Anfragesprachen für Textsuchsysteme

Reiner, Ulrike

Veröffentlichungsversion / Published Version
Sammelwerksbeitrag / collection article

Anfragesprachen für Textsuchsysteme

Ulrike Reiner
Technische Universität Berlin
e-mail:ul@cs.tu-berlin.de

1 Einleitung

Um sich mitzuteilen und zu verständigen, spricht und schreibt der Mensch, es entstehen Texte. So ist im Laufe der Menschheit eine gewaltige Textmenge entstanden, reich an unterschiedlichen Textarten und Notationen, wie z. B. Büchertexte, Formeltexte oder Notentexte. Im Unterschied zu früher, können Texte heute mit Hilfe des Computers noch schneller erstellt, verändert, kopiert, gespeichert, versendet und seit neuestem auch interpretiert werden [1].

Unabhängig vom jeweiligen Bearbeitungswunsch müssen die Texte - sofern sie nicht neu erstellt werden - zur Bearbeitung zunächst gesucht werden. Auch die Suche nach weit entfernt liegenden Texten oder Textteilen ist durch den Computer extrem beschleunigt worden: Anstatt z.B. wegen eines Textes zur 'Library of Congress' - Bibliothek nach Washington D.C., USA, über den Atlantik zu segeln, ist es heute unter bestimmten Voraussetzungen möglich, die Suche bequem von einem Sessel aus zu starten und sich den gefundenen Text auf dem Heimbildschirm anzeigen und bei Bedarf auch ausdrucken zu lassen.

Die bequeme und schnelle Suche hat jedoch auch ihren Preis. Zum einen müssen die technischen Voraussetzungen geschaffen und der Dienst bezahlt werden. Zum anderen müssen die Mittel zur Suche beherrscht werden. Dies sind Anfragesprachen vieler unterschiedlicher Suchsysteme, die sich meist nur syntaktisch, jedoch nicht wesentlich in ihrer semantischen Mächtigkeit unterscheiden.

Naheliegend ist daher, der Sprachenvielfalt mit einer einzigen Anfragesprache zu begegnen. Dies ist in [2] näher ausgeführt. Dort wird eine auf der Prädikatenlogik aufbauende Anfragesprache IQL (Intermediary Query Language) mit wohldefinierter Syntax und Semantik vorgestellt. Die Anfragesprache IQL spielt die Rolle einer Zwischensprache. Mit Compilergeneratoren können Anfragensprachübersetzer für häufig benutzte Retrieval- oder Datenbanksprachen wie z. B. IQL-MESSENGER oder IQL-SQL generiert werden. Für ungeübte oder geübte Benutzer werden dort zwei Benutzersprachen TUQL (Trained User Query Language)

und UUQL (Untrained User Query Language) vorgeschlagen.

Die vorliegende Arbeit knüpft an [2] an und konzentriert sich auf die Suchkomponente von Textsystemen (Kapitel 2). Textsuchsysteme werden wegen der wachsenden Menge an Online-Texten immer attraktiver und spielten im ATLAS-Projekt eine wichtige Rolle. Im ATLAS-Projekt ging es um '... die Entwicklung von Methoden und Werkzeugen, die dem menschlichen Interpreten die Auswertung und Verwaltung großer Textmengen erleichtern soll.' [1].

Das ATLAS-Archiv enthält vor allem Texte aus den sozialwissenschaftlichen Forschungsbereichen Technik, Umweltzerstörung und Stadtentwicklung. Die meisten Teilkorpora sind aus Interviewstudien hervorgegangen. In diesen Fällen liegen die Texte meistens sowohl als Tonbandaufzeichnung wie auch verschriftet vor. Texte sind z. B. Zeitungs- und Zeitschriftenartikel, Tagebücher, Briefe, transkribierte Alltagsgespräche, Interviews, Reden, Gruppendiskussionen, Rundfunk- und Fernsehbeiträge. Auf der Wunschliste der Psychologen als auch der Linguisten zu einer computergestützten Textinterpretation stehen u.a. Such- und Ordnungsfunktionen, Benutzerfreundlichkeit und Hilfe beim Auffinden von Textstellen und von Kontext (a.a.O.), wovon einige Suchmöglichkeiten von ATLAS/ti zur Verfügung gestellt werden (Kapitel 2.3).

Die Textsuche wird im wesentlichen von experimentellen und kommerziellen Information Retrieval Systemen, Datenbanksystemen und Textverarbeitungssystemen realisiert, die sich weitgehend unabhängig voneinander entwickelt haben. In dieser Arbeit werden die Suchmöglichkeiten heutiger Textsuchsysteme betrachtet, wobei die Anfragesprache IQL-DM aus [2] überarbeitet und z. B. um Suchmöglichkeiten mit regulären Ausdrücken - wie sie in Textverarbeitungssystemen Verwendung finden - erweitert wird. Das Ergebnis bildet die in Kapitel 3 vorgestellte Anfragesprache IQL-T (T für Text), die wie oben erwähnt, eine Zwischensprache bildet. Auf dieser Grundlage können die Anfragesprachen TUQL-T bzw. UUQL-T entwickelt werden, die in Zusammenarbeit mit den Benutzern erfolgen sollte. In Kapitel 3.3 wird die Textsuche anhand von Beispielen in den unterschiedlichen Anfragesprachen demonstriert.

2 Textsuchsysteme

Unter Textsuchsystemen werden hier Systeme bzw. Systemkomponenten verstanden, die eine Textsuche realisieren. Im Unterschied zu den anderen in diesem Kapitel behandelten computerunterstützten Systemen steht bei **Information Retrieval Systemen** - wie der Name schon sagt - die Suche im Vordergrund. Es kann mit Suchbedingungen wie Schlagwörtern (Deskriptoren), Thesaurusoperatoren, (maskierten) Zeichenketten oder Abstandsoperatoren nach Dokumenten

gesucht werden.

Bei **Datenbanksystemen**, insbesondere bei SQL-Datenbanksystemen, wird mit Suchbedingungen bestehend aus Attributnamen, Operatoren und Werten nach Attributwerten in einer oder mehreren Datenbanktabelle/n gesucht.

Im Projekt ATLAS [1] wurde im Rahmen der Projektaufgabe eine neue Systemklasse geschaffen und dafür der Name **Interpretations-Unterstützungs-Systeme** geprägt. Diese neuartigen Systeme dienen zur Arbeitsunterstützung der Textinterpreten. Mit dem System ATLAS/ti[1] ist eine Suche nach Kode-, Primärtext-, Memo- oder Zitatlisten, nach Textteilen und kodierten Textteilen (Zitaten) mit ev. disjunktiv verknüpften (maskierten) Zeichenketten und Schlagwörtern (Kodes) möglich.

Textverarbeitungssysteme sind z. B. Texteditoren, Text-Graphik-Editoren oder Dokumentenbearbeitungssysteme[2]. Vielen Textverarbeitungsfunktionen ist die Suche vorgeschaltet: um z. B. Zeichenketten zu verändern, löschen, abzuspeichern oder einzufügen, müssen die zu ändernden Stellen zunächst gesucht werden. UNIX-Textverarbeitungsprogramme ermöglichen eine Zeichenkettensuche mit regulären Ausdrücken. Damit sind z. T. mehr Sprachformulierungsmöglichkeiten vorhanden als in Datenbanksystemen und Information Retrieval Systemen.

Trotz gebietsspezifischer Unterschiede der unterschiedlichen Textsuchsysteme (vgl. Kapitel 2.1 - 2.4) kann die Textsuche vereinheitlicht werden: die Suche wird - wie in Kapitel 3 dargelegt - als eine Suche mit Text-, Werte-, Mustersymbolen usw. nach Texten bzw. Textteilen[3] aufgefaßt. In den Kapiteln 2.1 - 2.4 folgt eine Zusammenschau der Systemklassen mit ausgewählten Sprachen und Sprachelementen, die keinen Anspruch auf Vollständigkeit erhebt, was - durch schwer zugängliche Literatur[4] und Softwareweiterentwicklungen - ohnehin so gut wie unmöglich ist.

2.1 Information Retrieval Systeme

Information (auch Dokument) Retrieval ist ein Gebiet, das sich mit der Struktur, Analyse, Organisation, Speicherung und der Suche von Information beschäftigt [3], wobei effektive und effiziente Suchverfahren für bibliographische und textuelle Daten einen Forschungsschwerpunkt bilden. Volltextdatenbanken enthalten den vollständigen Text einer Dokumentationseinheit und sind meist zusätzlich

[1] ti für Textinterpretation

[2] z. B. zur Erstellung und Verschickung genormter Texte und Bilder in Rechnernetzen

[3] Zeilen, Kapiteln, Paragraphen, Wörtern etc.

[4] System- und Benutzerhandbücher sind selten in öffentlichen Bibliotheken verfügbar und veralten relativ schnell.

durch weitere Felder beschrieben, während bibliographische Datenbanken 'nur' Beschreibungen der Dokumentationseinheiten enthalten ([4], S. 181ff). Auf die heute ca. 5000 Online-Datenbanken kann mit unterschiedlichen Systemen und Sprachen zugegriffen werden [5]. Stellvertretend für Information Retrieval Sprachen werden hier einige wie MESSENGER, STAIRS, DIALOG, BRS/Search und die ISO-8777 Norm herausgegriffen.

MESSENGER ist eine Kommandosprache, um im Dialog von Europa, Nord Amerika und Japan direkt auf wissenschaftliche Datenbanken des STN (**S**cientific & **T**echnical Information **N**etwork) International zuzugreifen. Für die Freitextsuche stellt MESSENGER innerhalb des Suchkommandos SEARCH folgende Metazeichen[5] bereit [6]: *Rechtsmaskierung:* ? (0 oder beliebig viele beliebige Zeichen), # (0 oder ein beliebiges Zeichen), #...#$_n$ (max. n beliebige Zeichen); *Rechts- oder Innenmaskierung:* ! (ein beliebiges Zeichen), !...!$_n$ (n beliebige Zeichen); *Wortmaskierung:* (*W*) (ohne Wortabstand in angegebener Reihenfolge), (*NOTW*) (ohne Wortabstand nicht in angegebener Reihenfolge), (*nW*) (max. n Wörter Abstand in angegebener Reihenfolge), (*A*) (ohne Wortabstand in beliebiger Reihenfolge), (*nA*) (max. n Wörter Abstand in beliebiger Reihenfolge), (*L*) (im gleichen Attribut), (*S*) (im gleichen Satz), (*P*) (im gleichen Paragraphen).

Ein anderes bekanntes Retrievalsystem ist **STAIRS** der Firma IBM. Das Kommando SEARCH ermöglicht u.a. [7]: *Rechtsmaskierung:* $ (beliebig viele beliebige Zeichen), $*n* (max. n beliebige Zeichen), *Wortmaskierung:* (*ADJ*) (ohne Wortabstand in angegebener Reihenfolge), (*WITH*) (im gleichen Satz), (*SAME*) (im gleichen Paragraphen).

Mit **DIALOG**, der Sprache des ältesten Online Dienstes von Dialog Information Services [8], wird eine Textsuche innerhalb des SELECT-Kommandos mit folgenden Metazeichen durchgeführt [9]: *Rechtsmaskierung:* ? (ein beliebiges Zeichen), ? ? (beliebig viele beliebige Zeichen), ?...?$_n$ (n beliebige Zeichen); *Wortmaskierung:* (*W*) (ohne Wortabstand in angegebener Reihenfolge), (*nW*) (max. n Wörter Abstand in angegebener Reihenfolge), (*N*) (ohne Wortabstand in beliebiger Reihenfolge), (*nN*) (max. n Wörter in beliebiger Reihenfolge), (*F*) (im gleichen Attribut). Mit (NOT W), (NOT N) etc. kann die Wortmaskierung negiert werden.

