen entsprechen Alternativmengen, also Mengen von Intervallen. Gibt es mehrere Intervalle, so werden die entsprechenden Ausdrücke durch den `set`-Operator verbunden. Wenn es hingegen nur ein Intervall gibt, so kann dieses direkt mit Hilfe von `sem2synP_interval/2` umgeformt werden.

```
sem2synP_set( [Interval], Term) :- !,
  sem2synP_interval( Interval, Term).

sem2synP_set( Set, set(Terms)) :-
  findall( Term,
  (
    member( Interval, Set),
    sem2synP_interval( Interval, Term)
  ),
  Terms).
```

Ein Intervall hat die Struktur `i(T1,T2)`, wobei `T1` und `T2` Quintupel sind, die Zeitpunkte, also Elemente aus $\mathcal{C}$, repräsentieren. Solch ein Intervall wird im Normalfall in einen Ausdruck mit `between` umgeformt. Nur wenn die beiden Randpunkte identisch sind, reicht ein einfacher Ausdruck aus, der durch `sem2synP_timepoint/2` bestimmt wird.

```
sem2synP_interval( i(T,T), Term) :- !,
  sem2synP_timepoint( T, Term).
```

```
sem2synP_interval( i(T1,T2), between(Term1,Term2)) :-
  sem2synP_timepoint( T1, Term1),
  sem2synP_timepoint( T2, Term2).
```

Das Prädikat `sem2synP_interval/2` läßt sich leicht so erweitern, daß ein Intervall, das gerade einem Tag entspricht, nicht als Intervall mit `between` sondern als einfache Tagesangabe ausgegeben wird. Gleiches gilt für Intervalle, die einem ganzen Monat oder einem ganzen Jahr entsprechen. Außerdem können bei Intervallen, die um 0 Uhr beginnen und an einem späteren Tag um 23.59 enden, die Uhrzeitangaben in den Randpunkten weggelassen werden. Entsprechendes gilt für die Tagesangaben, wenn ein Intervall ganze Monate umfaßt, beziehungsweise sogar für Monatsangaben, wenn es ganze Jahre umfaßt. (Die entsprechenden sechs zusätzlichen Klauseln von `sem2synP_interval/2` sind hier nicht abgedruckt.)

Ein in Form eines Quintupels aus Jahres-, Monats-, Tages-, Stunden- und Minutenangabe spezifizierter Zeitpunkt kann unmittelbar in eine Liste von einfachen `POINT`-Ausdrücken umgeformt werden. Für die Monatsangabe muß eine Zahl in ein Akronym (`jan`, `feb`, etc.) umgewandelt werden. Liegt die Stundenangabe zwischen 12 und 23, so muß 12 subtrahiert werden und der Ausdruck `pod:pm` der Liste hinzugefügt werden (andernfalls `pod:am`).

```
sem2synP_timepoint( (Y,M,D,H,I), Result) :-
  month2number( Month, M), !,
  ( H < 12 -> POD = am, Hour = H ; POD = pm, Hour is H - 12 ),
  Result = [year:Y,month:Month,dom:D,pod:POD,tod:Hour:I].
```

Im folgenden Beispiel wird durch `interpretation/2` die Semantik von `of(2,dow:sun,[month:jun,year:1998])` berechnet. Der zweite Sonntag im Juni des Jahres 1998 ist der 14. Juni 1998. Wird auf das Ergebnis dieser Berechnung `sem2syn/2` angewandt, so wird die Variable `Syntax` mit dem entsprechenden kanonischen TEL-Ausdruck unifiziert.

```
| ?- interpretation( of(2,dow:sun,[month:jun,year:1998]), Semantics).
Semantics = [[i((1998,6,14,0,0),(1998,6,14,23,59))]]

| ?- sem2syn( [[i((1998,6,14,0,0),(1998,6,14,23,59))]], Syntax).
Syntax = [year : 1998,month : jun,dom : 14]
```

Im nächsten Beispiel wird zunächst der Ausdruck `next(week)` reduziert. Sodann wird das Ergebnis der Reduktion interpretiert (das Beispiel wurde am 8. Juni 1998 erzeugt). Der folgende Aufruf von `sem2syn/2` liefert den kanonischen TEL-Ausdruck, der `next(week)` in diesem Augenblick entspricht.

```
| ?- reduce( next(week), R), interpretation( R, I), sem2syn( I, S).
R = the_after(1,week,now) ,
I = [[i((1998,6,15,0,0),(1998,6,21,23,59))]] ,
S = between([year : 1998,month : jun,dom : 15],
[year : 1998,month : jun,dom : 21])
```

Die Prädikate `sem2synDUR/2` für Dauerangaben und `sem2synDATE/2` für Terminbeschreibungen können auf analoge Weise implementiert werden. Um auch vereinfachte Dauerangaben in TEL-Ausdrücke zurückverwandeln zu können, wurde als Gegenstück zu `simplify/2` das Prädikat `unsimplify/2` implementiert, um vom vereinfachten auf den allgemeinen Bereich abbilden zu können.

8.5 Mögliche Verbesserungen

Das vorgestellte System wurde unter UNIX mit QUINTUS PROLOG sowie unter WINDOWS mit LPA PROLOG getestet. Es arbeitet bisher nicht schnell genug, um in VERBMOBIL integriert werden zu können. Dies ist vor allem auf die Implementierung der Interpretation von Zeitpunktangaben zurückzuführen. Schon für sehr einfache Ausdrücke werden mitunter Listen mit sehr vielen Zeitpunkt-Intervallen generiert. Für den beschriebenen Prototyp wurde die Menge 'aller' Zeitpunkte auf zwei Jahre beschränkt. Das bedeutet, daß beispielsweise der TEL-Ausdruck `dow:mon` auf die Liste aller Zeitpunkt-Intervalle, die einen Montag innerhalb dieser zwei Jahre repräsentieren, abbgebildet wird. Dies sind immerhin über 100 Intervalle. Wird dieser Ausdruck nun mit einem Ausdruck vergleichbarer Komplexität geschnitten, so müssen etwa 100 mal 100 Schnittoperationen über Zeitpunkt-Intervallen vollzogen werden. Bei Uhrzeitangaben ist die Komplexität noch um ein Mehrfaches höher. Eine bestimmte Uhrzeit kommt an jedem Tag zweimal vor (*am* und *pm*), das heißt für einen Zeitraum von zwei Jahren müssen fast 1500 Intervalle erzeugt werden.

Trotzdem, denke ich, kann der vorgestellte Prototyp zu einem effizienten, einsatzfähigen System ausgebaut werden. Im folgenden betrachte ich einige Änderungsmöglichkeiten, die bei einer Erhaltung der beschriebenen Grundstruktur das Laufzeitverhalten deutlich verbessern könnten.

Bei der gegenwärtigen Darstellung bereiten Uhrzeitangaben die größten Schwierigkeiten; für sie werden mehr Intervalle als für jeden anderen Ausdruck erzeugt. Eine Lösung dieses Problems könnte darin bestehen, Zeitpunkte nicht mehr nur als Quintupel (Elemente aus $\mathcal{C}$) zu repräsentieren, sondern im Falle von Uhrzeiten ohne Tages- oder sonstige Angaben auch Paare aus Stunden und Minuten zuzulassen. Ein solches Paar stünde dann für alle Zeitpunkte des Kalenders, die gerade die angegeben Stunden- und Minutenwerte haben. Die Implementierung der Prädikate, deren Argumente Zeitpunkte sind, müßte entsprechend angepaßt werden. Dies betrifft insbesondere die Notation von Intervallen über $\mathcal{C}$ sowie die über ihnen erklärte Schnittoperation.

