Zur Syntax und Semantik von Anfragesprachen für Internet-Informationssysteme – Analyse eines logischen Ansatzes

Erhard Konrad

WBS-Bericht 2/2002, TU Berlin, November 2002

Kurzfassung: Eine Arbeit wird analysiert, deren Autor sich zum Ziel gesetzt hat, neue und mächtige Anfragesprachen für Internet-Informationssysteme zu entwerfen und exakt zu spezifizieren. Es wird dargelegt, daß der Autor sein Ziel nicht erreicht hat, insbesondere wird gezeigt, daß seine syntaktisch-semantischen Sprachkonstruktionen größtenteils inkorrekt sind.

1 Einleitung

Josef Willenborg hat eine Arbeit veröffentlicht (Willenborg 2001, Abk. WILL01), in der er die folgende Fragestellung formuliert: „*Es ist ein offenes Problem, einen universellen Zugang zu Internet-Informationssystemen zu schaffen, der die Anfragemächtigkeit der einzelnen Systeme ausreichend berücksichtigt.* “ (S. 1)

Im Zentrum seiner Arbeit steht für ihn die Aufgabe „*... Anfragesprachen mit spezifizierter Syntax und Semantik für Systemklassen (hier: Suche in einfach strukturierten Dokumenten, Suche in link-strukturierten Dokumenten, Suche in geschachtelten Dokumenten) aufzubauen und zu einer universellen Anfragesprache zusammenzuführen.*“ (S. 2)

Willenborg orientiert sich, was den logischen Ansatz angeht, an einem Forschungsbericht über Anfragesprachen für Dokumentennachweissysteme (Konrad & Reiner 1985) und an einer Monographie über Anfragesprachen für Informationssysteme (Reiner 1991b, vgl. auch Reiner 1991a). Er bemüht sich, die dort ausgeführten syntaktisch-semantischen Sprachkonstruktionen auf Anfragesprachen für Systeme mit vernetzten und verweisstrukturierten Dokumenten zu übertragen (Hypertextsysteme, vgl. Kuhlen 1991). Dabei umfassen Dokumente (in WILL01 zitiert) „*Akten, Bilder, Bücher, Briefe und Tonaufzeichnungen ... auf verschiedenen Informationsträgern ...*“ (Ewert & Umstätter 1997, S. 164). Was den Systemtyp betrifft, so beschränkt sich Willenborg auf Dokumentennachweissysteme (Information-Retrieval-Systeme), nicht einbezogen werden Datenbanksysteme und Expertensysteme (wie in Reiner 1991b und Mitschdörfer & Reiner 1991). Er betrachtet Systeme mit ungeordneten Antwortmengen, dagegen bleiben Anordnungssysteme (Systeme mit Ähnlichkeitsmaßen) unberücksichtigt (vgl. hierzu Bollmann & Konrad 1979, Salton & McGill 1983 und Konrad & Reiner 1986).

Exkurs: Bei der Konstruktion logischer Anfragesprachen (vgl. z.B. Reiner 1991b) werden Syntax und Semantik induktiv (rekursiv) definiert. Dadurch wird erreicht, daß die Eigenschaft einer Zeichenreihe, eine Anfrage zu sein, algorithmisch entscheidbar ist. Außerdem kann sichergestellt werden, daß jede Anfrage eine Bedeutung bekommt. Bedeutungen werden extensional gemäß einer mengentheoretischen Semantik erklärt (wie in der mathematischen Logik üblich).

Im vorliegenden Bericht wird untersucht, inwieweit es Willenborg gelungen ist, „*Anfragesprachen mit spezifizierter Syntax und Semantik ... aufzubauen ...*“ (WILL01, S. 2), und insbesondere, ob er Syntax und Semantik der Anfragesprachen korrekt definiert hat.