BRS/Search (**B**ibliographic **R**etrieval **S**ervices) basiert auf STAIRS mit demgegenüber sowohl eingeschränkten als auch erweiterten Sprachmöglichkeiten. Für die Textsuche gibt es u. a. nachfolgende Metazeichen [10]: *Rechtsmaskierung:* $ (beliebig viele beliebige Zeichen), $*n* (max. n beliebige Zeichen), ? (ein beliebiges Zeichen); *Rechts-, Links- oder Innenmaskierung:* $ (beliebig viele beliebige

[5]Zeichen, die für andere Zeichen stehen

Zeichen); *Wortmaskierung:* (*SAME*) (im gleichen Attribut), (*ADJ*) (in angegebener Reihenfolge im gleichen Satz), (*WITH*) (in beliebiger Reihenfolge im gleichen Satz); mit $SETNEARLEVEL = n$ bzw. $SETADJLEVEL = n$ kann ein Abstand von n Wörtern angegeben werden.

Der internationale Standard **ISO 8777** spezifiziert Kommandos für eine einheitliche Dialogsprache für Information Retrieval Systeme. Eine interaktive Suche wird mit *FIND* durchgeführt, wobei für die Zeichen- und Wortmaskierung folgende Metazeichen festgelegt sind [11]: *Zeichenmaskierung* ? (beliebig viele beliebige Zeichen), ?*n* (max. n beliebige Zeichen), # (ein beliebiges Zeichen), $\#...\#_n$ (n beliebige Zeichen); *Wortmaskierung* ! (ohne Wortabstand in angegebener Reihenfolge), % (ohne Wortabstand in beliebiger Reihenfolge), !*n* (max. n Wörter Abstand in angegebener Reihenfolge), %*n* (max. n Wörter Abstand in beliebiger Reihenfolge).

Bei allen Information Retrieval Sprachen gibt es die üblichen logischen Operatoren AND, OR, NOT, allerdings mit unterschiedlichen Prioritätsregelungen und die meisten Sprachen verfügen über Thesaurusoperatoren.

2.2 Datenbanksysteme

In dieser Arbeit werden nur Suchmöglichkeiten *relationaler* Datenbanksysteme betrachtet, die Suche in Hypermedia-Datenbanken oder objektorientierten Datenbanken findet keine Berücksichtigung, zumal die Suche z. B. nach Bildern, Photos, Filmen oder gesprochener Sprache zur Zeit kaum realisiert ist. Der heute als klassisch geltende Artikel von Codd [12] hat die Forschung an den Universitäten und die Industrie soweit inspiriert, daß relationale Datenbanksysteme weit verbreitet sind. Hinzu kommt, daß sich die Datenmanipulationssprache SQL als Standard durchgesetzt hat und über 100 relationale SQL-Datenbankprodukte im kommerziellen Einsatz sind. Wenn heute vom SQL-Standard gesprochen wird, ist meist der ISO-Standard SQL2[6] gemeint, für 1995 ist der Standard SQL3 geplant [13], der u.a. Rekursion[7] und objektorientierte Eigenschaften berücksichtigt. Trotz Standardisierung gleicht keine Implementierung einer anderen und jedes System bietet über den Standard hinaus zusätzliche Funktionen.

Hier wird exemplarisch auf weitverbreitete oder bekannte Systeme eingegangen. INFORMIX-SQL, ORACLE-SQL*TextRetrieval und DBASE IV bieten Textsuchmöglichkeiten, wobei auf eventuelle Einschränkungen aufgrund von Implementierungsdetails wiederum nicht eingegangen wird, da diese bei neueren Versionen meist nicht mehr bestehen.

[6] auch SQL/92

[7] In SQL2 können nicht alle rekursiven Funktionen dargestellt werden.

Im Datenbanksystem **INFORMIX-SQL** der Informix Software Inc. wird eine Zeichenkettensuche - wie im SQL2-Standard festgelegt - mit dem Schlüsselwort *LIKE* realisiert, das im *WHERE*-Teil einer *SELECT*-Anweisung in der Form

$$char-column\ [NOT]\ LIKE\ 'string'\ [ESCAPE\ escape-character]$$

stehen kann. Die Zeichenkette (*string*) kann Metazeichen mit folgenden Bedeutungen enthalten [14]: % (0 oder beliebig viele beliebige Zeichen) und _ (ein beliebiges Zeichen). Außerdem bietet INFORMIX-SQL zusätzliche Sprachmöglichkeiten bei Verwendung des Schlüsselwortes *MATCHES* (statt *LIKE*), wobei % durch * und _ durch ? ersetzt ist: * (0 oder beliebig viele beliebige Zeichen), ? (ein beliebiges Zeichen), $[a-z]$ (ein Kleinbuchstabe); Angabe von Zeichen in Zeichenklasse wie z. B. $[a0-2-]$ (Zeichen *a*, 0, 1, 2 oder $-$) und Zeichen nicht in Zeichenklasse $[\hat{}a-zA-Z]$ (kein Buchstabe). Metazeichen verlieren ihre besondere Bedeutung durch Voranstellen von \ oder durch Angabe eines *escape*-Zeichens hinter dem Schlüsselwort *ESCAPE*.

Mit **ORACLE-SQL*TextRetrieval** [15] wird das relationale Datenbanksystem Oracle um Textsuchmöglichkeiten bereichert. Zusätzlich zum Standard-SQL Schlüsselwort *LIKE* mit den Metazeichen % und _ werden das Schlüsselwort *CONTAINS*, Symbole für Wortbeziehungen und Möglichkeiten einer phonetischen Suche[8] eingeführt. Es sind z. B. Anfragen folgender Art möglich (aus [15], S. 308):

Beispiel 2.2.1

SELECT *Verfasser, Datum, Text*
FROM *Dokumentation*
WHERE *Verfasser = 'MikeHartstein'*
AND *Datum LIKE '%90'*
AND *Text CONTAINS 'Datenbank'& AND 'Installationsanweisung' < 2*

Es werden alle Texte mit Verfasser und Datum des Verfassers *Mike Hartstein* aus dem mit dem Datum mit '90 endenden Jahr gesucht, die das Wort *Datenbank* oder eines seiner Synonyme (&-Zeichen) und das Wort *Installationsanweisung* oder eines seiner Oberbegriffe (<-Zeichen) der nächsten beiden Ebenen (2) enthalten.

Mit **dBase IV** ist es möglich, auch mit der Sprache SQL zu arbeiten. Der $dBase-IV-SQL-Modus$ wird durch das Kommando *SET SQL ON* aktiviert [16], in dem die übliche Zeichenkettensuche mit dem Schlüsselwort *LIKE* und dessen Metazeichen % und _ möglich ist. Darüberhinaus bietet dBase IV im

[8] Suche mit ähnlich klingenden Wörtern mit Hilfe des Soundex-Algorithmus

Nur – dBase – Modus weitere Textsuchmöglichkeiten im Menü *SUCHEN* [17]. Es kann der erste, letzte oder ein beliebiger Datensatz anhand einer Datensatznummer gesucht werden. Die Suchrichtung wird durch *VORWÄRTS SUCHEN* bzw. *RÜCKWÄRTS SUCHEN* bestimmt. Um eine Suche mit mehreren Zeichenketten gleichzeitig durchzuführen, gibt es wiederum abkürzende Schreibweisen: Zeichenketten können 'Joker' enthalten, hier die Metazeichen ? (ein beliebiges Zeichen) und * (0 oder beliebig viele beliebige Zeichen), Auswahl von Groß- oder Kleinschreibung, Verwendung des Operators *LIKE* oder *SOUNDS LIKE*[9].

Wie bei den Information Retrieval Sprachen sind auch bei der Sprache SQL die logischen Operatoren und bei ORACLE-SQL*TextRetrieval Thesaurusoperatoren verfügbar.

2.3 Interpretations-Unterstützungs-Systeme

Es gibt seit Mitte der 80er Jahre eine Reihe von Systemen, die als Interpretations-Unterstützungs-Systeme aufgefaßt werden können. Eine vergleichende Beschreibung dieser Systeme wie AQUAD, NUDIST oder QUALPRO etc. ist in [1] zu finden. Exemplarisch wird hier auf das im Rahmen des interdisziplinären Forschungsprojektes ATLAS ([1], [18], [19], [20], [21]) von den Informatikern in Zusammenarbeit mit Linguisten und Psychologen entworfene, entwickelte und in SMALLTALK/V 286 implementierte Interpretations-Unterstützungs-System ATLAS/ti eingegangen.

Von den vielfältigen ATLAS/ti Systemfunktionen werden wiederum die Funktionen herausgegriffen, die die Textsuche betreffen. Im System ATLAS/ti lassen sich folgende Sucharten unterscheiden: Inkrementelle Suche, Zeichenkettensuche, Zitat-Retrieval und transitive Suche.

Die **inkrementelle Suche** ist eine Suche mit Zeichen nach bspw. Kodes, Memos, Primärtexten oder Zitaten, die umso spezifischer wird, je mehr Zeichen eingegeben werden. Gegeben sei folgende Kodeliste (aus [19], Abb. 2, S. 71):

Secret
Air
Alchemie
Earth
⋮

Wird beispielsweise der Buchstabe *A* eingegeben, wird der Kode *'Air'*, wird danach das Zeichen *l* eingegeben, der Kode *'Alchemie'* gefunden.

[9] Soundex-Funktion für ähnlich klingende Wörter wie in *ORACLE – SQL*

Bei der **Zeichenkettensuche** kann mit Zeichenketten (Q_1), mit maskierten Zeichenketten[10] am Wortanfang bzw. am Wortende (Q_2 bzw. Q_3), mit disjunktiv[11] verknüpften Zeichenketten - dort 'Suchschwärme' genannt - (Q_4), mit mit Kategoriennamen[12] versehenen Suchschwärmen (Q_5,Q_6) oder mit Kategoriennamen enthaltenden Suchschwärmen nach Textteilen gesucht werden (Q_7) (Beispiele aus [18], S. 64-67):

Q_1 : *qualitativ*
Q_2 : $*ich$
Q_3 : $geh*$
Q_4 : $*arum|*eshalb|Grund*|Ursach*|da|weil$
Q_5 : $\$Kunstlicht$ (z. B. definiert als $Lampe*|Laser|Glühbirne*$)
Q_6 : $\$Naturlicht$ (z. B. definiert als $Sonne*|Kerze*|Stern*$)
Q_7 : $\$Licht$ (z. B. definiert als $\$Kunstlicht|\$Naturlicht$)

Die Suchrichtung kann mit *SEARCHFORWARD* bzw. *SEARCHBACK* bestimmt, eine Suchwiederholung durch *SEARCHAGAIN* erreicht werden.

Beim **Zitat-Retrieval** ist es möglich, mit einem Kode[13] (Q_8), mit konjunktiv[14] verknüpften Kodes (Q_9) oder mit disjunktiv[15] verknüpften Kodes (Q_{10}) nach Zitaten[16] aller Primärtexte (Primärtexte konjunktiv verknüpft)[17] oder eines Primärtextes aller Primärtexte (Primärtexte disjunktiv verknüpft)[18] zu suchen (Beispiele aus ([18], S. 71):

Q_8 : *Gold*
Q_9 : $Gold \wedge Fire$
Q_{10} : $Gold \vee Fire$

Eine **transitive Suche nach Zitaten** liefert alle Zitate, die einem Kode K oder einem in transitiver Beziehung zu K stehendem Kode zugeordnet sind ([18], S. 81-82).