Momentan ist die Menge der zulässigen Zeitpunkte fest implementiert. Sie werden durch die Prädikate `get_a_year/1`, `get_a_month/1` etc. generiert. Durch `get_maximal_timepoint/1` und `get_minimal_timepoint/1` können außerdem die (ebenfalls im Programm festgeschriebenen) Werte für den spätesten und den frühesten Zeitpunkt abgerufen werden. Eine Einschränkung dieses 'Fensters' (der Zeitpunkte-Grundgesamtheit) zieht auch im gegenwärtigen System unmittelbar eine Verbesserung des Laufzeitverhaltens nach sich. Idealerweise sollte eine solche Einschränkung abhängig vom Dialogverlauf und möglichst präzise erfolgen. Hierbei ist allerdings zu berücksichtigen, daß zum Beispiel die Semantik von `of(40,week,year)` nur dann korrekt berechnet werden kann, wenn mindestens der Zeitraum vom Beginn eines Jahres bis zu seiner 40. Woche innerhalb der Zeitpunkte-Grundgesamtheit liegt. Gegebenenfalls könnte man das jeweilige Zeitpunkt-Fenster trotzdem sehr eng fassen, und solche Ausdrücke als Sonderfälle behandeln, und ihre Interpretation mit einem anderen Verfahren berechnen.

Im allgemeinen kann die Effizienz sicherlich weiter verbessert werden, wenn man auch andere Phänomene einer Sonderbehandlung unterzieht. Dies bietet sich zum Beispiel für Ausdrücke mit `min_between` an. Bei diesen wird (im Moment) zunächst der entsprechende `between`-Ausdruck interpretiert. Da diese Ausdrücke zwei Argumente haben, hat solch eine Berechnung eine quadratische Komplexität. Durch die Minimierungsfunktion μ werden danach ein Großteil der generierten Zeitpunkt-Intervalle wieder gestrichen. Eine direkte Interpretation eines `min_between`-Ausdrucks, bei der jedem Punkt aus dem ersten Argument nur der unmittelbar nächste Punkt aus dem zweiten zugeordnet wird, wäre sicherlich effizienter.

Eine grundlegendere Änderung als die bisher angesprochenen besteht in einer neuen Darstellung von Mengen (vergleiche Abschnitt 8.3.1). Anstatt eine Mege von Intervallen als die Liste dieser Intervalle darzustellen, könnte man einen Datentyp kreieren, der diese Liste durch die sie generierende Bedingung kodiert. Eine Schwierigkeit bei diesem Ansatz besteht jedoch darin, etwa bei Ausdrücken der Kategorie `ORDINAL` die Abzählung eines Ausdrucks *innerhalb* eines anderen (mit vertretbarem Aufwand) zu realisieren.

8.6 Zusammenfassung

In diesem Kapitel wurde eine erste Implemtierung der in dieser Arbeit definierten Funktionen dokumentiert. Mit Hilfe des Prädikats `category/2` kann die syntaktische Korrektheit eines Ausdrucks der TEMPORAL EXPRESSION LANGUAGE überprüft beziehungsweise seine Kategorie gefunden werden. Das entsprechende Prädikat für den Kern von TEL ist `ccategory/2`. Zur Reduktion eines TEL-Ausdrucks auf einen Ausdruck der Kernsprache wird `reduce/2` verwendet. Vollständig reduzierte Ausdrücke können durch `interpretation/2` semantisch interpretiert werden. Die den auf den semantischen Bereichen erklärten Funktionen (inklusive Inferenzoperationen) entsprechenden Prädikate wurden in Abschnitt 8.3 beschrieben. Um TEL-Ausdrücke aus semantischen Objekten zurückzugewinnen, wird `sem2syn/2` benutzt. Schließlich wurden diverse Möglichkeiten aufgezeigt, wie das bestehende System verbessert werden kann, um dann in VERBMOBIL integriert werden zu können.

Kapitel 9

Schlußbemerkungen

Vorbei! ein dummes Wort.
Johann Wolfgang Goethe: Faust.

Im folgenden werden die wichtigsten Ergebnisse noch einmal zusammengefaßt und diskutiert. Außerdem wird ein Ausblick auf mögliche Weiterentwicklungen der Arbeit gegeben.

9.1 Ergebnisse

Auf Grundlage der in Kapitel 2 vorgenommenen Klassifikation zeitlicher Ausdrücke bei Terminvereinbarungsdialogen konnte die Repräsentationssprache ZeitGram zur Temporal Expression Language weiterentwickelt werden. Die Grammatiken beider Sprachen sind in Anhang B wiedergegeben. TEL schließt eine Reihe der für ZeitGram identifizierten Abdeckungslücken. Darüber hinaus wurde die Sprache neu strukturiert; es wird jetzt deutlicher als zuvor zwischen Ausdrücken, die sich auf zeitliche Dauerangaben, und solchen, die sich auf die Lage von Zeitpunkten oder -räumen beziehen, unterschieden. Innerhalb der Subsprache $L(\texttt{DATE})$, die letztendlich der Beschreibung von Terminen dient, werden die verschiedenen Merkmale eines Termins durch die Subsprachen $L(\texttt{POINT})$ und $L(\texttt{DURATION})$ spezifiziert.

Die Temporal Expression Language stellt eine Abstaktion zeitlicher Ausdrücke in deutscher und englischer Sprache dar. Es kann davon ausgegangen werden, daß auch Ausdrücke in anderen natürlichen Sprachen – abgesehen von wenigen Spezialphänomenen — in dieser formalen Sprache repräsentiert werden können. Trotz dieses Abstraktionsgrades reflektieren TEL-Ausdrücke die Struktur der ihnen entsprechenden natürlichsprachlichen Äußerungen, ein Umstand, der bei der Übersetzung sicherlich von Vorteil ist. Einige der mit dieser Sprachnähe einhergehenden Redundanzen können durch das System der in Kapitel 5 behandelten Transformationsregeln eliminiert werden. Diese Reduktion bildet die Temporal Expression Language auf ihren Kern ab, dessen Grammatik ebenfalls in Anhang B zu finden ist.

Wichtigstes Ziel dieser Arbeit war es, eine umfassende formale Semantik für TEL-Ausdrücke zu definieren. Für die drei Subsprachen L(`DURATION`), L(`POINT`) und L(`DATE`) wurden in Kapitel 6 je ein semantischer Bereich und eine Interpretationsfunktion zur Abbildung der Ausdrücke auf diese Bereiche angegeben. Mit Festlegung einer Semantik konnten in Kapitel 7 diverse Inferenzen über den semantischen Bereichen, das heißt letztlich auch über den zeitlichen Ausdrücken selbst, erklärt werden. Das gesamte System, von den syntaktischen Aspekten, über die Reduktion, bis hin zur semantischen Interpretation und Deutung dieser Interpretation durch die Auswertung von Inferenzoperationen wurde in PROLOG implemntiert. Diese Implementierung wurde in Kapitel 8 dokumentiert und diskutiert.

9.2 Ausblick

Vor der Weiterarbeit an den in dieser Arbeit behandelten Themen im engeren Sinne muß vor allem dreierlei geleistet werden, und zwar:

(i) die Implementierung eines Moduls zur Extraktion des temporalen Gehalts aus einem Redebeitrag (der in VERBMOBIL als VIT – als *Verbmobil Interface Term*, siehe [Quantz *et al.*, 1997] – vorliegt) und zur Generierung entsprechender TEL-Ausdrücke,

(ii) der Aufbau eines Systems zur Auflösung temporaler Anaphern, mit dessen Hilfe Teilausdrücke der Kategorien `ANA_POINT` und `ANA_DUR` aus TEL-Ausdrücken *vor* ihrer semantischen Interpretation eliminiert werden können, sowie

(iii) eine Evaluierung, welche Inferenzoperationen genau zur Auflösung semantischer Ambiguitäten nützlich sein können.