2 Zu den mathematischen Grundlagen

2.1 Gerichtete Graphen

In Kap. 2.3 (WILL01) stellt Willenborg im Anschluß an die englischsprachige Monographie von G. Schmidt und T. Ströhlein (1993) Definitionen aus der Graphentheorie zusammen. Grundlegend ist der Begriff des gerichteten Graphen (als Modell eines Informationsnetzes). Ein gerichteter Graph ist definiert als Paar, bestehend aus einer Menge von Knoten (Punkten) und einer assoziierten Relation (Abk.: „C“ statt „Conn“, denn „Conn“ wird in WILL01 beim Aufbau der Anfragesprachen auch als Prädikatenkonstante benutzt):

$$G = (N, C) \text{ mit } C \subset N \times N \text{ (kartesisches Produkt).}$$

Vorbereitend für die Konstruktion von Anfragesprachen wird die Menge der Geschwister eines Knotens x wie folgt zu definiert (Def. 11, S. 15):

$$\textit{Menge der Geschwister von } x =_{Df} C^T Cx - \{x\}$$

Dabei ist Cx die Menge der (direkten) Vorgänger des Knotens x und $C^T x$ die Menge der (direkten) Nachfolger von x. C^T ist die Transponierte von C, d.h. $C^T = \{\langle x, y\rangle : \langle y, x\rangle \in C\}$.
In Worten:

> *Die Menge der Geschwister eines Knotens x ist die Menge der direkten Nachfolger der direkten Vorgänger von x ohne x selbst.*

Wie das folgende Beispiel zeigt, ist die Definition inkorrekt:

Sei $N = \{a, b, c, d, e\}$ und $C = \{\langle a, c\rangle, \langle a, d\rangle, \langle b, d\rangle, \langle b, e\rangle\}$. Dann erhält man als Geschwister des Knotens d: $C^T Cd - \{d\} = \{c, e\}$. Die Knoten c und e haben jedoch keinen gemeinsamen (direkten) Vorgänger (vgl. Schmidt & Ströhlein 1989, S. 27: „*... Geschwister ..., also ... Punkte, die einen gemeinsamen* [direkten] *Vorgänger besitzen*“). Notwendig und hinreichend dafür, daß zwei Knoten x und y einen gemeinsamen (direkten) Vorgänger haben, ist die Bedingung

$$xy^T \subset C^T C$$

(vgl. Schmidt & Ströhlein 1989, S. 27 bzw. 1993, S. 26). Diese Bedingung ist jedoch für x = c und y = e verletzt, da $ce^T = \{\langle c, e\rangle\}$ und
$C^T C = \{\langle c, c\rangle, \langle c, d\rangle, \langle d, c\rangle, \langle d, d\rangle, \langle d, e\rangle, \langle e, d\rangle, \langle e, e\rangle\}$ gilt.

Ähnliche Beispiele können in beliebiger Zahl konstruiert werden.

Die Definition der Geschwister eines Knotens stellt Willenborg für die Konstruktion der Anfragesprachen LinkQL, Nested&LinkQl und StructuredQL bereit (WILL01, Kap. 4.3, 4.4 und 4.5). Dort fehlt jedoch ein Rückbezug.

2.2 Logiksprachen

Als Hilfsmittel zur Konstruktion von Anfragesprachen stellt Willenborg in Kap. 2.4 (WILL01) formale Sprachen der klassischen Logik dar. Über die Standard-Sprache der Prädikatenlogik (Quantorenlogik) erster Stufe hinaus versucht er, Sprachen der mehrsortigen Logik und der Stufenlogik in geeigneter Form zu adaptieren (in Anlehnung an Carnap 1958 und Hilbert & Ackermann 1972).

Für die Sprache der Logik erster Stufe mit n verschiedenen Objektsorten s_1, s_2, ..., s_n (Kap. 2.42) ist die Definition der Prädikaten- und Funktionssymbole lückenhaft, denn es werden nur Prädikatensymbole der Sorte $(s_1, s_2, ..., s_n)$ und Funktionssymbole der Sorte $(s_1, s_2, ..., is_n : s)$ berücksichtigt. Es fehlen Symbole der Sorten $(s_{i1}, s_{i2}, ..., s_{im})$ bzw. $(s_{i1}, s_{i2}, ..., s_{im} : s)$, wobei s_{i1}, s_{i2}, ..., s_{im} eine beliebige Folge von Sorten s_1, s_2, ..., s_n ist.