2.4 Textverarbeitungssysteme

Wie in Kapitel 2 einleitend erwähnt, benutzen verschiedene UNIX-Textverarbeitungsprogramme reguläre Ausdrücke (im weiteren kurz URE für **U**nix **Re**gular

[10] * (0 oder beliebig viele beliebige Zeichen)
[11] |-Zeichen
[12] beginnend mit $-Zeichen
[13] Schlagwort für einen Textteil
[14] ∧-Zeichen
[15] ∨-Zeichen
[16] d. h. kodierten Textteilen
[17] dies ist eine dynamische Veränderung des Suchraumes, was sich auf den Bereich T (s. Kapitel 3.2) auswirkt.
[18] dto.

Expressions), um Zeichen bzw. Zeichenketten zu suchen und z. B. zu ersetzen.

Die durch reguläre Ausdrücke beschreibbaren Sprachen werden von endlichen Automaten akzeptiert. Sie stehen auf der untersten Stufe der Chomsky-Hierarchie[19] und sind echte Teilmenge der kontext-freien Sprachen, die echte Teilmenge der kontext-sensitiven Sprachen und diese wiederum echte Teilmenge der rekursiv-aufzählbaren Sprachen sind. Letzere werden von einer Turing-Maschine[20] definiert. Eine rekursive Definition der URE ist z. B. in [22] zu finden.

Die URE werden weitgehend einheitlich von den verschiedenen UNIX-Programmen interpretiert. 'So erlauben die Shell und einige andere Programme, die mit Dateinamen operieren (z. B. find), reguläre Ausdrücke in Dateinamen; die meisten Editoren und Suchprogramme erlauben reguläre Ausdrücke in Suchmustern.' ([23], S. 131). UNIX-Programme, die reguläre Ausdrücke benutzen, sind z.B. der batch-orientierte Editor *sed*, der zeilenorientierte interaktive Texteditor *ed* mit der Erweiterung *ex*, der bildschirmorientierte Editor *vi*, die *grep*-Programme (Suche in Dateien nach Zeilen mit URE) *grep* (eingeschränkte URE), *egrep* (volle Ausdruckfähigkeit der URE mit Alternative, schnellstes Programm), *fgrep* (Zeichenketten ohne Metazeichen) und *awk* (Sprache zur Verarbeitung von Textmustern).

Die nachfolgende aus [23], S. 136 überarbeitete Tabelle gibt einen Überblick über Metazeichen und ihre Bedeutung in den unterschiedlichen UNIX-Programmen:

Tabelle 2.4.1

URE	eingeschränkt	eingeschränkt	vollständig	erweitert
	Dateiname	sed,grep	awk,egrep	ed,ex,vi
1.	?	.	.	.
2.	*	.*	.*	.*
3.	-	*	*	*
4.	-	-	+	\{1,n\}
5.	-	-	?	\{0,1\}
6.	-	-	-	\{n,m\}
7.	[...]	[...]	[...]	[...]
8.	[!...]	[^...]	[^...]	[^...]
9.	-	^muster	^muster	^muster
10.	-	muster$	muster$	muster$
11.	*xyz	-	-	\ <muster
12.	xyz*	-	-	muster\ >
13.	-	-	$s_1\|s_2$	$s_1\|s_2$

[19] Chomskys Modell für natürliche Sprachen
[20] mathematisches Modell eines Allzweck-Computers

Erläuterung
Der Tabellenwert '-' bedeutet, daß die entsprechende Sprachmöglichkeit nicht vorhanden ist. Unter 1. steht das Metazeichen für ein beliebiges Zeichen, unter 2. für eine beliebige Zeichenkette. Das Metazeichen für Zeichenwiederholung steht unter 3. (0-n mal), 4. (1-n mal), 5. (0-1 mal) und unter 6. (n-m mal). K(ein) Zeichen aus einer Zeichenklasse wird unter 8.(7.) ausgedrückt. Soll ein Zeichenmuster (muster) am Anfang oder Ende stehen, gibt es folgende Ausdrucksmöglichkeiten: Zeichen/kette am Zeilenanfang (9.), am Zeilenende (10.), am Wortanfang (11.), am Wortende (12.). Eine Zeichenkette s_1 kann mit einer Zeichenkette s_2 disjunktiv (|-Teichen) verknüpft werden (13.).

3 Anfragesprache IQL-T für Textsuche

Wie einleitend erwähnt und in Kapitel 2 erkennbar, unterscheiden sich die Anfragesprachen der Textsuchsysteme nur oberflächlich[21], weshalb ein Entwurf einer einheitlichen Anfragesprache naheliegt. Dieser erfolgt in der Sprache der Prädikatenlogik der 1. Stufe, für die die Gleichmächtigkeit von modelltheoretischer und beweistheoretischer Behandlung des Schließens gilt[22]. Dazu wird der modelltheoretische Weg nach [24] eingeschlagen. Wesentlich bei dieser Methode ist die strenge Trennung von Objekt- und Metasprache. Der Sinn der Unterscheidung liegt in der Vermeidung von Zirkularitäten der Bedeutung und in der Vermeidung von Paradoxien. Die Objektsprache[23] ist in diesem Fall die Anfragesprache IQL-T, in der Metasprache[24] wird die Bedeutung mit Hilfe von mengentheoretischen und natürlichsprachlichen Elementen festgelegt.

Der rekursive Aufbau der abstrakten Syntax in Kapitel 3.1 ist notwendig, um eine potentiell unendliche Menge an zulässigen Anfragen zu definieren. Anfragen bestehen aus einem λ-Teil[25] und einem Formelteil F[26], der Fragevoraussetzung, wobei die zulässigen Anfragensymbole im Alphabet des Kapitels 3.1 festgelegt sind. Formeln können atomar[27] oder molekular[28] sein. Die Anfragen sind nach dem syntaktischen Aufbau zunächst Zeichenketten ohne Bedeutung. Um ihnen eine Bedeutung zu geben, wird in Kapitel 3.2 eine zur Syntax eng entsprechende, rekursive, extensionale (mengentheoretische) Semantik aufgebaut. Die dort angegebene Interpretation $\Im$ ist eine Abbildung, die jeder Anfrage ungeordnete

[21] im wesentlichen syntaktisch
[22] Gödelscher Vollständigkeitssatz
[23] Definition in Kapitel 3.1: Syntax
[24] Definition in Kapitel 3.2: Semantik
[25] wonach gefragt wird
[26] womit gefragt wird
[27] nicht zusammengesetzt
[28] mit logischen Symbolen $\wedge, \vee, \neg, \exists, \forall$ zusammengesetzt

Teilmengen der in Kapitel 3.2 eingangs definierten Bereiche zuordnet[29].

Beim Aufbau der Anfragesprache in der Sprache der Prädikatenlogik wird wie in [2] auf das Objekt-Attribut-Wert-Modell aus [25] Bezug genommen. Die zu modellierenden Objekte können z. B. Menschen, Pilze, Wolken und wie hier Texte sein, die zusätzlich durch Attribute wie Augenfarbe, Grad der Giftigkeit, Wolkenhöhe und bei Texten durch Autor, Titel etc. charakterisiert sein können. Werte sind mögliche Ausprägungen der Attribute wie z. B. blau, tödlich, 10000m, Arno Schmidt, Zettels Traum etc.. Möchte man die Systemantworten einer Anfrage ordnen, kann das Objekt-Attribut-Wert Modell um einen zusätzlichen Wert erweitert werden. Je nach Modell ist dies z. B. ein Wahrscheinlichkeits-, Zugehörigkeits-,[30] Sicherheitsgrad oder Ähnlichkeitswert. Die Aussage, daß Janos ein sehr blauäugiger Mensch ist, könnte in einem Fuzzy-Modell folgendermaßen dargestellt werden:

$$Augenfarbe(Janos, blau, 0.9).$$

Im vorliegendem Beitrag ist die Semantik für ein Boolesches Objekt-Attribut-Wert-Modell erklärt. Die Texte werden als eine Menge von 2-Tupeln der Art $A(T, V)$ dargestellt, mit A als Attribut, T als Textobjekt und V als Wert. Attribute sind z. B. TX für Freitext, TI für den Titel eines Textes, $TX - L_4$ für die 4. Zeile oder $TX - C_2$ für das 2. Kapitel eines Freitextes TX. Bestünde ein Text T_1 nur aus dem einleitenden Satz des vorliegenden Beitrags, kann dieser mit der hier gewählten Darstellung folgendermaßen aussehen:

$$\begin{array}{l} AU(T_1, Ulrike\ Reiner), \\ TI(T_1, Anfragesprachen\ für\ Textsuchsysteme), \\ TX(T_1, Um\ sich\ mitzuteilen\ und\ zu\ verständigen, \\ spricht\ und\ schreibt\ der\ Mensch,\ es\ entstehen\ Texte.) \\ \vdots \qquad\qquad\qquad\qquad\qquad\qquad \vdots \end{array}$$

Im Spezialfall - wie z.B. bei Dateien - kann es sein, daß eine Textcharakterisierung nur aus einem Textnamen besteht, also z. B.

$$INTERVIEW(T_7, I : sie\ haben\ eben\ gesagt, \ldots).$$

Hier besteht die Textcharakterisierung[31] aus dem Dateinamen $INTERVIEW$, das charakterisierte Objekt ist die Textnummer T_7 für Text 7. Der Wert V ist der Text selber, d. h. I : *sie haben eben gesagt,...* . In Information Retrieval Systemen kommen auch molekulare Attribute, wie z. B. die Attribute BI (basic index), SO (source) und CC (classification code) vor. Diese Attribute sind abkürzende

[29] Boolesches Retrieval
[30] zu einer Fuzzy Menge
[31] d. h. das Attribut

Schreibweisen für eine Menge von atomaren Attributen und stellen keine Sprachbereicherung dar; ein molekulares Attribut $A^m(\ldots)$ kann als molekulare Formel $A^1(\ldots) \vee A^2(\ldots) \vee \ldots \vee A^m(\ldots)$ dargestellt werden. Je nach Datenbank verbergen sich hinter diesen Attributen mehrere Attribute, z. B. steht in der MESSENGER-Datenbank MATH das molekulare Attribut BI für die Attribute TI (title), AB (abstract text), CC (classification text) und UT (uncontrolled term) während in der MESSENGER-Datenbank PHYS andere Attribute, nämlich die Attribute TI, CT(controlled term) und AB zusammenfaßt werden.

Für ein tieferes Verständnis der hier betretenen Gebiete Informationssysteme und Logik, insbesondere Modelltheorie und Interrogativlogik und deren Zusammenhänge, die hier nicht dargelegt werden können sei die Lektüre von [2] und die dort genannte einschlägige Literatur empfohlen. Es folgt nun die Definition der Anfragesprache IQL-T.

3.1 Syntax der Sprache IQL-T

Definition 3.1.1 (Freitext-Alphabet $\aleph$)
Das Freitext-Alphabet $\aleph$ besteht aus einer Menge von Buchstaben, Ziffern und Sonderzeichen z. B. des ISO 8859-1-Zeichencodes: $\aleph := \{\ldots, !, \ldots, \#, \ldots, *, +, 0, \ldots, 9, \ldots, A, B, C, \ldots, \hat{}, \ldots, a, b, c, \ldots \tilde{}, \ldots, \hat{u}, \ldots, \ddot{y}\}$.

Definition 3.1.2 (Zeichen a)
Ein Zeichen (Symbol) **a** ist ein Element aus $\aleph$.

Definition 3.1.3 (Zeichenkette s)
Eine Zeichenkette **s** ist eine Aneinanderreihung[32] von $n \in N$ [33] Zeichen des Freitext-Alphabetes $\aleph$.