Erst dann ist eine Einbindung der TEMPORAL EXPRESSION LANGUAGE in das Gesamtsystem VERBMOBIL möglich und sinnvoll.

Die Zunahme an Ausdruckskraft von TEL im Vergleich zu ZEITGRAM hat auch zur Folge, daß die Generierung von Ausdrücken in dieser Sprache anspruchsvoller wird. Hierzu sind allerdings zwei wichtige Punkte anzumerken. Erstens handelt es sich um einen graduellen und nicht um einen prinzipiellen Anstieg der Komplexität der Generierung. Der entscheidende Unterschied besteht offensichtlich in den bei der TEMPORAL EXPRESSION LANGUAGE hinzugekommenen Labeln zur Kennzeichnung von Termin-Merkmalen. Das damit einhergehende Problem der Erkennung, ob sich zum Beispiel die Eingrenzung eines Zeitpunktes auf einen Terminanfang oder aber auf den gesamten Verlauf eines Termins bezieht, besteht – allerdings in abgeschwächter Form – auch in ZEITGRAM. Es manifestiert sich dort in der Wahl eines geeigneten dritten Arguments bei Ausdrücken der Kategorie `BOUNDARIES`. Dabei steht `point_between` für die Eingrenzung eines Zeitpunktes durch ein Intervall und `from_to` für den direkten Bezug auf das bezeichnete Intervall (vergleiche [Küssner und Stede, 1995] sowie die eingangs Kapitel 2 gemachten Bemerkungen über zeitliche Ausdrücke).

Die zweite wichtige Anmerkung, die in Bezug auf die Generierung von TEL-Ausdrücken zu machen ist, besteht darin, daß durch eine geeignete Einschränkung der Grammatik der TEMPORAL EXPRESSION LANGUAGE die Sprache ZEITGRAM *simuliert* werden kann. In Kapitel 2 wurde bereits analysiert, auf welche Merkmale eines Termins sich ZEITGRAM-Ausdrücke beziehen. Die Simulation besteht also in einer dem Ergebnis dieser Analyse entsprechenden festen Zuordnung von Labeln zu Ausdrücken aus L(`POINT`) beziehungsweise L(`DURATION`). Ausdrücke der ZEITGRAM-Kategorie `BOUNDARIES` mit Argument `from_to` werden also TEL-Ausdrücke der Kategorie `LIMITS` mit Label `interval` zugeordnet und solchen mit Argument `point_between` `LIMITS`-Ausdrücke mit Label `from`. In ZEITGRAM drückt `between` die Ungewissheit, ob `from_to` oder `point_between` gemeint ist, aus; in TEL gibt es zu dem gleichen Zwecke Label-Listen (beziehungsweise den `one_out_of`-Operator). Alle übrigen Zeitpunktangaben werden für die Simulation mit `from` gelabelt. Ausdrücke der Kategorie `DURATION` werden ohnehin stets durch `for` eingeleitet. Eine noch weiter vereinfachende Simulation könnte sich ganz auf die Label `from` und `for` beschränken. Das würde die Generierung deutlich erleichtern, auf der anderen Seite aber natürlich die Ausdrucksstärke der Sprache stark mindern.

Der wohlbegründete Entschluß, bei der semantischen Interpretation von Ausdrücken der Kategorie `DATE` nicht alle möglichen Termine explizit zu generieren, sondern sie stattdessen indirekt durch Quadrupel aus Angaben zu Terminbeginn, -ende, -verlauf und -dauer darzustellen, hat in Kapitel 7 bei der Angabe einer Inkonsistenzbedingung sowie bei der Definition einer semantischen Äquivalenzrelation zu Schwierigkeiten geführt. Inwieweit dieses Problem auch bei einer effizienteren Implementierung als der aktuellen von Bedeutung sein wird, ist momentan noch unklar. Trotzdem haben die in Kapitel 2 bei der Kritik an ZEITGRAM angeführten Argumente für eine Unterscheidung der diversen Merkmale eines Termins natürlich nach wie vor Gültigkeit.

Ein möglicher Kompromiß besteht darin, Dauer- und Zeitpunktangaben wie bisher durch die jeweiligen TEL-Subsprachen darzustellen, deren Zuordnung zu bestimmten Merkmalen eines Termins jedoch nicht mehr prinzipiell und mittels einer Repräsentationssprache anzugeben, sondern die nötigen Inferenzen nur bei Bedarf gezielt zu berechnen. Eine Reihe von Zeitpunktangaben würde dann also zum Beispiel zunächst einmal als eine Reihe von Ausdrücken der Kategorie `POINT` gespeichert. Nur für diejenigen Ausdrücke, für die die Zuordnung zu einem Termin-Merkmal (entsprechend den Labeln in L(`DATE`)) unmittelbar ergründet werden kann, wird auch diese Information gespeichert. Bereitet nun die Einordnung eines neu hinzukommenden Ausdrucks Schwierigkeiten, so kann er mit ausgewählten Vertretern der gespeicherten Ausdrücke verglichen werden. Nur wenn für die benötigte Information das 'Labeling' überhaupt eine Rolle spielt, muß es bei der Auswertung der jeweiligen Inferenzoperationen miteinbezogen werden. Vor dem Entwurf eines deratigen Systems bedarf es jedoch, wie bereits angemerkt, zunächst einmal einer Klärung, welche Arten von Inferenzen in welchen Situationen erforderlich sind.

Eine entsprechende Weiterentwicklung des vorgestellten Prototyps zu einem effizienten, in das VERBMOBIL-System integrierbaren Modul auf dem im letzten

Kapitel bereits angedeuteten Weg scheint zwar implementierungstechnisch anspruchsvoll, stellt aber wohl keine unüberwindliche Hürde dar.

Obschon die bearbeitete Problemstellung sich aus dem VERBMOBIL-Projekt ergeben hat, haben die Ergebnisse auch darüber hinaus Gültigkeit. Insbesondere durch den modularen Aufbau der TEMPORAL EXPRESSION LANGUAGE ergibt sich die Möglichkeit, einzelne Komponenten, wie etwa die zur Verarbeitung von Zeitpunkten, in andere Systeme zu integrieren.

Anhang A

Beispiele aus dem Verbmobil-Korpus

Für das VERBMOBIL-Projekt sind zahlreiche Transliterationen von Dialogen gesammelt worden. An diesen Daten hat sich die Entwicklung der Sprachen ZEITGRAM und TEL ausgerichtet. Im folgenden ist eine Reihe von Äußerungen mit zeitliche Ausdrücken aus diesem Korpus wiedergegeben. Die Quelle einer jeden Äußerung steht in Klammern[1]; die erste Stelle bezeichnet den Dialog, die zweite den jeweiligen Redebeitrag innerhalb des Dialogs. So wird etwa durch (G107A, NAR006) auf Redebeitrag NAR006 aus Dialog G107A verwiesen. Um die Daten nicht zweimal abdrucken zu müssen, ist der entsprechende TEL-Ausdruck (der Kategorie `DATE`, siehe Kapitel 4) jeweils gleich mitangegeben.

A.1 Deutsch: Teil I

In [Heine und Worm, 1995] sind typische Beispiele für semantische Phänomene aus dem VERBMOBIL-Korpus zusammengetragen worden, unter anderem auch Beispiele für zeitliche Ausdrücke. Diese Liste ist hier vollständig wiedergegeben.