Die Sprachen der Stufenlogik (WILL01, Kap. 2.4.3 und 2.4.4) sind in Anlehnung an die Versionen in Carnap (1958) und Hilbert & Ackermann (1972) aufgebaut. Die Synthese beider Versionen ist jedoch fehlerhaft: Bei der induktiven Definition der Typen bleibt der Typ der Funktionssymbole undefiniert (S. 19 und 21), atomare Formeln werden nicht erklärt (S. 20 und 22) und (unsinnige) λ-Ausdrücke mit Prädikatenkonstanten als gebundene Variablen gebildet (S. 20 und 22). Für korrekt gebildete λ-Ausdrücke mit Prädikatenvariablen wird eine Pseudo-Interpretation angegeben (S. 20 und 23), da die Relation rel nicht erklärt ist.

Es erhebt sich die Frage, warum Willenborg Sprachen der allgemeinen Stufenlogik (Typenlogik) vorstellt, obwohl er für den Aufbau von Anfragesprachen nur Logiksprachen zweiter Stufe benutzt (vgl. WILL01, Kap. 4.4 und 4.5). Eine Sprache erster Stufe läßt sich zu einer Sprache zweiter Stufe erweitern, indem Quantoren und λ-Operatoren mit gebundenen Prädikatenvariablen sowie Prädikatenprädikate in den Aufbau einbezogen werden (wie in den genannten Werken von Carnap und Hilbert & Ackermann ausgeführt, vgl. auch Hermes 1972).

Exkurs:

Bei der Definition der Semantik von Logiksprachen spielt der Begriff der logischen Äquivalenz von Formeln eine wichtige Rolle (vgl. Carnap 1958, S. 21 bzw. 1973, S. 21).

Def.: Zwei Formeln F_1 und F_2 sind genau dann logisch äquivalent, wenn jede Interpretation $\mathfrak{S}$, die ein Modell von F_1 ist, auch ein Modell von F_2 ist und umgekehrt. Symbolisch:

$$F_1 \text{ äq } F_2 \text{ gdw}_{Df} \text{ Mod } \mathfrak{S} \; F_1 \text{ gdw Mod } \mathfrak{S} \; F_2$$

Mit dem Definiens gleichbedeutend sind die beiden folgenden Aussagen:

a) Aus F_1 folgt logisch F_2 und umgekehrt. Symbolisch: $F_1 \models F_2$ und $F_2 \models F_1$.
b) $F_1 \leftrightarrow F_2$ ist logisch gültig (allgemeingültig, tautologisch).

Die logische Äquivalenz von Formeln ist bei der Definition der Semantik von Anfragesprachen zu beachten. Im Folgenden wird gezeigt, daß Willenborg beim

Aufbau der Anfragesprachen LinkQl, Nested&LinkQL und StructuredQL die logische Äquivalenz verletzt.

2.3 Anfragesprachen

In Kap. 2.5 (WILL01) erörtert Willenborg die Mächtigkeit (Ausdrucksstärke) von Anfragesprachen. Dazu heißt es (S. 24):

> „*Man kann allgemein sagen, daß eine Anfragesprache A mächtiger als eine Anfragesprache B ist, wenn A den Dokumentenbestand schärfer trennt als B.*“

Das Prädikat „trennt den Dokumentenbestand schärfer als “ wird nicht definiert. Was mit „mächtiger“ gemeint ist, bleibt daher offen.

Es folgt eine zweite Erklärung zur Mächtigkeit (S. 24):

> „*Eine Anfragesprache A ist mächtiger als eine Anfragesprache B, wenn die Menge der in A bildbaren Suchergebnismengen die Menge der in B bildbaren Suchergebnismengen umfaßt. (Adaption aus Konrad(1986) S. 566)*“

Die angegebene Literaturstelle existiert nicht. In der Erklärung des Prädikates „mächtiger“ wird ein syntaktischer Begriff mit einem semantischen Begriff vertauscht. Statt „bildbar“ muß es richtig „beschreibbar“ heißen. Anfragen beschreiben (Suchergebnis-)Mengen. Anfragen sind sprachliche Objekte, Mengen dagegen nichtsprachliche Objekte. Eine korrekte Definition der Mächtigkeit von Anfragesprachen findet sich bei Reiner 1991b, S. 18.