Definition 3.1.4 (Wort w)
Ein Wort **w** ist eine aus dem Freitext-Alphabet $\aleph$ aus $n \in N$ Zeichen bestehende, sprachlich sinnvoll gebildete Zeichenkette s, die von $\sqcup$-Zeichen als Wortgrenze umschlossen wird.

Definition 3.1.5 (Satz S)
Ein Satz **S** - als gedanklich zusammenhängende Einheit - besteht aus einer Kette von $n \in N$ Wörtern. $\bullet$-Zeichen umschließen einen Satz.

[32] Verkettung
[33] Die leere Zeichenkette hat mathematische, jedoch keine praktische Bedeutung, deshalb $n \in N$.

Definition 3.1.6 (Paragraph P)
Ein Paragraph **P** besteht aus einer Kette von $n \in N$ Sätzen. ¶-Zeichen umschließen einen Paragraphen.

Definition 3.1.7 (Kapitel C)
Ein Kapitel **C** ist eine Paragraphen-Kette mit $n \in N$ Paragraphen. ◯-Zeichen umschließen ein Kapitel.

Definition 3.1.8 (Freitext, Volltext oder einfach Text TX)
Ein Text **TX** ist eine Kapitel-, Paragraphen-, Satz- oder Wortkette oder im einfachsten Fall eine Zeichenkette.

Definition 3.1.9 (Zitat Q)
Ein Zitat **Q** besteht aus einer Zeichenkette, die von " umschlossen ist.

Definition 3.1.10 (Zeile L)
Eine Zeile **L** besteht aus einer Kette von Wörtern, die am Zeilenanfang mit dem ⌈-Zeichen beginnt und am Zeilenende mit dem ⌋-Zeichen endet.

Erläuterung
Die gewählten Abgrenzungszeichen ␣, •, ¶, ◯, ", ⌈ bzw. ⌋ zur Erkennung von Wörtern, Sätzen, Paragraphen, Kapiteln, Zitaten, Zeilenanfang bzw. Zeilenende sind hier frei gewählte Abgrenzungszeichen, die im Einzelfall entweder bei der Implementierung oder von den Systembenutzern[34] festgelegt worden sind. Von Fall zu Fall kann z. B. das Satzabschlußzeichen • ein . (Punkt), ein ! (Ausrufungszeichen) oder ein ? (Fragezeichen) sein. Bei Bedarf können zusätzlich zu den oben definierten, den Text TX unterteilende Einheiten, weitere, von Abgrenzungszeichen umschlossene Einheiten, wie z. B. Verse definiert werden. Die Wahl der Abgrenzungszeichen als auch die Anzahl der in einem System definierten syntaktischen Einheiten üben einen direkten Einfluß auf die Suchmöglichkeiten aus. Bei einer Suche können bspw. nur dann spezifische Suchbedingungen wie *im gleichen Satz*, *am Zeilenanfang* oder *im gleichen Attribut* angegeben werden, wenn im System Sätze, Zeilen oder Attribute ausgezeichnet sind (vgl. Beispiele in Kapitel 3.3).

Der folgende Beispieltext (Ausschnitt aus [26], Zeile 26-28, ausführlicher in Kapitel 3.3) besteht aus zwei Sätzen, 22 Wörtern, zwei Zeilenanfängen und zwei Zeilenenden:

• ␣da␣␣sag␣␣ich␣␣immer␣␣,␣⌋ ⌈ ␣na␣␣warte␣␣doch␣␣bis␣␣se␣␣voll␣␣ist␣••
␣ich␣␣mach␣⌋ ⌈ ␣die␣␣nur␣␣an␣␣␣,␣␣wenn␣␣se␣␣voll␣␣ist␣•

[34] Durch 'Anklicken' können im ATLAS-System z. B. Zitate festgelegt werden.

Bei Zusammentreffen zweier gleicher Abgrenzungszeichen kann vereinfachend ein Abgrenzungszeichen geschrieben werden, also:

> • ␣ da ␣ sag ␣ ich ␣ immer ␣ , ␣ ⌋ ⌈ ␣ na ␣ warte ␣ doch ␣ bis ␣ se ␣ voll ␣ ist␣• ␣ ich ␣ mach␣⌋ ⌈ ␣ die ␣ nur ␣ an ␣ , ␣ wenn ␣ se ␣ voll ␣ ist␣•

Als nächstes werden die Zeichen der Sprache IQL-T eingeführt, aus denen Anfragen gebildet werden können:

Alphabet

1.	Textsymbole:	$t_1, t_2, \ldots, T_1, T_2, \ldots$
2.	Wertesymbole:	$v_1, v_2, \ldots, V_1, V_2, \ldots$
3.	Zeichenmustersymbole:	$s_1^{n_1,p_1}, s_2^{n_2,p_2}, \ldots$
4.	Wortmustersymbole:	$w_1^{p_1}, w_2^{p_2}, \ldots, w_6^{n_6,o_6}, w_7^{n_7,o_7}, \ldots$
5.	Thesaurussymbole:	$\tau_1, \tau_2, \ldots$ z. B. für $\uparrow^\alpha, \downarrow_\beta, \Uparrow, \Downarrow, \Updownarrow, \Leftrightarrow, \oint$
6.	Prädikatensymbole:	$TX, TX-L, TX-S, TX-P, TX-Q, \ldots,$ $TI-S_i, AB, AB-P_i, \ldots, C, CT, CY, ND,$ $BI, SO, CC, \ldots, EQ, NE, LT, \ldots$
7.	Logische Symbole:	$\neg, \wedge, \vee, \exists, \forall, \lambda$
8.	Technische Symbole:	(,)

Erläuterung

Die Symbole des Alphabetes sind Anfangsbuchstaben englischer Begriffe (t: text, v: value, s: string, w: word, n: number, p: pattern, o: order, C: comparison predicate) oder häufig verwendete bzw. genormte Attributensymbole der Common Command Language (CCL) aus [27], wie z. B. TX (free text), TI (title), AB (abstract), CT (controlled term)[35], CY (publication country), ND (number of document), BI (basic index), SO (source) und CC (classification code). Bei Eindeutigkeit werden Individuensymbole (z. B. Text- oder Wertesymbole), Funktionssymbole (z. B. Thesaurussymbole) und Prädikatensymbole auch ohne Indizes verwendet. t_i bezeichnet Textvariablen, T_i Textkonstanten, v_i Wertevariablen und V_i Wertekonstanten mit $i \in N$.

Mit dem Zeichenmustersymbol $s^{n,p}$ werden Zeichenketten ausgezeichnet, die die im Exponenten genannten Bedingungen erfüllen. Mit dem Exponenten $n \in N$ wird die Zeichenanzahl angegeben, wobei $n \in \{-$ (beliebig), $\sim$ (irrelevant)[36], c (fest), $\leq c$ (mind. c), $\geq c$ (max. c), $\geq c_1 \wedge \leq c_2$ (mind. c_1 und max. c_2)$\}$. Im Exponenten p erfolgt die Angabe des Zeichenmusters (IRE), ein regulärer Ausdruck nach Definition 3.1.11 (s. u.).

[35] Ein festgelegter Begriff, der z. B. in einem Thesaurus verwendet wird.

[36] d. h. Zeichenanzahl spielt keine Rolle, undzwar bei Zeichenmustersymbolen, bei denen das Zeichenmuster keine Metazeichen enthält, vgl. Beispiele 3.3.2, 3.3.3, 3.3.10 usw.

Mit den Wortmustersymbolen $w_i^{p_i}$ (1. Art) bzw. $w_j^{n_j,o_j}$ (2. Art) $(i,j \in N)$ können Wörter bzw. Wortbedingungen formuliert werden. Das Wortmustersymbol w^p (p ist wieder ein wie in Definition 3.1.11 definiertes IRE) ist eine abkürzende Schreibweise für Wörter, d. h. ausgezeichneten Zeichenketten[37]. Statt $s\sqcup voll\sqcup$ kann w^{voll} geschrieben werden. Bei dem Wortmustersymbol $w^{n,o}$ $(n \in N_0)$ steht der Exponent n wiederum für eine Anzahl - hier Anzahl der Wörter, die zwischen den rechts und links des Wortsymbols $w^{n,o}$ stehenden Zeichenketten p_i $(i \in N)$ stehen können. Im Exponenten o wird die Reihenfolge der Zeichenketten p_i $(i \in N)$ festgelegt. Durch einen Rechtspfeil $\rightarrow$ wird ausgedrückt, daß die Zeichenketten in der angegebenen, durch einen Linkspfeil $\leftarrow$, daß sie in umgekehrter Reihenfolge stehen sollen. Ein Rechts-Links-Pfeil $\leftrightarrow$ steht für eine beliebige Reihenfolge der umgebenden Zeichenketten; er ist eine abkürzende Schreibweise für $\rightarrow$ oder $\leftarrow$.

τ_i $(i \in N)$ sind Thesaurussymbole für beliebige generische und analytische Deskriptor-, Begriffs- oder Wortbeziehungen [28], die z. B. in einem Thesaurus oder Wörterbuch festgelegt sind. Ein (polyhierarchischer) Thesaurus [29] kann graphisch als Wald dargestellt werden. Ein Wald ist eine Menge von Bäumen, wobei der Baum ein wohldefinierter mathematischer Begriff ist [38]. Im einzelnen bedeuten die Begriffsbeziehungen $(\alpha, \beta \in N)$: $\uparrow^\alpha$ (α Generationen von Vorfahren), $\downarrow_\beta$ (β Generationen von Nachfahren), $\Uparrow$ (alle direkten Vorfahren), $\Downarrow$ (alle Nachfahren), $\Updownarrow$ (alle direkten Vorfahren und alle Nachfahren), $\Leftrightarrow$ (alle Geschwister einer Tiefe), $\oint$ (alle Synonyme). In kommerziellen Information Retrieval Systemen werden dafür unterschiedliche Thesaurussymbole verwendet, z. B. $\downarrow_5$ lautet *DOWN*5 in der Sprache CCL, $\oint$ heißt *+all* in der Sprache MESSENGER, *RELATE* in der ISO 8777 Norm und & in der Sprache SQL*TextRetrieval.

Prädikatensymbole ohne Bindestrich wie z. B. TX (free text) und AB (abstract) bezeichnen nicht weiter unterteilte *Texte*, Prädikatensymbole mit Bindestrich wie z. B. $TX-L$, $TX-S$, $TX-P$, $TX-Q$, $TI-S_i$, $AB-P_i$ stehen für *Textteile* und bedeuten in der angegebenen Reihenfolge: Zeile (Line), Satz (Sentence), Paragraph (Paragraph), Zitat (Quotation) eines Freitextes, Satz i eines Titels und Paragraph i eines Abstracts. C steht für Vergleichsprädikate zweier Werte wie z. B. EQ für gleich (EQUAL), NE für ungleich (Not Equal), LT für kleiner als oder vor (Less Than), GT für größer als oder nach (Greater Than), LE für kleiner gleich (Less or Equal to), GE für größer als (Greater Equal to).

Es folgt eine rekursive Sprachdefinition für reguläre Ausdrücke oder Muster - IRE (für **IQL** **R**egular **E**xpressions) genannt:

[37] vgl. Definition 3.1.4
[38] rekursive Definition z. B. in [30]

Definition 3.1.11 (IRE)

1. $a, \backslash a, -, \sqcup, \lceil, \rfloor, [a], [\hat{}\, a], [a_1 a_2], [\hat{}\, a_1 a_2], [a_1 a_2 a_3], [\hat{}\, a_1 a_2 a_3], \ldots, [a_1 - a_n], [\hat{}\, a_1 - a_n], [a_{11} - a_{1n} a_{21} - a_{2n}], [\hat{}\, a_{11} - a_{1n} a_{21} - a_{2n}], \ldots$ sind IRE.
2. Wenn s, s_1, s_2 IRE sind, dann sind auch $(s), s_1 s_2, s_1 | s_2$ IRE.