A.1.1 Uhrzeitangaben

(1) "Ja, ich schlage so vierzehn Uhr vor." (mts2_mwl1_tsponti11.trans, mts2_1_10)
`from:fuzzy([tod:2:0,pod:pm])`

(2) "Sind Sie so ungefähr um fünfzehn Uhr dreißig bei mir." (G106A, UTB020)
`from:fuzzy([tod:3:30,pod:pm])`

[1] Die meisten Dialog-Daten liegen auf CD-ROM vor. Einige der Beispiele entstammen [Bade *et al.*, 1994].

(3) "Wann würde es Ihnen denn am liebsten passen, um neunzehn oder um zwanzig Uhr?" (G107A, NAR006)
`from:set([[tod:7:0,pod:pm],[tod:8:0,pod:pm]])`

(4) "Da kann ich erst ab achtzehn Uhr." (G192A, SAR017)
`from:in_after([tod:6:0,pod:pm])`

(5) "Von dreizehn bis sechzehn oder siebzehn Uhr, ja?" (G297A, THIP007)
`interval:min_between([tod:1:0,pod:pm],`
`set([[tod:4:0,pod:pm],[tod:5:0,pod:pm]]))`

(6) "Kommen Sie doch so vielleicht gegen halb zehn zu uns zum Frühstück." (G106A, UTB003)
`from:fuzzy(tod:9:30)`

(7) "Ja, und auch nachmittags würde es erst ab viertel vor drei gehen." (N038K, MJK001)
`from:[pod:afternoon,in_after(tod:2:45)]`

(8) "Ab acht Uhr, wenn es denn sein muß, können wir das gerne machen." (G147A, SOK004)
`from:in_after(tod:8:0)`

(9) "Dann würd' ich sagen um zehn Uhr fünfzehn." (mwj1_mps1_tsponti1.trans, mps1_1_16)
`from:tod:10:15`

(10) "Treffen wir uns um die Mittagszeit?" (mts2_mwl1_tsponti1.trans, mwl1_1_09)
`from:fuzzy(pod:midday)`

A.1.2 Datumsangaben

(11) "Also, ich dachte noch in der nächsten Woche, auf jeden Fall noch im April." (fbs1_mps1_tsponti1.trans, fbs1_1_02)
`during:[next(week),month:apr]`

(12) "In der zweiten Oktober-Woche geht's bei mir nur Montag und Dienstag, denn ab Mittwoch bin ich auf einer Konferenz in Berlin." (fsp2_mps1_tsponti1.trans, mps1_1_03)
`during:[only(set([dow:mon,dow:tue]),of(2,week,month:oct))),`
`not(in_after(dow:wed))]`

(13) "Tut mir leid, da hab' ich schon einen Termin in Lübeck, von Montag bis Dienstag." (M118D, ALE005)
`interval:min_between(dow:mon,dow:tue)`

(14) "Vielleicht könnten wir uns dann aber im Mai treffen, und zwar von fünften bis neunten." (M123D, KLA003)
`interval:[month:may,min_between(dom:5,dom:9)]`

(15) "Tut mir leid, in der Woche vom siebenundzwanzigsten bis ersten Februar bin ich nicht da." (N025K, MAM001)
`during:not(week_between(dom:27,[dom:1,month:feb]))`

(16) "Montags nachmittags irgendwann am frühen Vormittag." (fmw1_mps1_tsponti1.trans, fmw1_1_10)
`from:[dow:mon,pod:afternoon,early(pod:morning_ger2)]`
(inkonsistenter Ausdruck)

(17) "Auf jeden Fall werde ich am Donnerstag kommen." (mms4_moc1_tsponti1.trans, moc1_1_1_16)
`from:dow:thu`

(18) "Da müßt ich nur noch meine Verwandten fragen, ob's denen was ausmacht, daß ich am Totensonntag 'n Geschäftsessen halte." (M116D, JAN007)
`during:holiday:last_sunday_before_advent`

(19) "Vom fünfundzwanzigsten Februar bis zweiten März." (M118D, MON010)
`interval:min_between([dom:25,month:feb],[dom:2,month:mar])`

A.1.3 Gemischte Uhrzeit- und Datumsangaben

(20) "Also gut, treffen wir uns am Montag den vierzehnten Juni um zehn Uhr dreißig zu unserem Termin." (mhw3_mps1_tsponti1.trans, mps1_1_01)
`from:[dow:mon,dom:14,month:jun,tod:10:30]`

(21) "Am Donnerstag Vormittag so um neun wär' mir recht." (fbs1_mps1_tsponti1.trans, fbs1_1_05)
`from:[dow:thu,pod:morning_ger2,fuzzy(tod:9:0)]`

(22) "Wie wär's denn mit Donnerstag den siebzehnten Dezember um acht Uhr dreißig?" (mwj1_mps1_tsponti1.trans, mwj1_1_03)
`from:[dow:thu,dom:17,month:dec,tod:8:30]`

(23) "Am Mittwoch den siebenundzwanzigsten Oktober um dreizehn Uhr ruf' ich Sie dann an." (M116D, STI042)
`from:[dow:wed,dom:27,month:oct,tod:1:0,pod:pm]`

(24) "Dann treffen wir uns am besten gleich um dreizehn Uhr am Freitag." (G191A, KAE024)
`from:[tod:1:0,pod:pm,dow:fri]`

(25) "Morgen hätt' ich von acht bis fünfzehn Uhr Zeit." (G207A, BLA011)
`during:[tomorrow,min_between(tod:8:0,[tod:3:0,pod:pm])]`

(26) "Am Dienstag nachmittag hab' ich von vierzehn bis sechzehn Uhr keine Zeit." (G201A, BLA004)
`during:not([dow:tue,pod:afternoon,`
`min_between([tod:2:0,pod:pm],[tod:4:0,pod:pm])])`

(27) “Freitag ist prinzipiell gar nicht schlecht und zwar nachmittags um dreiviertel drei.” (fsp2_mps1_tsponti1.trans, mps_1_1_07)
`from:[dow:fri,pod:afternoon,tod:2:45]`

A.2 Deutsch: Teil II

(28) “Vielleicht müssen wir es doch früher machen?” (N035K, MGS005)
`from:ex_before(ana_point)`

(29) “Vielleicht sollten wir zum August übergehen?” (M119D, KAN032)
`during:month:aug`

(30) “Ich glaube, anderthalb Stunden müßten reichen.” (mhk1_mjk2_tsponti1.trans, mjk2_1_17)
`for:dur(1:1:2,hours)`

(31) “Ich denke, wir sollten das Ganze dann doch auf die nächste Woche verschieben.” (fsp2_mps1_tsponti1.trans, mps1_1_05)
`during:next(week)`

(32) “Meinen Sie Donnerstag den achten oder Donnerstag den fünfzehnten Juli?” (mhm1_mps1_tsponti1.trans, mps1_1_09)
`from:set([[dow:thu,dom:8],[dow:thu,dom:15,month:jul]])`

(33) “Vielleicht könnten wir direkt den ersten Mittwoch nehmen.” (M022N, VEB020)
`from:of(1,dow:wed,ana_point)`

(34) “Also heute ist der zwölfte. Ich hätte Zeit morgen ab drei Uhr.” (Z006D, AIM001)
`statement:[today,dom:12]`
`from:[tomorrow,in_after(tod:3:0)]`

(35) “Donnerstag, Freitag ginge in der zwölften Woche.” (N140K, HH1004)
`during:[set([dow:thu,dow:fri]),woy:12]`

(36) “Mein Vorschlag wäre im November, vom zweiundzwanzigsten, Montag, bis Freitag, den sechsundzwanzigsten elften.” (M111D, GAT001)
`interval:[month:nov,`
`min_between([dom:22,dow:mon],[dow:fri,dom:26,month:nov])]`