Über den Zusammenhang zwischen der Mächtigkeit einer Anfragesprache und ihrer Eignung für den praktischen Einsatz führt Willenborg aus (S. 25):

> „*Die mächtigste denkbare Anfragesprache könnte die Potenzmenge des Dokumentenbestands (Menge aller Teilmengen des Dokumentenbestands) bilden. Diese Sprache ist jedoch im praktischen Einsatz ungeeignet, da sie für die unterschiedlichen Benutzeranfragesprachen zu mächtige Sprachmittel bereitstellt. Systemsprachen müßten diese unnötige Mächtigkeit mit dem Preis einer geringeren Systemperformanz im Informationssystem nachbilden.*“

Die Argumentation ist in dieser Allgemeinheit nicht stichhaltig. Die Laufzeit zur Beantwortung von Anfragen hängt auch davon ab, ob und wie die ursprüngliche Anfrage optimiert wird, d.h. semantisch äquivalent umgeformt wird, bevor sie dem (Anfrage-)Interpreter übergeben wird (vgl. hierzu Ullman 1982, S. 268 ff.).

Um den Begriff der Mächtigkeit zu beleuchten, soll ein einfaches Boolesches Dokumentennachweissystem betrachtet werden (zur allgemeinen Definition vgl. Bollmann & Konrad 1979, Salton & McGill 1983 und Reiner 1991b):

Seien $\Delta = \{D_1, D_2, D_3, D_4\}$ ein Dokumentenbestand und $\Sigma = \{a, b, c\}$ eine Menge von Deskriptoren, mit denen Anfragen der Form $\alpha_1 \wedge \alpha_2$ (Konjunktion), $\alpha_1 \vee \alpha_2$ (Disjunktion) und $\neg\alpha$ (Negation) gebildet werden können (Anfragesprache L). Die Dokumente von Δ seien wie folgt indexiert: D_1 mit a, b und c; D_2 mit a und b; D_3 mit a und c; D_4 mit a.

Die Anfragesprache L hat maximale Mächtigkeit, denn alle Teilmengen von Δ (alle Elemente der Potenzmenge von Δ) können durch Anfragen beschrieben werden. Dabei ist eine Teilmenge M von Δ in der Sprache L beschreibbar, wenn es in L einen Ausdruck α (Anfrage) gibt, so daß $\Im(\alpha) = M$ für eine Interpretation $\Im$ von L gilt.
Wird entweder auf die Konjunktion oder die Disjunktion verzichtet, so ändert sich die Mächtigkeit der Anfragesprache L nicht. Ohne die Negation können jedoch die Dokumentenmengen $\{D_1, D_4\}$ und $\{D_2, D_3\}$ in L nicht beschrieben werden.
Wird z.B. der Deskriptor c entfernt, so können die Dokumentenmengen $\{D_1, D_3\}$ und $\{D_2, D_4\}$ in L nicht beschrieben werden.

Das Beispiel zeigt, daß die Mächtigkeit einer Anfragesprache von ihrer logischen Struktur und ihrem Vokabular abhängt. Wird einer der beiden Faktoren eingeschränkt, so kann es vorkommen, daß manche Dokumentenmengen mit den Mitteln der Anfragesprache nicht auffindbar sind.

In diesen Kontext gehört die „These 2“, die unter „Erzielte Ergebnisse“ (WILL01, Kap. 8, S. 172) zu finden ist:

> „*Wenn zur Bestimmung eines Suchergebnisses die Anfrage an unterschiedlich mächtige Informationssysteme gesendet wird, kann in diesen keine Vereinigungsmenge, sondern nur die Schnittmenge des Suchergebnisses gebildet werden.*“