Erläuterung
Es gibt Nuancen bei der Definition von regulären Ausdrücken. Hier betten sie sich in den Kontext einer einheitlichen Anfragesprache für unterschiedliche Textsuchsysteme ein und sind Teil prädikatenlogischer Formeln[39]. Im Unterschied zu den URE[40] wird für eine einsichtige Semantik bei den Zeichenmustersymbolen $s^{n,p}$ die Zeichenanzahl n vom Zeichenmuster p getrennt. Durch diese Trennung als auch dem Aufbau der Terme, Formeln bzw. Anfragen (s.u.) wird Einheitlichkeit erreicht und bei der Definition der IRE fallen im Unterschied zu URE z. B. die Symbole . , $*, +, ?, \backslash <, \backslash \{$ weg, wie sie z. B. bei den URE $s*$ (Kleensche Hülle), $s+$ (positive Hülle) oder $s\backslash\{m,n\backslash\}$ zu finden sind. Unter 1. bezeichnen a bzw. a_i $(i \in N)$ ISO-Zeichen des Freitext-Alphabetes $\aleph$. Wenn a ein Metazeichen ist, wird durch $\backslash a$ die besondere Bedeutung des Zeichens a als Metazeichen aufgehoben. '–' bezeichnet ein beliebiges ISO-Zeichen, $\sqcup$ das Worttrennzeichen, $\lceil$ das Zeichen für Zeilenbeginn und $\rfloor$ für das Zeilenende. Zeichen a_i $(i \in N)$, die gemeinsam von einer eckigen Klammer umschlossen werden, bilden eine Zeichenklasse, wobei jedes der darin genannten Zeichen ein Teil bzw. - wenn durch $\hat{}$ markiert - kein Teil des Zeichenmusters ist. Mit einem in der eckigen Klammer zwischen zwei Zeichen stehenden '–' können Zeichenbereiche angegeben werden. Durch 2. lassen sich Zeichen bzw. Zeichenketten klammern, verketten oder alternativ angeben[41].

Terme

1. Textvariablen t_i, Textkonstanten T_i, Wertevariablen v_i, Wertekonstanten V_i, Wortmustersymbole w^p und Wortmustersymbole mit Thesaurussymbolen $\tau_i w_j^{p_j}$ $(i, j \in N)$ sind Terme. Sei $\zeta_i^{c_i}$ ein (Zeichen- oder Wort-) Mustersymbol mit der Musterbedingung[42] $c_i \in \{n, p, o\}$. Dann ist eine Mustersymbolkette $\zeta_1^{c_1} \zeta_2^{c_2} \zeta_3^{c_3} \ldots \zeta_n^{c_n}$ auch ein Term, wobei Wortmustersymbole der 2. Art $(w^{n,o})$ nur zusammen mit mindestens zwei weiteren Mustersymbolen[43] auftreten können. Ein Wortmustersymbol $w^{n,o}$ steht innerhalb einer Mustersymbolkette stellvertretend genau an der Stelle, an der die n zu ersetzenden Wörter stehen sollen.

[39] als Muster im Exponenten der Mustersymbole (s. auch Terme und Formeln)
[40] vgl. Kapitel 2.4
[41] Zeichenkette s_1 oder Zeichenkette s_2
[42] condition
[43] nur Zeichenmustersymbole oder Wortmustersymbole der 1. Art

2. Das sind alle Terme.

Beispiel 3.1.1
Die Satzteile, die zum Ausdruck bringen, daß von jemandem etwas immer gesagt wird ('da sag ich immer', 'da sagst du immer', ...) kann durch den Term

$$w_1^{da} w_2^{sag(t|st|en)} w_3^{1,\rightarrow} w_4^{immer}$$

ausgedrückt werden. Im Wortmustersymbol w_3 ist angegeben, daß zwischen dem Zeichenmuster $\sqcup da \sqcup sag(t|st|en)\sqcup$ und dem Zeichenmuster $\sqcup immer\sqcup$ genau ein Wort in der angegebenen Reihenfolge stehen soll.

Beispiel 3.1.2
Die Satzteile '*ich mach die nur an*' bzw. '*die mach ich nur an*' können abkürzend mit folgendem, aus Wortmustersymbolen der 1. Art (w_1, w_3), der 2. Art (w_2) und einem Zeichenmustersymbol (s_4) bestehenden Term ausgedrückt werden:

$$w_1^{ich} w_2^{1,\leftrightarrow} w_3^{die} s_4^{\sqcup nur \sqcup an \sqcup}.$$

Formeln

1. Formeln sind 2-stellige Prädikate $P(t,v)$, $P(T,v)$, $C(v_1,v_2)$ bzw. allgemein $P(t,\mathbf{t})$ mit den oben im Alphabet eingeführten Text- und Wertesymbolen bzw. mit $\mathbf{t}$ als ein unter Terme eingeführter Term und P als ein Prädikatensymbol des Alphabetes außer C. Spezialfälle sind $CT(t,w^p)$, $CT(t,\tau w^p)$, $TX(t,w^p)$, $TX(t,\zeta_1^{c_1}\zeta_2^{c_2}\ldots\zeta_n^{c_n})$, $TI-S_1(t,s_1^{n_1,p_2}s_2^{n_2,p_2})$, $TX-L(t,\zeta_1^{c_1}\zeta_2^{c_2})$.

2. Wenn F, F_1, F_2 Formeln sind, dann auch $\neg F, (F_1 \wedge F_2), (F_1 \vee F_2), (\exists t)F$, $(\forall t)F, (\exists v)F, (\forall v)F$.

3. Das sind alle Formeln.

Anfragen

1. Wenn F eine Formel ist, dann ist $(\lambda P_1 \ldots P_l C_1 \ldots C_m)(\lambda x_1 \ldots x_n)F$ eine Anfrage mit x_i $(i \in N)$ als Individuensymbole für Text-, Werte- bzw. Wortsymbole.

2. Das sind alle Anfragen.

3.2 Semantik der Sprache IQL-T

Um den Anfragen eine Bedeutung zu geben, werden **disjunkte Bereiche** von Individuensorten zugrunde gelegt. Gegeben seien folgende, nichtleere Mengen: $\aleph$ (Freitext-Alphabet), Σ (Zeichenketten über $\aleph$), $\mathcal{T}$ (Texte), $\mathcal{V}$ (Werte), $\mathcal{W}$ (Worte), $\mathcal{L} \subseteq \Sigma$ (Zeilen), $\mathcal{S} \subseteq \Sigma$ (Sätze), $\mathcal{P} \subseteq \Sigma$ (Paragraphen), $\mathcal{C} \subseteq \Sigma$ (Kapitel), $\mathcal{Q} \subseteq \Sigma$ (Zitate), Π (Deskriptor-, Begriffs- bzw. Wortbeziehungen). Durch eine Interpretation $\Im$ werden den Individuensymbolen Elemente entsprechender Bereiche und den Prädikatensymbolen Mengen von Individuen bzw. Individuentupeln zugeordnet:

Interpretation der nichtlogischen Symbole

$\Im(t) \in \mathcal{T}$, $\Im(T) \in \mathcal{T}$, $\Im(v) \in \mathcal{V}$, $\Im(V) \in \mathcal{V}$, $\Im(w^p) \in \mathcal{W}$, $\Im(C) \subseteq \mathcal{V} \times \mathcal{V}$, $\Im(TX) \subseteq \mathcal{T} \times \Sigma$, $\Im(TX-L) \subseteq \mathcal{T} \times \mathcal{L}$, $\Im(TX-S) \subseteq \mathcal{T} \times \mathcal{S}$, $\Im(TX-P) \subseteq \mathcal{T} \times \mathcal{P}$, $\Im(TX-C) \subseteq \mathcal{T} \times \mathcal{C}$, $\Im(TX-Q) \subseteq \mathcal{T} \times \mathcal{Q}$, $\Im(AB) \subseteq \mathcal{T} \times \Sigma$, $\Im(AB-P_i) \subseteq \mathcal{T} \times \mathcal{P}$, $\Im(CT) \subseteq \mathcal{T} \times \Pi$, $\Im(CY) \subseteq \mathcal{T} \times \mathcal{V}$, $\Im(ND) \subseteq \mathcal{T} \times \mathcal{V}$.

Terme

1. $\Im(t)$, $\Im(T)$, $\Im(v)$, $\Im(V)$, $\Im(w^p)$ schon erklärt. $\tau_i w_j^{p_j}$ $(i,j \in N)$ und $\zeta_1^{c_1}\zeta_2^{c_2}\zeta_3^{c_3}\ldots\zeta_n^{c_n}$ werden simultan mit den Formeln erklärt.

Formeln

1. $\Im$ ist Modell von $P(t,v)$ gdw $< \Im(t), \Im(v) > \in \Im(P)$.
   $\Im$ ist Modell von $P(T,v)$ gdw $< \Im(T), \Im(v) > \in \Im(P)$.
   $\Im$ ist Modell von $C(v_1,v_2)$ gdw $< \Im(v_1), \Im(v_2) > \in \Im(C)$.
   $\Im$ ist Modell von $P(t,\mathbf{t})$ gdw $< \Im(t, \Im(\mathbf{t}) > \in \Im(P)$.
   $\Im$ ist Modell von $TX(t,w^p)$ gdw $< \Im(t), \Im(w^p) > \in \Im(TX)$.
   $\Im$ ist Modell von $CT(t,w^p)$ gdw $< \Im(t), \Im(w^p) > \in \Im(CT)$.
   $\Im$ ist Modell von $CT(t,\tau w^p)$ gdw $< \Im(t), \Im(w_1^{p_1}) > \in \Im(TX)$ oder
   $< \Im(t), \Im(w_2^{p_2}) > \in \Im(TX)$ oder
   $\vdots$
   $< \Im(t), \Im(w_n^{p_n}) > \in \Im(TX)$.
   $\Im$ ist Modell von $TX(t,\zeta_1^{c_1}\zeta_2^{c_2}\ldots\zeta_n^{c_n})$ gdw $\bigotimes < \Im(t), \Sigma_n > \in \Im(TX)$ $\Sigma_n \in \{\Im(\zeta_1^{c_1}) \times \ldots \times \Im(\zeta_n^{c_n})\} \subseteq \Sigma$
   $\Im$ ist Modell von $TI-S_1(t, s_1^{n_1,p_1} s_2^{n_2,p_2})$ gdw $\bigotimes < \Im(t), \Sigma_n > \in \Im(TI-S_1)$ $\Sigma_n \in \{\Im(s_1^{n_1,p_1}) \times \Im(s_2^{n_2,p_2})\} \subseteq \Sigma$

	$\Im$ ist Modell von $TX - L(t, \zeta_1^{c_1}\zeta_2^{c_2})$	gdw	$\bigotimes < \Im(t), \Sigma_n > \in \Im(TX - L)$ $\Sigma_n \in \{\Im(\zeta_1^{c_1}) \times \Im(\zeta_2^{c_2})\} \subseteq \Sigma$
	⋮		⋮
2.	$\Im$ ist Modell von $\neg F$	gdw	$\Im$ ist nicht Modell von F.
	$\Im$ ist Modell von $(F_1 \wedge F_2)$	gdw	$\Im$ ist Modell von F_1 und $\Im$ ist Modell von F_2.
	$\Im$ ist Modell von $(F_1 \vee F_2)$	gdw	$\Im$ ist Modell von F_1 oder $\Im$ ist Modell von F_2.
	$\Im$ ist Modell von $(\exists t)$ F	gdw	$\Im_t^{\mathrm{T}}$ ist Modell von F für mindestens ein $\mathrm{T} \in \mathcal{T}$.
	$\Im$ ist Modell von $(\forall t)$ F	gdw	$\Im_t^{\mathrm{T}}$ ist Modell von F für alle $\mathrm{T} \in \mathcal{T}$.