(37) “Würde sagen, machen wir es gleich im Januar, am sechzehnten, sonntags früh.” (G256A, WEM001)
`from:[month:jan,dom:16,early(dow:sun)]`

(38) “Ja, ich muß sagen, die Woche vor Ostern wäre sehr knapp.” (G075A, HAH007)
`during:not(the_before(week,holiday:easter))`

(39) "Mir würde es am besten passen in der ersten Dezemberhälfte bis zum sechzehnten einschließlich." (G081A, THS001)
`during:[firsthalf(month:dec),in_before(dom:16)]`

(40) "Aber wir könnten zum Beispiel Donnerstag den dreißigsten bis Montag den dritten nehmen." (J462A, THW005)
`interval:min_between([dow:thu,dom:30],[dow:mon,dom:3])`

(41) "Ich möchte mich gerne noch im ersten Quartal mit Ihnen in Hamburg mal für fünf Tage treffen." (J501A, NIN000)
`[during:qoy:1, for:dur(5,days)]`

(42) "Ich sehe gerade, am zweiten Juno ist ein Feiertag." (Gespräch 21, WIN016)
`statement:[dom:2,month:jun,holiday]`

(43) "Da würd' ich doch lieber vorschlagen, daß wir uns auf Ende April einigen." (G145A, BAC010)
`during:end(month:apr)`

(44) "Also, die erste Novemberwoche ist bei mir ausgebucht, da hab' ich schon eine Vorstandssitzung. Ich könnte am elften in der nächsten Woche." (L009N, SIH017)
`[during:not(of(1,week,month:nov)), from:[dom:11,`
`one_out_of([next(week),the_after(week,ana_point)])]]`
(Anapherresolution: `ana_point` $\longmapsto$ `of(1,week,month:nov)`)

(45) "Ja, drei Minuten nach der vollen Stunde fährt immer ein Zug." (G014AC, ABE013)
`from:moh:3`

(46) "Vielleicht sollten wir das gleich noch in der letzten Juliwoche machen." (G015AC, ABE007)
`during:last_of(week,month:jul)`

(47) "Soll ich Ihnen da von donnerstags auf Freitag ein Zimmer mitbuchen?" (G015AC, ABE024)
`interval:min_between(dow:thu,dow:fri)`

(48) "Also, vier Stunden eher ist dann um sieben." (G015AC, ABO038)
`statement:[neg_shift(dur(4,hours),ana_point),tod:7:0]`

(49) "Ich denke, wir sollten das schon innerhalb der nächsten drei Monate angehen." (G024AC, AER003)
`during:next(int:dur(3,months))`

(50) "Ja, das Wochenende um den zweiundzwanzigsten April herum." (G445A, THC005)
`from:the_around(pow:weekend,[dom:22,month:apr])`

(51) "Okay, laß uns mal lieber nach 'nem anderen Tag gucken." (N126K, WH1005)
`from:other(day)`

(52) "Dann kommen wir jetzt zum dritten Termin für ein fünftägiges Arbeitstreffen in der Filiale in Karlsruhe." (M254D, ADO002)
`for:dur(5,days)`

(53) "Ich habe nur Zeit vom sechsten Mai bis elften Mai, vom achtundzwanzigsten Mai bis fünften Juni und vom dreizehnten Juni bis zwanzigsten Juni." (J472A, INL000)
`during:set([min_between([dom:6,month:may],[dom:11,month:may]), min_between([dom:28,month:may],[dom:5,month:jun]), min_between([dom:13,month:jun],[dom:20,month:jun])])`

(54) "Ich habe höchstens eine Möglichkeit vom einundreißigsten Januar in den Februar hinein." (J461A, CLK001)
`interval:min_between([dom:31,month:jan],month:feb)`

(55) "Ich finde, wir sollten die letzten vier Termine vor die Jahrestagung in Wien legen." (M004N, FRB005)
`during:ex_before(ana_point)`
(Anapherresolution: `ana_point` $\longmapsto \pi$, wobei $\pi \in L(\texttt{POINT})$ die Lage der "Jahrestagung" beschreibe)

(56) "Ich denke, bis maximal dreieinhalb Stunden könnten wir dafür brauchen." (N110K, MJ5006)
`for:at_most(dur(3:1:2,hours))`

(57) "Wie sieht's bei Ihnen nach Ostern aus, fünfter bis zehnter April?" (G205A, BLA005)
`during:[ex_after(holiday:easter), min_between(dom:5,[dom:10,month:apr])]`

(58) "Ich würde vorschlagen, machen wir das zum ersten zweitägigen Termin, Montag und Dienstag, den ersten und zweiten November dreiundneunzig." (G121A, OLV010)
`[for:dur(2,days), during:[set([dow:mon,dow:tue]), set([dom:1,dom:2]), month:nov, year:1993]]`

(59) "Das paßt mir sehr gut. Mittwoch, den zehnten, Donnerstag, den elften Januar neunzehnhundert sechsundneunzig." (M137N, JKO012)
`during:[set([[dow:wed,dom:10],[dow:thu,dom:11]]), month:jan, year:1996]`

(60) "Geht das vielleicht ein, zwei Tage früher?" (M035N, MIM009)
`from:neg_shift(set([dur(1,days),dur(2,days)]),ana_point)`

(61) "In der Woche vorher, geht das?" (M035N, OLS014)
`during:the_before(week,ana_point)`

(62) "Ja, das ist ja wunderbar. Das sind zwar Feiertage, aber das ist machbar." (J511A, ULP017)
`during:[ana_point,holiday]`

(63) "Gut, Rosenmontag in Hamburg." (J511A, JMP018)
`from:holiday:monday_before_lent`

(64) "Im März könnt' ich die zweite und die dritte Woche." (N180K, SH1008)
`during:set([of(2,week,month:mar),of(3,week,month:mar)])`

A.3 Englisch

(65) "Perhaps in the next two weeks." (moko1, JBT000)
`during:next(int:dur(2,weeks))`

(66) "Mr. Helwig, I would like to get in touch with you some time later this month to talk to you for about two hours." (moko3, BFH000)
`[during:[ex_after(now),this(month)], for:fuzzy(dur(2,hours))]`

(67) "I have less than half an hour left at the end of the day, before five o'clock." (moko1, JBT002)
`[for:less(dur(1:2,hours)),`
`during:[end(day),ex_before(tod:5:0)]]`

(68) "I'm busy from two to four thirty on that afternoon. Could you do something after that?" (moko2, ISN007)
`[during:not([min_between(tod:2:0,tod:4:30),`
`that(pod:afternoon)], from:ex_after(ana_point)])`
(Anapherresolution: `ana_point` $\longmapsto$ `[min_between(tod:2:0,tod:4:30),`
`that(pod:afternoon)]`)

(69) "I could meet you after ten am and I just have a class at two. So we could pick a time in there." (R137C, NBS002)
`[from:ex_after([tod:10:0,pod:am]), during:not(tod:2:0),`
`during:min_between(ana_point,ana_point)]`
(Anapherresolution: `ana_point` $\longmapsto$ `[tod:10:0,pod:am]` beziehungsweise `ana_point` $\longmapsto$ `tod:2:0`)

(70) "How 'bout the afternoon of Monday the ninth?" (R286C, VMR000)
`during:[pod:afternoon,dow:mon,dom:9]`

(71) "I could do it on a weekend, from Friday the sixth of May to Saturday the seventh of May." (Gespräch 31, BIL005)
`interval:[pow:weekend, min_between([dow:fri,dom:6,month:may],`
`[dow:sat,dom:7,month:may])]`