Erläuterung

Unter 1.[44] wird die Wahrheitsbedingung definiert. $\Im$ ist Modell einer Formel, wenn es eine Belegung gibt, so daß die Formel gilt. Anders gesagt: Eine Formel ist bei einer Interpretation genau dann wahr, wenn das Individuentupel ein Element der durch das Attribut[45] definierten Klasse ist, ansonsten ist die Formel nicht wahr. Die Wörter $w_1^{p_1}, \ldots, w_n^{p_n} (n \in N)$ bezeichnen je nach Thesaurussymbol τ[46] z. B. Vorfahren, Nachkommen, Geschwister oder Synonyme eines polyhierarchischen Thesaurus' bzw. Wörterbuches. $\bigotimes$ mit der Bedingung für Σ_n steht für ein metasprachliches Oder und ist eine abkürzende Schreibweise für Σ_1 oder Σ_2 oder ... oder Σ_n. Die Anzahl $n \in N$ der Zeichenketten Σ_n hängt von den Musterbedingungen c_i der Mustersymbolkette $\zeta_1^{c_1}\zeta_2^{c_2}\zeta_3^{c_3}\ldots\zeta_n^{c_n}$ ab. Unter 2.[47] wird der Modellbegriff durch Induktion über dem Aufbau der Formeln erklärt, d. h. für molekulare Formeln. Bei der Interpretation der Formeln mit Existenz- bzw. Allquantor muß die Interpretation $\Im$ an der Stelle t abgeändert werden. Für alle an dieser Stelle abgeänderten Abbildungen wird $\Im_t^{\mathrm{T}}$ geschrieben, wobei T den Bereich $\mathcal{T}$ durchläuft. Die Interpretation $\Im_t^{\mathrm{T}}$ ist von der Interpretation $\Im$ höchstens für die Variable t verschieden, aber nicht notwendigerweise. Die Quantoren anderer Variablen wie z. B. v werden analog durch sortentreue Interpretation der gebundenen Variablen abgebaut.

Anfragen

1. $\Im((\lambda P_1 \ldots P_l C_1 \ldots C_m)(\lambda x_1 \ldots x_n)F) = \{< \mathcal{A}_1, \ldots, \mathcal{A}_m, \delta_1, \ldots, \delta_n >:$

   $\Im_{P_1 \ldots C_m x_1 \ldots x_n}^{\mathcal{A}_1 \ldots \mathcal{A}_m \delta_1 \ldots \delta_n}$ ist Modell von F$\}$

[44] Rekursionsanfang
[45] z. B. $\Im(P)$
[46] s. Alphabet unter 5.
[47] Rekursionsschritt

3.3 Beispielanfragen in der Sprache IQL-T

Die Anfragesprache IQL-T ist so konstruiert, daß damit alle[48] Suchwünsche formuliert werden können. Im Unterschied hierzu bieten die verfügbaren Anfragesprachen der Textsuchsysteme nicht immer alle Formulierungsmöglichkeiten. So ist es z. B. mit Anfragesprachen von Information Retrieval Systemen nicht möglich, Anfragen mit regulären Ausdrücken zu formulieren, während die Wortmaskierung z. B. bei Anfragesprachen von Datenbanksystemen oder Textverarbeitungssystemen nur teilweise oder überhaupt nicht möglich ist.

In diesem Kapitel wird der Gebrauch der formalen Sprache IQL-T anhand von Beispielen, die auf zwei Beispieltexte Bezug nehmen, veranschaulicht. Bei dem ersten Text [26] handelt es sich um einen Auschnitt eines Interviews, das im Rahmen eines Forschungsprojektes zum Thema Umweltbewußtsein und Wohnen im Juli 1990 durchgeführt wurde. Die im Text auftretenden Transkriptionszeichen [31] bedeuten: * steht für eine Kurzpause, ** für eine längere Pause und *x* für eine x-Sekundenpause. // meint einen Wort- oder Satzabbruch und # deutet Simultansprechen an; Kleinschreibung kennzeichnet den transkribierten Text, Großschreibung innerhalb von Klammern einen Kommentar oder eine unverständliche Stelle; Großschreibung ohne Klammern bedeutet Betonung. Es folgt der Text mit Angabe der Zeilennummern (1-37):

T_1: Interview zum Thema Umweltbewußtsein und Wohnen:

1. I: sie haben eben gesagt, also als die
2. startbahn west gebaut werden sollte, also
3. in diesen auseinandersetzungen, da hat
4. sich auch bei ihnen wohl in sachen
5. umweltbewußtsein was ** also das g// (CA.
6. 2 UNVERSTÄNDLICHE WORTE). gibts da noch
7. andere ereignisse, die wenn sie so
8. zurückdenken.
9.
10. A: ja STÄNDIG. hm brauch man eigentlich nich
11. zurückdenken ne. dis ozonloch äh die
12. ganzen umweltkatastrophen, die jetzt sind.
13. für meine begriffe hängt dis alles, dis
14. erdbeben, für meine begriffe hä// äh äh,
15. ich weiß nicht, dis hängt irgendwie alles
16. mit, mit dem ** ja mit der verSCHWENDUNG
17. zusammen. wie die leute d// äh ds daß man

[48] Eventuell müssen im Einzelfall zusätzliche Definitionen, Symbole, Terme oder Formeln eingeführt werden, was durch einfaches Hinzufügen geschieht.

18. nicht genug NACHdenkt. ich m paß auch auf
19. beim haarspray, * also dis kein fkc äh
20. umweltbewußtes haarspray ** # öhm
21.
22. I: hm hm #
23.
24. A: (RÄUSPERN) meine mutter macht zum beispiel
25. die waschmaschine äh wegen, spülmaschine
26. wegen n paar tassen an. da sag ich immer,
27. na warte doch bis se voll ist. ich mach
28. die nur an, wenn se voll ist. ** ja und
29. man, man liest dis ja auch in der ZEItung
30. immer. Dis öh *3* ja gedanken, auf der
31. andern seite dis, ich finde auch *3*
32. gedanken alleine äh äh hilft nicht mehr,
33. daß man sich macht. man muß auch was äh
34. selbst dazu beitragen. aber ich weiß nicht
35. was. und also bis was ich mache, aber *
36. was soll man persönlich jetzt ähm ** mehr
37. noch machen. (RÄUSPERN)

Der zweite Text T_2: *Die Geschichte von den Flöhen* stammt aus [32]:

1. Es war einmal ein Mann, der hatte einen Hund. Der Hund hatte Flöhe.
2. Die Flöhe verbreiteten sich im ganzen Haus. Der Mann mußte also
3. etwas dagegen unternehmen.
4.
5. Zunächst versuchte er, sie einzeln mit einer Fliegenklatsche zu
6. erledigen. Das erwies sich als ausgesprochen unwirksame Methode.
7. Dann versuchte er es mit einer Flohklatsche - ebenfalls ohne nennens-
8. werten Erfolg.
9.
10. Da kam ihm plötzlich der Gedanke: 'Es gibt doch auch noch die
11. Wissenschaft! Wissenschaft hat Erfolg! Mit moderner amerikanischer
12. Ausrüstung müßte es gehen!' Er besorgte sich eine Dose mit
13. Giftspray - 'Garantiert tödlich bei allen Arten von Flöhen' -
14. und versprühte es reichlich im ganzen Haus. Und selbstverständlich
15. waren nach drei Tagen alle Flöhe tot. Freudig rief er: 'Dies
16. Flohspray ist wunderbar! Dies Flohspray wirkt enorm!'
17.
18. Aber der Mann lag völlig schief. In Wirklichkeit war das Flohspray
19. völlig unwirksam. Tatsächlich war folgendes geschehen: Obwohl das
20. Spray selbst völlig unwirksam war, roch es äußerst stark. Deshalb

21. mußte der Mann alle Fenster und Türen öffnen, um zu lüften.
22. Infolge der Zugluft erkälteten sich die armen Flöhe und starben.

Die beiden Texte T_1 und T_2 werden - wie in Kapitel 3 einleitend beschrieben - im Objekt-Attribut-Wert-Modell als Menge von 2-Tupeln der Art $A(T, V)$ dargestellt, auf die die Interpretation $\Im$ beim Rekursionanfang Bezug nimmt. Die Interpretation der Prädikate[49] lautet:

$$\Im(TI) = \{ \quad < T_1,\ \textit{Interview zum} \ldots >, \\ < T_2,\ \textit{Die Geschichte} \ldots >\}.$$

Entsprechend werden alle anderen Objekt-Attribut-Werte dargestellt, also z. B.

$$\Im(TX - L_1) = \{ \quad < T_1,\ \textit{I : sie haben} \ldots \textit{als die} >, \\ < T_2,\ \textit{Es war einmal} \ldots \textit{hatte Flöhe.} >\}.$$

Analog werden die übrigen Zeilen, Sätze, Paragraphen, Kapitel und der Gesamttext repräsentiert. Es wird angenommen, daß beide Texte aus einem Kapitel, das Kapitel des Textes T_1 aus drei Paragraphen (Zeilen 1-9, 10-23, 24-37), das Kapitel des Textes T_2 aus vier Paragraphen (Zeilen 1-4, 5-9, 10-17, 18-22) bestehen. Sätze und Zitate stimmen nicht mit Zeilen überein:

$$\Im(TX - S_1) = \{ \quad < T_1,\ \textit{I : sie haben} \ldots \textit{WORTE}). >, \\ < T_2,\ \textit{Es war einmal} \ldots \textit{hatte einen Hund.} >\}; \\ \Im(TX - S_2) = \{ \quad < T_1,\ \textit{gibts da} \ldots \textit{zurückdenken.} >, \\ < T_2,\ \textit{Der} \ldots \textit{Flöhe} >\}.$$

Entsprechend werden die anderen Sätze repräsentiert. Als Zitate seien einerseits die 'man'-Sätze des Textes T_1 festgelegt:

$$\Im(TX - Q_1) = \{ \quad < T_1,\ \textit{hm brauch} \ldots \textit{ne.} >\}; \\ \Im(TX - Q_2) = \{ \quad < T_1,\ \textit{wie die leute} \ \ldots \ \textit{NACHdenkt.} >\}; \ldots; \\ \Im(TX - Q_6) = \{ \quad < T_1,\ \textit{und also} \ \ldots \ \textit{noch machen.} >\}$$

und andererseits die 'ich'-Sätze:

$$\Im(TX - Q_7) = \{ \quad < T_1,\ \textit{für meine Begriffe} \ \ldots \ \textit{zusammen.} >\}; \\ \Im(TX - Q_8) = \{ \quad < T_1,\ \textit{ich paß} \ldots \ \textit{öhm} >\}; \ldots; \\ \Im(TX - Q_{11}) = \{ \quad < T_1,\ \textit{aber ich} \ \ldots \ \textit{nicht was.} >\}.$$

Sei CO das Code-Attribut, dann ist

$$\Im(CO) = \{ \quad < TX - Q_1,\ \textit{man} >, \ldots, < TX - Q_6,\ \textit{man} >, \\ < TX - Q_7,\ \textit{ich} >, \ldots, < TX - Q_{11},\ \textit{ich} >\}.$$

Es folgen nun Beispielanfragen und deren Antworten ausgewählter Sprachen des

[49] z. B. stellt $\Im(TI)$ das Attribut *title* dar

2. Kapitels, die mit Q_{L_n}[50] bezeichnet werden, wobei L für ATL (Atlas/ti), INF (INFORMIX-SQL), IQL (Intermediary Query Language), ISO (ISO 8777), MES (MESSENGER) und URE[51] (erweiterte Unix Regular Expressions nach Tabelle 2.4.1) steht. Auf die schrittweise Interpretation[52] der Anfragen wird aus Platzgründen vezichtet. Interessierte Leser finden dafür genügend Beispiele in [2].