(72) "What about twenty ninth through to the second of April?" (R409C, PSL002)
`interval:min_between(dom:29,[dom:2,month:apr])`

(73) "Today is Friday, January twenty second nineteen ninety three." (R409C, PSL007)
`statement:[today,[dow:fri,month:jan,dom:22,year:1993]]`

(74) "We need to get another meeting going for about two hours in the next few weeks." (R340C, CFF000)
`[for:fuzzy_dur(dur(2,hours)),`
`during:next(int:dur(several,weeks))]`

(75) "Are you free on Monday the twenty sixth after twelve o'clock or how 'bout sometime in the morning on Tuesday the twenty seventh?" (R340C, CFF000)
`from:set([[dow:mon,dom:26,ex_after(tod:12:0)],`
`[pod:morning,dow:tue,dom:27]])`

(76) "However on the twenty eighth, which is Wednesday, I've got one to four pm free." (R340C, SXL001)
`during:[dom:28,dow:wed,`
`min_between([tod:1:0,pod:pm],[tod:4:0,pod:pm])]`

(77) "Would you like to meet me the next three weeks?" (R001K, CH1002)
`during:next(int:dur(3,weeks))`

(78) "In the morning at ten." (R001K, FJ2009)
`from:[pod:morning,tod:10:0]`

(79) "Okay, how 'bout Tuesday March the sixteenth sometime after twelve o'clock pm?" (R206C, EPP002)
`from:[dow:tue,month:mar,dom:16,ex_after([tod:0:0,pod:pm])]`

(80) "Let's say from between twenty two to twenty sixth or on the thirty first that would be fine with me." (R409C, NFH001)
`from:set([min_between(dom:22,dom:26),dom:31])`

Anhang B

Grammatiken

B.1 ZeitGram

Im folgenden wird die in [Küssner und Stede, 1995] vorgestellte Grammatik von ZeitGram, einer Sprache zur Beschreibung zeitlicher Ausdrücke, wiedergegeben. Die grundlegende Form eines ZeitGram-Ausdrucks ist eine Liste von Objekten der Kategorie `DATE_EXPR`, der mit Hilfe des Operators `tempex/2` ein Name zugeordnet werden kann.

B.1.1 Top-Level

```
TEMPEX        ::=  tempex(NAME,LIST_OF_DATE_EXPRS)
LIST_OF_DATE_EXPRS  ::=  DATE_EXPR+
DATE_EXPR     ::=  TEMPOBJ | QUANTIFIER(DATE_EXPR*,DATE_EXPR*) |
                   ao(LIST_OF_DATE_EXPRS+) | FUZZYFIER(TEMPOBJ+)
QUANTIFIER    ::=  all | any | not | only
FUZZYFIER     ::=  about
TEMPOBJ       ::=  POINT | INTERVAL | COMPOSED_POINTLIKE |
                   MOD(MODIFIABLE_POINTLIKE) |
                   MODIFIABLE_POINTLIKE | MOD
MODIFIABLE_POINTLIKE  ::=  UNIT | POINTLIKE | COUNT_POINTLIKE
MOD           ::=  early | late | begin | middle | end |
                   firsthalf | secondhalf
```

B.1.2 Einfache Ausdrücke

```
POINT         ::=  tod:TOD
TOD           ::=  00:00 | ... | 12:59
```

```
POINTLIKE    ::=  pod:POD | dow:DOW | poy:POY | woy:WOY |
                  wom:WOM | dom:DOM | moy:MOY | year:YEAR |
                  holiday:HOLIDAY | xoy:NUMBER | ana

POD          ::=  morning | beforenoon | noon | afternoon |
                  evening | night | daytime | am | pm

DOW          ::=  mon | tue | wed | thu | fri | sat | sun |
                  today | tomorrow | totomorrow | yesterday |
                  yeyesterday | workday | weekend

POY          ::=  spring | summer | fall | winter

WOY          ::=  1 | 2 | ... | 52

WOM          ::=  1 | 2 | 3 | 4 | 5

DOM          ::=  1 | 2 | ... | 31

MOY          ::=  jan | feb | mar | apr | may | jun | jul | aug |
                  sep | oct | nov | dec

YEAR         ::=  1900 | ... | 1999

HOLIDAY      ::=  neujahr | hlg-drei-koenige | ...
```

B.1.3 Intervalle

```
INTERVAL     ::=  DURATION | LIMIT | BOUNDARIES

DURATION     ::=  dur(NUMBER,UNIT)

UNIT         ::=  year  | month | week | day | hour | minute

LIMIT        ::=  BEFORE_OR_AFTER(TEMPOBJ+)

BEFORE_OR_AFTER  ::=  before | after | ex_before | ex_after |
                  in_before | in_after

BOUNDARIES   ::=  boundaries(TEMPOBJ+,TEMPOBJ+,
                  <from_to|between|point_between>)
```

B.1.4 Ausdrücke mit Referenzzeitpunkt

```
COMPOSED_POINTLIKE  ::=  in(DURATION) | in(DOW,DURATION) |
                  <before|after>(COUNTER,<UNIT|DOW>,POINTLIKE) |
                  POINT_MOD(week,POINTLIKE)

POINT_MOD    ::=  before | after | around
```

B.1.5 Ausdrücke mit Abzählung

```
COUNT_POINTLIKE  ::=  counted(COUNTER,UNIT) |
                      DEIC_COUNTER(<DOW|MOY>) |
                      counted(COUNTER,<week|DOW>,COUNT_UNIT_OR_MOY)

COUNT_UNIT_OR_MOY ::= DEIC_COUNTER(<year|month>) | MOY

COUNTER       ::=  NUMBER | DEIC_COUNTER

DEIC_COUNTER  ::=  this | last | lastlast | lastlastast | next |
                   nextnext

NUMBER        ::=  1 | 2 | ...
```

B.2 TEMPORAL EXPRESSION LANGUAGE

An dieser Stelle wird die in Kapitel 4 definierte Grammatik der Sprache TEL (TEMPORAL EXPRESSION LANGUAGE) nochmals 'am Stück' gelistet. Sie besteht aus drei einzelnen Sprachen, je einer für Zeitdauern (`DURATION`), Zeitpunkte (`POINT`) und Termine (`DATE`). Durch den Operator `tempex/2` wird einem Ausdruck der Kategorie `DATE` eine Bezeichner zugewiesen.

B.2.1 Beschreibung von Zahlen

```
INTEGER       ::=  0 | 1 | -1 | 2 | -2 | 3 | -3 | ...

NUMBER        ::=  1 | 2 | 3 | ...

NUMBER0       ::=  0 | NUMBER

FRACTION      ::=  NUMBER | NUMBER:NUMBER | NUMBER:NUMBER:NUMBER
```

B.2.2 Beschreibung einer Dauer (DURATION)

```
DURATION      ::=  set(DURATION+) | one_out_of(DURATION+) |
                   BASIC_DUR | RANGE | OPEN_RANGE | FUZZY_DUR |
                   ANA_DUR

BASIC_DUR     ::=  dur(<FRACTION|several>,UNIT)

UNIT          ::=  years | months | weeks | days | hours | minutes

RANGE         ::=  range(DURATION,DURATION)

OPEN_RANGE    ::=  <at_least|at_most|more|less>(DURATION)

FUZZY_DUR     ::=  fuzzy_dur(DURATION)

ANA_DUR       ::=  ana_dur
```

B.2.3 Beschreibung eines Zeitpunktes (POINT)

B.2.3.1 Top-Level

```
POINT        ::=  POINT* | set(POINT+) | one_out_of(POINT+) |
                  not(POINT) | only(POINT,POINT) | BASIC |
                  COMPLEX | MODIFIED | FUZZY | QUANTIFIED |
                  ANA_POINT

GENERIC      ::=  day | week | month | quarter_year | half_year |
                  year

MODIFIED     ::=  MOD(<POINT|GENERIC>)

MOD          ::=  early | late | begin | middle | end |
                  firsthalf | secondhalf

FUZZY        ::=  fuzzy(POINT)

QUANTIFIED   ::=  quantified(QUANTIFIER,<POINT|GENERIC>) |
                  quantified(not(QUANTIFIER),<POINT|GENERIC>) |
                  quantified(every:NUMBER,COUNTABLE1)

QUANTIFIER   ::=  NUMBER | several | every

ANA_POINT    ::=  ana_point | that(<POINT|GENERIC>) |
                  other(<POINT|GENERIC>)
```

B.2.3.2 Einfache Ausdrücke

```
BASIC        ::=  now | moh:MOH | tod:TOD | pod:POD | dow:DOW |
                  pow:POW | DEICTIC_DAY | dom:DOM | woy:WOY|
                  month:MONTH | season:SEASON | qoy:QOY |
                  hoy:HOY | year:YEAR | holiday | holiday:HOLIDAY

MOH          ::=  NUMBER0

TOD          ::=  NUMBER0:NUMBER0

POD          ::=  am | pm | morning | morning_ger1 | morning_ger2 |
                  midday | afternoon | evening | night | daytime

DOW          ::=  mon | tue | wed | thu | fri | sat | sun

POW          ::=  workday | weekend

DEICTIC_DAY  ::=  today | tomorrow | yesterday |
                  deictic_day:INTEGER

DOM          ::=  NUMBER

WOY          ::=  NUMBER

MONTH        ::=  jan | feb | mar | apr | may | jun | jul | aug |
                  sep | oct | nov | dec

SEASON       ::=  spring | summer | fall | winter
```

```
QOY          ::=  NUMBER

HOY          ::=  NUMBER

YEAR         ::=  NUMBER

HOLIDAY      ::=  advent | first_advent | second_advent |
                  third_advent | fourth_advent | all_saints_day |
                  ascension_day | ash_wednesday | assumption_day |
                  boxing_day | carnival | christmas |
                  christmas_day | christmas_eve | corpus_christi |
                  day_of_prayer_and_repentance | easter |
                  easter_monday | easter_sunday | epiphany |
                  german_unity_day | good_friday |
                  harvest_festival | holy_saturday | holy_week |
                  last_sunday_before_advent | maundy_thursday |
                  may_day | monday_before_lent | mother_s_day |
                  national_day_of_mourning | new_year_s_day |
                  new_year_s_eve | reformation_day |
                  shrove_tuesday | st_nicholas_day |
                  st_valentine_s_day | whitsun | whit_monday |
                  whit_sunday
```

B.2.3.3 Komplexe Ausdrücke

```
COMPLEX      ::=  LIMIT | LIMITS | SHIFTED | RELATED | ORDINAL |
                  DEICTIC

LIMIT        ::=  BEFORE_AFTER(POINT)

LIMITS       ::=  between(POINT,POINT) | min_between(POINT,POINT)

SHIFTED      ::=  <pos_shift|neg_shift>(DURATION,POINT) |
                  <in|ago>(DURATION)

RELATED      ::=  the_around(COUNTABLE,POINT) |
                  week_between(POINT,POINT) |
                  <the_before|the_after>(NUMBER,COUNTABLE,POINT) |
                  <the_before|the_after>(COUNTABLE,POINT) |
                  <dow_before|dow_after>(dow:DOW,DURATION,POINT) |
                  <dow_before|dow_after>(dow:DOW,DURATION)

ORDINAL      ::=  of(NUMBER,COUNTABLE,<POINT|GENERIC>) |
                  last_of(NUMBER,COUNTABLE,<POINT|GENERIC>) |
                  last_of(COUNTABLE,<POINT|GENERIC>)

DEICTIC      ::=  this(COUNTABLE) | next(COUNTABLE) |
                  last(COUNTABLE) | deictic(INTEGER,COUNTABLE)

BEFORE_AFTER ::=  before | after | in_before | in_after |
                  ex_before | ex_after

COUNTABLE    ::=  COUNTABLE1 | int:DURATION
```

```
COUNTABLE1   ::=  GENERIC | pod:POD | dow:DOW | pow:POW |
                  dom:DOM | season:SEASON | holiday |
                  holiday:HOLIDAY | ana_point
```

B.2.4 Beschreibung eines Termins (DATE)

```
TEMPEX       ::=  tempex(NAME,DATE)

DATE         ::=  DATE* | set(DATE+) | one_out_of(DATE+) |
                  not(DATE) | for:DURATION | POINTLABEL:POINT |
                  POINTLABEL+:POINT

POINTLABEL   ::=  from | to | during | interval | statement
```

B.3 Der Kern von TEL

Einige Ausdrücke in TEL werden zur Berechnung ihrer Semantik zunächst in andere Ausdrücke umgeformt (siehe Kapitel 5). Interpretationsfunktionen mußten daher auch nur für den sogenannten Kern von TEL definiert werden. Die Grammatik dieser Kernsprache wird im folgenden wiedergegeben.

B.3.1 Beschreibung von Zahlen

```
INTEGER      ::=  0 | 1 | -1 | 2 | -2 | 3 | -3 | ...

NUMBER       ::=  1 | 2 | 3 | ...

NUMBER0      ::=  0 | NUMBER

FRACTION     ::=  NUMBER | NUMBER:NUMBER | NUMBER:NUMBER:NUMBER
```

B.3.2 Beschreibung einer Dauer (DURATION)

```
DURATION     ::=  set(DURATION+) | one_out_of(DURATION+) |
                  BASIC_DUR | RANGE | OPEN_RANGE | MODIFIED_DUR |
                  ANA_DUR

BASIC_DUR    ::=  dur(FRACTION,UNIT)

UNIT         ::=  years | months | weeks | days | hours | minutes

RANGE        ::=  range(DURATION,DURATION)

OPEN_RANGE   ::=  <at_least|at_most|more|less>(DURATION)

MODIFIED_DUR ::=  modified_dur(FRACTION,FRACTION,DURATION)

ANA_DUR      ::=  ana_dur
```

B.3.3 Beschreibung eines Zeitpunktes (POINT)

B.3.3.1 Top-Level

```
POINT        ::=  POINT* | set(POINT+) | one_out_of(POINT+) |
                  not(POINT) | BASIC | COMPLEX | MODIFIED |
                  QUANTIFIED | ANA_POINT
GENERIC      ::=  day | week | month | quarter_year | half_year |
                  year
MODIFIED     ::=  modified(LAMBDA,<POINT|GENERIC>)
LAMBDA       ::=  lambda(LAMBDA_ARG,LAMBDA_ARG)
LAMBDA_ARG   ::=  left | right | plus(<left|right>,FRACTION) |
                  minus(<left|right>,FRACTION)
QUANTIFIED   ::=  quantified(QUANTIFIER,<POINT|GENERIC>) |
                  quantified(not(QUANTIFIER),<POINT|GENERIC>) |
                  quantified(every:NUMBER,COUNTABLE1)
QUANTIFIER   ::=  NUMBER | every
ANA_POINT    ::=  ana_point | that(<POINT|GENERIC>) |
                  other(<POINT|GENERIC>)
```

B.3.3.2 Einfache Ausdrücke

```
BASIC        ::=  now | moh:MOH | tod:TOD | pod:POD | year:YEAR
MOH          ::=  NUMBER0
TOD          ::=  NUMBER0:NUMBER0
POD          ::=  am | pm
YEAR         ::=  NUMBER
```