Beispiel 3.3.1

Anfrage: Welche Texte enthalten die Zeichenketten *'spülmaschine'* oder *'waschmaschine'*?

Antwort: $\{T_1\}$

Q_{IQL_1}: $(\lambda t)\ TX(t, s^{\sim,(spül|wasch)maschine})$

Q_{ATL_1}: *spülmaschine | waschmaschine*

Q_{INF_1}: *SELECT text FROM tabelle WHERE text MATCHES '* [spülwach]maschine*';*

Q_{ISO_1}: *FIND spülmaschine OR waschmaschine*

Q_{MES_1}: *S spülmaschine OR waschmaschine*

Q_{URE_1}: *(spül|wasch)maschine*

Beispiel 3.3.2

Anfrage: Welcher Text enthält in welcher Zeile die Zeichenkette *'ozo'*?

Antwort: $\{(\{T_1\}, \{11\})\}$

Q_{IQL_2}: $(\lambda t)\ (\lambda TX - L)\ TX - L(t, s^{\sim,ozo})$

Q_{ATL_2}: **ozo**

Q_{URE_2}: *ozo*

Beispiel 3.3.3

Anfrage: In welchen Textzeilen ist Simultansprechen notiert?

Antwort: $\{20, 22\}$

Q_{IQL_3}: $(\lambda TX - L)\ (\exists t)\ TX(t, s^{\sim,\#})$

Q_{INF_3}: *SELECT ... MATCHES '* #*';*

Q_{URE_3}: #

Beispiel 3.3.4

Anfrage: Welche Texte enthalten ein beliebiges Zeichen?

Antwort: $\{T_1, T_2\}$

Q_{IQL_4}: $(\lambda t)\ TX(t, s^{1,-})$

Q_{INF_4}: *SELECT ... MATCHES '?'*; oder *SELECT ... LIKE '_';*

Q_{ISO_4}: #

Q_{MES_4}: !

Q_{URE_4}: . bzw. ? bei Dateinamen

---

[50] L: language, n: number

[51] Das Suchkommando wird hier nicht explizit angegeben, da es unterschiedlich lautet, z. B. leitet ein Schrägstrich im *vi* eine Suche ein, im (e)*grep* werden die URE hinter der $-e$ Option angegeben.

[52] rekursiver Abbau

Beispiel 3.3.5
Anfrage: Welche Texte enthalten genau vier beliebige Zeichen?
Antwort: $\{T_1, T_2\}$
Q_{IQL_5}: $(\lambda t)\ TX(t, s^{4,-})$
Q_{ISO_5}: ####
Q_{MES_5}: !!!!
Q_{URE_5}: .... oder .\{4\}

Beispiel 3.3.6
Anfrage: Welche Texte enthalten beliebig viele beliebige Zeichen?
Antwort: $\{T_1, T_2\}$
Q_{IQL_6}: $(\lambda t)\ TX(t, s^{*,-})$
Q_{ISO_6}: ?
Q_{URE_6}: .*

Beispiel 3.3.7
Anfrage: Welche Texte enthalten maximal 2 beliebige Zeichen?
Antwort: $\{T_1, T_2\}$
Q_{IQL_7}: $(\lambda t)\ TX(t, s^{\leq 2,-})$
Q_{ISO_7}: ?2
Q_{MES_7}: ##
Q_{URE_7}: .\{,2\}

Beispiel 3.3.8
Anfrage: Welche Texte enthalten mindestens 1500 beliebige Zeichen?
Antwort: $\{T_1\}$
Q_{IQL_8}: $(\lambda t)\ TX(t, s^{\geq 1500,-})$
Q_{URE_8}: .\{1500,\}

Beispiel 3.3.9
Anfrage: In welchen Textzeilen welcher Texte sind betonte Textstellen, die mindestens 8 Zeichen und höchstens 10 Zeichen enthalten?
Antwort: $\{(\{16\}, \{T_1\})\}$
Q_{IQL_9}: $(\lambda TX - L)\ (\lambda t)\ TX(t, s^{\geq 8 \wedge \leq 10,[A-Z]})$
Q_{URE_9}: $[A-Z]\backslash\{8,10\backslash\}$

Beispiel 3.3.10
Anfrage: Welche Textzeilen des Textes T_2 enthalten keine Umlaute?
Antwort: $\{3, 4, 6, 7, 8, 9, 11, 16, 17\}$
$Q_{IQL_{10}}$: $(\lambda TX - L)\ TX - L(T_2, s^{\sim,[\hat{}\ \ddot{A}\ddot{O}\ddot{U}\ddot{a}\ddot{o}\ddot{u}]})$
$Q_{URE_{10}}$: $[\hat{}\ \ddot{A}\ddot{O}\ddot{U}\ddot{a}\ddot{o}\ddot{u}]$
$Q_{INF_{10}}$: $SELECT \ldots\ MATCHES\ '*[\hat{}\ \ddot{A}\ddot{O}\ddot{U}\ddot{a}\ddot{o}\ddot{u}]*';$

Beispiel 3.3.11
Anfrage: Welche Zeilen des Textes T_1 enthalten *'die'* am Zeilenanfang oder *'tung'* am Zeilenende?
Antwort: $\{25, 28, 29\}$
$Q_{IQL_{11}}$: $(\lambda TX - L)\ TX - L(T_1, s^{\sim,([die|tung])})$
$Q_{URE_{11}}$: $\hat{}die|tung\$$

Beispiel 3.3.12
Anfrage: Welche Textzeilen welcher Texte enthalten '*Floh*' am Wortanfang oder '*spray*' am Wortende?
Antwort: $\{(\{19,20\},\{T_1\}),(\{7,13,16,18\}\{T_2\})\}$
$Q_{IQL_{12}}$: $(\lambda TX-L)\ (\lambda t)\ TX-L(t, s^{\sim,(\sqcup Floh|spray\sqcup)})$
$Q_{URE_{12}}$: $\backslash < Floh|spray \backslash >$
$Q_{ATL_{12}}$: $Floh * \mid * spray$[53]

Beispiel 3.3.13
Anfrage: Welche Textzeilen welcher Texte enthalten '*Fl*' am Wortanfang und '*ray*' am Wortende?
Antwort: $\{(\{16,18\},\{T_2\})\}$
$Q_{IQL_{13}}$: $(\lambda TX-L)\ (\lambda t)\ TX-L(t, s_1^{\sim,\sqcup Fl} s_2^{-,-} s_3^{\sim,ray\sqcup})$
$Q_{URE_{13}}$: $\backslash < Fl.*ray \backslash >$

Beispiel 3.3.14
Anfrage: In welchen Textzeilen des Textes T_1 stehen längere Pausen?
Antwort: $\{5,16,20,28,36\}$
$Q_{IQL_{14}}$: $(\lambda TX-L)\ TX-L(T_1, s^{\sim,**})$
$Q_{URE_{14}}$: $\backslash * \backslash *$
$Q_{INF_{14}}$: $SELECT \ldots\ MATCHES\ '* \backslash * \backslash * *'$;[54]

Beispiel 3.3.15
Anfrage: In welchen Textzeilen des Textes T_1 stehen beliebig viele '*'?
Antwort: $\{5,16,20,28,36\}$
$Q_{IQL_{15}}$: $(\lambda TX-L)\ TX-L(T_1, s^{-,*})$
$Q_{URE_{15}}$: $\backslash * *$
$Q_{INF_{15}}$: $SELECT \ldots\ MATCHES\ '* \backslash * *'$;

Beispiel 3.3.16
Anfrage: In welchen Textzeilen des Textes T_1 sind Pausen[55]notiert?
Antwort: $\{5,16,19,20,28,36\}$
$Q_{IQL_{16}}$: $(\lambda TX-L)\ TX-L(T_1, s^{\geq 1,*})$
$Q_{URE_{16}}$: $\backslash * +$

Beispiel 3.3.17
Anfrage: In welchen Textzeilen des Textes T_1 steht keine Pause[56] oder eine Kurzpause[57]?
Antwort: $\{1,2,\ldots,37\}$
$Q_{IQL_{17}}$: $(\lambda TX-L)\ TX-L(T_1, s^{\geq 0 \wedge \leq 1,*})$
$Q_{URE_{17}}$: $\backslash *?$

---

[53]Zeichenkettensuche
[54]Suche nach Zeilen allerdings nicht möglich, nur ob überhaupt im Text.
[55]ein oder beliebig viele '*'
[56]0 '*'
[57]1 '*'

Beispiel 3.3.18

Anfrage: Welche Zitate des Textes T_1 enthalten die Zeichenkette 'nich'?
Antwort: $\{1,2,4\}$
$Q_{IQL_{18}}$: $(\lambda TX - Q)\ TX - Q(T_1, s^{\sim,nich})$
$Q_{URE_{18}}$: $nich$
$Q_{INF_{18}}$: $SELECT \ldots\ MATCHES\ '*nich*'$;

Beispiel 3.3.19

Anfrage: Welche Zitate des Textes T_1 sind mit dem Kode 'man' und dem Kode 'ich' kodiert?
Antwort: $\{4,6\}$
$Q_{IQL_{19}}$: $(\lambda TX - Q)\ (CO(TX - Q, man)\ \wedge\ CO(TX - Q, ich))$
$Q_{ATL_{19}}$: $man\ \wedge\ ich$[58]

Beispiel 3.3.20

Anfrage: Welche Zitate des Textes T_1 enthalten das Wort 'man' mehrfach[59]?
Antwort: $\{3\}$
$Q_{IQL_{20}}$: $(\lambda TX - Q)\ TX - Q(T_1, s^{>1,\sqcup man\sqcup})$
$Q_{URE_{20}}$: $\backslash < man \backslash > .* \backslash < man \backslash >$[60]

Beispiel 3.3.21

Anfrage: In welchen Textzeilen des Textes T_2 steht das Wort, das mit *'ver'* beginnt, gefolgt von beliebig vielen Wörtern und danach die Zeichenkette *'im ganzen Haus'*?
Antwort: $\{2,14\}$
$Q_{IQL_{21}}$: $(\lambda TX - L)\ TX - L(T_2, s_1^{\sim,\sqcup ver} s_2^{-,-} s_3^{\sim,\sqcup} w_4^{-,\rightarrow} s_5^{\sim,\sqcup im\sqcup ganzen\sqcup Haus\sqcup})$
$Q_{URE_{21}}$: $ver.*im\ ganzen\ Haus$
$Q_{ATL_{21}}$: $ver*\ im\ ganzen\ Haus$[61]
$Q_{MES_{21}}$: $S\ ver?\ (nW)$[62] $im\ ganzen\ Haus$

Beispiel 3.3.22

Anfrage: In welchen Textzeilen des Textes T_2 stehen die Wörter *'Flöhe'* und *'sich'* in beliebiger Reihenfolge mit höchstens 3 Wörtern Abstand?
Antwort: $\{2,22\}$
$Q_{IQL_{22}}$: $(\lambda TX - L)\ TX - L(T_2, w_1^{Flöhe} w_2^{\leq 3,\leftrightarrow} w_3^{sich})$
$Q_{URE_{22}}$: $Flöhe.*sich|Flöhe.*.*sich|Flöhe.*.*.*sich|$ $sich.*Flöhe|sich.*.*Flöhe|sich.*.*.*Flöhe$[63]
$Q_{ISO_{22}}$: $FIND\ Flöhe\ \%3\ sich$
$Q_{MES_{22}}$: $S\ Flöhe\ (3A)\ sich$

[58] Zitatretrieval
[59] mehr als einmal
[60] nur Zeilen, keine Zitate möglich
[61] nur innerhalb Zeilen möglich
[62] beliebiger Wortabstand nicht möglich, z. B. n=1000 setzen
[63] bei z. B. 10 Wörtern Abstand sehr umfangreiche Anfrageformulierung

Beispiel 3.3.23

Anfrage: In welchen Textzeilen des Textes T_1 stehen das Wort *'mit'* und das Wort, das mit *'de'* beginnt, gefolgt von genau 1 Zeichen, direkt hintereinander?