B.3.3.3 Komplexe Ausdrücke

```
COMPLEX      ::=  LIMIT | LIMITS | SHIFTED | RELATED | ORDINAL |
                  DEICTIC
LIMIT        ::=  <in_before|in_after|ex_before|ex_after>(POINT)
LIMITS       ::=  between(POINT,POINT) | min_between(POINT,POINT)
SHIFTED      ::=  <pos_shift|neg_shift>(DURATION,POINT)
RELATED      ::=  the_around(COUNTABLE,POINT) |
                  <the_before|the_after>(NUMBER,COUNTABLE,POINT)
ORDINAL      ::=  of(NUMBER,COUNTABLE,<POINT|GENERIC>) |
                  last_of(NUMBER,COUNTABLE,<POINT|GENERIC>)
COUNTABLE    ::=  COUNTABLE1 | int:DURATION
COUNTABLE1   ::=  POINT | GENERIC
```

B.3.4 Beschreibung eines Termins (DATE)

```
TEMPEX       ::=  tempex(NAME,DATE)

DATE         ::=  DATE* | set(DATE+) | one_out_of(DATE+) |
                  not(DATE) | for:DURATION | POINTLABEL:POINT |
                  POINTLABEL+:POINT

POINTLABEL   ::=  from | to | during | statement
```

Literaturverzeichnis

[Allen, 1983] James F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, November 1983.

[Allen, 1984] James F. Allen. Towards a General Theory of Action and Time. *Artificial Intelligence*, 23:123–154, 1984.

[Allen *et al.*, 1991] James F. Allen, Henry A. Kautz, Richard N. Pelavin und Josh D. Tenenberg. *Reasoning about Plans.* Morgan Kaufmann Publishers, 1991.

[Åqvist, 1978] Lennart Åqvist. A System of Chronological Tense Logic. In F. Guenthner und S. J. Schmidt (eds.), *Formal Semantics and Pragmatics for Natural Language*, 223–254, D. Reidel Publishing Company, Dordrecht, 1978.

[Baader und Hanschke, 1991] Franz Baader und Philipp Hanschke. *A Scheme for Integrating Concrete Domains into Concept Languages.* DFKI Research Report RR-91-10, Deutsches Forschungszentrum für Künstliche Intelligenz, Kaiserslautern, 1991.

[Bade *et al.*, 1994] Ute Bade, Susanne Heizmann, Susanne Jekat-Rommel, Shinichi Kameyama, Detlev Krause, Ilona Maleck, Birte Prahl und Wiebke Preuß. *Wizard-of-Oz-Experimente mit dem Verbmobil-Simulator.* Verbmobil Memo 24, Juni 1994.

[Bien, 1988] Reinhold Bien. Kalenderkunst und Kalendergeschichte. *Sterne und Weltraum*, 12:710–713, 1988.

[Brachman und Schmolze, 1985] Ronald J. Brachmann und James G. Schmolze. An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science*, 9(2):171–216, April 1985.

[Bratko, 1990] Ivan Bratko. *Prolog Programming for Artificial Intelligence.* 2nd edition, Addison-Wesley Publishing Company, 1990.

[Fischer, 1992] Martin Fischer. *The Integration of Temporal Operators into a Terminological Representation System.* KIT Report 99, Technische Universität Berlin, Dezember 1992.

[Fisher und Owens, 1993] Michael Fisher und Richard Owens. An Introduction to Executable Modal and Temporal Logics. In M. Fisher und R. Owens (eds.), *Executable Modal and Temporal Logics, IJCAI '93 Workshop Proceedings*, 1–20, August 1993.

[Fitting, 1993] Melvin Fitting. Basic Modal Logic. In Dov Gabbay, Chris Hogger und J. H. Robinson (eds.), *Handbook of Logic in Artificial Intelligence and Logic Programming (Volume I: Logical Foundations)*, 1:368–448, Clarendon Press, 1993.

[Goldblatt, 1992] Robert Goldblatt. *Logics of Time and Computation.* 2nd edition, CSLI Lecture Notes (7), 1992.

[Heine und Worm, 1995] Julia E. Heine und Karsten L. Worm. *Semantic Phenomena for German with Examples.* Verbmobil Memo 86, Universität des Saarlandes, Computerlinguistik, August 1995.

[Hemme, 1991] Heinrich Hemme. Wann ist Ostern? *Bild der Wissenschaft*, 3:132–133, 1991.

[Hildebrandt, 1994] Bernd Hildebrandt. *Struktur und Bedeutung Temporaler Konstituenten in einem Sprachverstehenden Dialogsystem.* Dissertation, Universität Bielefeld, 1994.

[Hildebrandt *et al.*, 1993] Bernd Hildebrandt, Gernot A. Fink, Franz Kummert und Gerhard Sagerer. Modeling of Time Constituents for Speech Understanding. In *Proceedings of the European Conference on Speech Communication and Technology*, 2247–2250, Berlin, 1993.

[Knuth, 1962] Donald E. Knuth. The Calculation of Easter. *Communications of the ACM*, 5(4):209–210, April 1962.

[Küssner und Stede, 1995] Uwe Küssner und Manfred Stede. *Zeitliche Ausdrücke: Repräsentation und Inferenz.* Verbmobil Memo 100, Technische Universität Berlin, Dezember 1995.

[Lipson, 1981] John D. Lipson. *Elements of Algebra and Algebraic Computing.* Benjamin/Cummings Publishing Company, 1981.

[Mitton, 1978] Simon Mitton (ed.). *Cambridge Enzyklopädie der Astronomie.* Bertelsmann Lexikon-Verlag, 1978.

[Moore, 1989] Patrick Moore (ed.). *Lexikon der Astronomie.* Verlag Herder, Freiburg im Breisgau, 1989.

[Nebel, 1990] Bernhard Nebel. *Reasoning and Revision in Hybrid Representation Systems.* Lecture Notes in Artificial Intelligence (422), Springer-Verlag, 1990.

[Neuwirth, 1993] Andreas Neuwirth. *Inferences for Temporal Object Descriptions in a Terminological Representation System: Design and Implementation.* KIT Report 107, Technische Universität Berlin, April 1993.

[Quantz *et al.*, 1995] J. Joachim Quantz, Guido Dunker, Frank Bergmann und Ivonne Kellner. *The FLEX System.* KIT Report 124, Technische Universität Berlin, Dezember 1995.

[Quantz *et al.*, 1997] J. Joachim Quantz, Bernd Mahr, Birte Schmitz, Uwe Küssner, Manfred Stede, Guido Dunker, Frank Bergmann und Ivonne Kellner. *Semantische Auswertung für die Übersetzung – Abschlußbericht Projekt KIT-VM11.* KIT Report 139, Technische Universität Berlin, Juni 1997.

[Schild, 1991] Klaus Schild. *A Tense-Logical Extension of Terminological Logics.* KIT Report 92, Technische Universität Berlin, September 1991.

[Schmiedel, 1990] Albrecht Schmiedel. A Temporal Terminological Logic. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, 2:640–645, American Association for Artificial Intelligence, 1990.

[Schmitt, 1997] Peter H. Schmitt. *Nichtklassische Logiken.* Vorlesungsskriptum, Universität Karlsruhe, 1997.

[Tantzen, 1963] Robert G. Tantzen. Algorithm 199: Conversions between Calendar Date and Julian Day Number. *Communications of the ACM*, 6(8):444, August 1963.

[Wahlster, 1993] Wolfgang Wahlster. *Verbmobil – Translation of Face-to-Face Dialogs.* DFKI Research Report RR-93-34, Deutsches Forschungszentrum für Künstliche Intelligenz, Juni 1993.