Antwort: $\{16\}$

$Q_{IQL_{23}}$: $(\lambda TX - L)\ TX - L(T_1, w_1^{mit} w_2^{0,\rightarrow} s_3^{\sim,de} s_4^{1,-} s_5^{\sim,\sqcup})$

$Q_{URE_{23}}$: *mit de.*

$Q_{MES_{23}}$: *S mit (W) de!*

$Q_{ISO_{23}}$: *FIND mit!de#*

Beispiel 3.3.24

Anfrage: Welche Texte enthalten im 3. Paragraphen zusammen die Zeichenkette *'edanke'* und das Zeichen *'n'* gefolgt von max. drei beliebigen Zeichen gefolgt von dem Zeichen *'h'*?

Antwort: $\{T_1, T_2\}$

$Q_{IQL_{24}}$: $(\lambda t)\ (TX - P_3(t, s_1^{\sim,edanke}) \wedge TX - P_3(t, s_2^{\sim,n} s_3^{\leq 3,-} s_4^{\sim,h}))$

$Q_{MES_{24}}$: $S\ ?edanke?\ (P)\ n\#\#\#h$[64]

4 Ausblick

Der vorliegende Artikel führt in die Welt der Freitextsuche ein. Es zeigt sich, wie z. B. bei Anfragesprachen für Fakten-, Bibliographie- und Zeitreihendatenbanken, daß die Vielfalt der sprachlichen Mittel unerschöpflich ist. Dieser wird mit der einheitlichen Anfragesprache IQL-T begegnet. Die hier vorstellte Sprache IQL-T kann für Hypertext-[65] bzw. Hypermediasysteme verallgemeinert werden[66]: Anstelle der Textsymbole t bzw. T können Objektsymbole o bzw. O für beliebige Objekte wie z. B. für Bilder, Filme, Musikstücke, Tabellen, Graphiken, Formeln und Diagramme eingeführt werden. Für Objektteile kann dann wiederum die Bindestrich-Notation verwendet werden, z. B. $PI - 1$ für ein Musikstück (piece of music), 1. Satz. Bei Hypertext- bzw. Hypermediasystemen ist nicht nur über der Menge der Begriffe, sondern auch über der Menge der Objekte eine Struktur festgelegt. Zu den atomaren Formeln $P(o, \mathbf{t})$ für Objekt-Wert Beziehungen bzw. $C(v_1, v_2)$ für Wert-Wert Beziehungen treten Formeln wie $R(o_1, o_2)$ (R: relation) hinzu. Dadurch kann mit und auch nach Begriffs- bzw. Objektbeziehungen gefragt werden. R können beliebige Beziehungen wie z. B. Musikstück A ist schöner als Musikstück B, Film C ist aufregender als Film D usw. sein. Hypermediasysteme ermöglichen einerseits klassische Anfragemöglichkeiten mit Schlagwörtern und Freitextmustern, aber auch Navigationsmöglichkeiten in Suchräumen, die man gerade bei dem Schritt von den Netzwerkmodellen

[64] intellektuell prüfen, ob im 3. Paragraphen

[65] Z. B. ist es im System ATLAS/ti möglich, nach allen Kodes, Memos und Zitaten zu suchen, die mit einem gegebenen Zitat in Verbindung stehen.

[66] Aus der Sicht der Prädikatenlogik ist es eine Spezialisierung.

zu den relationalen Datenbankmodellen stolz ad acta gelegt hatte: der Benutzer sollte seinen Suchwunsch nicht mehr prozedural, sondern 'nur noch' deklarativ formulieren müssen. Ein wichtiger Unterschied besteht jedoch: bei Hypermediasystemen wird der prozedurale Weg einfach durch 'Anklicken' erzeugt, d. h. die Benutzeroberflächen haben sich hierzu wesentlich vereinfacht. Ob sich durch Hypermediasysteme prinzipiell neue Suchmöglichkeiten eröffnen oder aber schon in 'alten' Systemen enthalten sind, ist zukünftiger Forschungsgegenstand der Autorin.

Dank

Ich danke Thomas Muhr für seine engagierte Durchsicht und Peter Deussen für Hilfestellungen, Tips und Tricks beim Umgang mit LaTeX.

Literatur

[1] Andreas Böhm; Andreas Mengel; Thomas Muhr; Josef Willenborg, (Hrsg.). *Methodenentwicklung für ein 'Archiv für Technik, Lebenswelt und Alltagssprache - Endbericht'.* Bericht aus dem Interdisziplinären Forschungsprojekt ATLAS, Nr. 93-3. TU Berlin, Hardenbergstr. 28, 10623 Berlin, 1993.

[2] Ulrike Reiner. *Anfragesprachen für Informationssysteme.* Reihe Informationswissenschaft der DGD. Band 1. Deutsche Gesellschaft für Dokumentation, Frankfurt am Main, 1991.

[3] Gerard Salton. *Automatic Information Organization and Retrieval.* McGraw-Hill Book Company, New York u.a., 1968.

[4] Josef L. Staud. *Online Datenbanken - Aufbau, Struktur, Abfragen.* Addison-Wesley Publishing Company, Bonn u.a., 1991.

[5] AFI Arbeitsgemeinschaft Fachinformation, (Hrsg.). *Who is Who. Adreßbuch der Online-Szene 1992/93.* b.team B. Breidenstein GmbH, Frankfurt am Main 1, 1993.

[6] American Chemical Society, 2540 Olentangy River Road, P.O. Box 3012, Columbus, Ohio 43210. *Using CAS Online - The Registry File, Vol. III, Dictionary Searching*, June 1985.

[7] Gerard Salton; Michael J. McGill, (Hrsg.). *Introduction to Modern Information Retrieval.* McGraw-Hill International Book Company, London u.a., 1983.

[8] Kamran Parsaye; Mark Chignell; Setrag Khoshafian; Harry Wong. *Intelligent Databases. Object-Oriented, Deductive Hypermedia Technologies.* John Wiley & Sons, Inc., New York u.a., 1989.

[9] Saiedeh von Keitz; Wolfgang von Keitz; Hubertus Gerlach. *Modernes Online-Retrieval. Der Weg zu den Wissensspeichern der Welt am Beispiel der DIALOG-Datenbanken.* VCH Verlagsgesellschaft mbH, Weinheim u.a., 1993.

[10] Malcolm Bain; Richard Bland; Lou Burnard; Jon Duke; Colin Edwards; David Lindsey; Nicholas Rossiter; Peter Willett. *Free Text Retrieval Systems - A Review and Evaluation.* Taylor Graham, 500 Chesham House, 150 Regent Street, London, 1989.

[11] ISO 8777 tc 46/sc 4 Documentation. Commands for Interactive Text Searching, 1993.

[12] E. F. Codd. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 13(6):377–387, June 1970.

[13] Setrag Khoshafian; Arvola Chan; Anna Wong; Harry K.T. Wong. *A Guide to Developing Client/Server SQL Applications.* The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, Inc., San Mateo, CA 94403, 1992.

[14] Jonathan Leffler. *Using Informix-SQL.* Addison-Wesley Publishing Company, Bonn u.a., 1989.

[15] Jürgen Dreller. SQL*TextRetrieval. *Nachrichten für Dokumentation*, 42(4):306–309, August 1991.

[16] Jack L. Hursch; Carolyn J. Hursch. *dBASE IV SQL. Ein Leitfaden zum effektiven Einsatz von SQL unter dBase IV.* Markt & Technik Verlag AG, München, 1989.

[17] Alan Simpson. *Das dBASE IV Buch.* SYBEX-Verlag, Düsseldorf u.a., 1989.

[18] Thomas Muhr. *ATLAS/ti - Computerunterstützte Textinterpretation. Manual zur Version 1.0 D*, 1993.

[19] Thomas Muhr. Atlas/ti - ein Interpretations-Unterstützungs-System. In: Norbert Fuhr, (Hrsg.), *Information Retrieval - GI/GMD-Workshop, Darmstadt*, S. 64–77, Berlin u.a., 1991. Springer-Verlag.

[20] Thomas Muhr. Atlas/ti - A Prototype for the Support of Text Interpretation. In: R. Tesch, (Hrsg.), *Qualitative Sociology, Bd. 14*, S. 349–371, New York, 1991. Human Science Press.

[21] Josef Willenborg. Atlas-PfleSaurus: Ein objektorientiertes System zur Unterstützung der Thesauruspflege. In: Norbert Fuhr, (Hrsg.), *Information Retrieval - GI/GMD-Workshop, Darmstadt*, S. 51–63, Berlin u.a., 1991. Springer-Verlag.

[22] John E. Hopcroft; Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley Publishing Company, Reading, Massachusetts, Menlo Park, California, 1979.

[23] Jürgen Gulbins. *UNIX - Eine Einführung in Begriffe und Kommandos von UNIX - Version 7, bis System V.3. 3. Aufl.* Springer Compass. Hrsg. von M. Nagl; P. Schnupp; H. Strunz. Springer-Verlag, Berlin u.a., 1988.

[24] Alfred Tarski. Der Wahrheitsbegriff in den formalisierten Sprachen. Erstveröffentlichung 1936. In: Karel Berka; Lothar Kreiser, (Hrsg.), *Logik-Texte - Kommentierte Auswahl zur Geschichte der modernen Logik*, Kapitel XII, S. 447–559. Akademie-Verlag, Berlin, 1971.

[25] Z. Pawlak. Mathematical Foundations of Information Retrieval. CC PAS Reports 101, 1973.

[26] Andreas Böhm; Heiner Legewie; Thomas Muhr. *Kursus Textinterpretation: Grounded Theory.* Bericht aus dem Interdisziplinären Forschungsprojekt ATLAS, Nr. 92-3. TU Berlin, Hardenbergstr. 28, 10623 Berlin, 1992.

[27] A. E. Negus, (Hrsg.). *EURONET Guideline: Standard Commands for Retrieval Systems. Final Report.* INSPEC, London, England, December 1977.

[28] B.C. Vickery. *Dokumentationssysteme.* Verlag Dokumentation, München-Pullach, Berlin, 1971.

[29] K. Laisiepen; E. Lutterbeck; K.-H. Meyer-Uhlenried. *Grundlagen der praktischen Information und Dokumentation. Eine Einführung.* Verlag Dokumentation K.G. Saur, München-Pullach, 1972.

[30] A.V. Aho; Y. Sagiv; J.D. Ullman. *Data Structures and Algorithms.* Addison-Wesley Publication, Inc., Reading, Massachusettes, 1983.

[31] Andreas Böhm; Friedrich Braun; Hanna Pishwa. *Offene Interviews - Dokumentation, Transkription und Datenschutz*, 1990.

[32] Raymond M. Smullyan. *Buch ohne Titel - Eine Sammlung von Paradoxa und Lebensrätseln.* Vieweg, Braunschweig & Wiesbaden, 1983.