Willenborgs These ist unsinnig. Vereinigung und Durchschnitt (Schnittmenge) aller möglichen Suchergebnisse (Antworten) mehrerer Systeme können stets gebildet werden. Die Antwort zu einer Anfrage liegt in jedem Fall in der Vereiningung, aber nicht im Durchschnitt der Mengen möglicher Antworten der Systeme, da dieser leer sein kann. Möglich ist auch, daß die Antwort sowohl Element der Vereinigung als auch des Durchschnitts der Antwortmengen ist. Dies soll präzisiert werden:

Seien $S_1 = (\Delta, L_1)$ und $S_2 = (\Delta, L_2)$ zwei Boolesche Dokumentennachweissysteme mit demselben Dokumentenbestand Δ . Seien ferner L_1 und L_2 Anfragesprachen unterschiedlicher Mächtigkeit und $\mathcal{B}(L_1)$ und $\mathcal{B}(L_2)$ die Mengen der durch L_1 bzw. L_2 beschreibbaren Teilmengen von Δ (d.h. $\mathcal{B}(L_1)$ und $\mathcal{B}(L_2)$ sind Teilmengen der Potenzmenge von Δ). Dann liegt die Antwort $\Im(\alpha)$ (Teilmenge von Δ) einer Anfrage α, die in L_1 und L_2 formulierbar ist, im Durchschnitt der beschreibbaren Mengen von S_1 und S_2 (falls dieser nicht leer ist):

$$\Im(\alpha) \in \mathcal{B}(L_1) \cap \mathcal{B}(L_2).$$

Wenn nun auch $\mathcal{B}(L_1) \subset \mathcal{B}(L_2)$ oder $\mathcal{B}(L_2) \subset \mathcal{B}(L_1)$, so

$$\Im(\alpha) \in \mathcal{B}(L_1) \cup \mathcal{B}(L_2).$$

D.h. die Antwort auf die Anfrage α liegt sowohl im Durchschnitt als auch in der Vereinigung der beschreibbaren Mengen, wenn eine Anfragesprache mächtiger ist als die andere. Nur wenn die Anfragesprachen nicht vergleichbar sind, d.h. $\mathcal{B}(L_1)$ und $\mathcal{B}(L_2)$ in keiner Teilmengenbeziehung zueinander stehen, liegt die Antwort ausschließlich im Durchschnitt $\mathcal{B}(L_1) \cap \mathcal{B}(L_2)$, falls diese Menge nicht leer ist. Wenn aber $\mathcal{B}(L_1) \cap \mathcal{B}(L_2)$ leer ist, so liegt die Antwort in der Vereinigungsmenge $\mathcal{B}(L_1) \cup \mathcal{B}(L_2)$, d.h. in $\mathcal{B}(L_1)$ oder $\mathcal{B}(L_2)$.

Exkurs: Auch bei Anordnungssystemen, die mit Ähnlichkeitsmaßen arbeiten, ist ein Mächtigkeitsvergleich möglich. Zu einem Beweis für die Gleichmächtigkeit von vektoriellen Systemen mit dem Skalarprodukt als Ähnlichkeitsmaß und dem Ähnlichkeitsmaß von Tanimoto vgl. Konrad (1982).

3 Zu den aufgebauten Anfragesprachen

Zur Zielsetzung führt Willenborg aus (WILL01, S. 110):

„*Die aufgebauten Anfragesprachen sind nicht für den Benutzer, sondern als Zwischensprachen zwischen Benutzeranfragesprachen und den Informationssystemen konzipiert. Nur ein Rechercheexperte benutzt die Anfragesprachen direkt. (...) Für jede Anfragesprache wird deren Syntax und Semantik exakt spezifiziert.*"

Willenborg konstruiert vier Anfragesprachen, die von ihm dann „*zu einer universellen Anfragesprache zur Suche in strukturierten Dokumenten (Structured-QL) zusammengeführt*" (S. 110) werden. Den Terminus „universell" präzisiert er nicht.

Für die Sprachkonstruktionen wird vorausgesetzt: „*In den Anfragesprachen werden 2-stellige Prädikate verwendet.*" (S. 110) Diese Voraussetzung wird nicht strikt beachtet, denn in vier der fünf vorgestellten Sprachen werden n-stellige Prädikate (n ≥ 3) verwendet, ohne ihre Reduktion auf 2-stellige Prädikate durchzuführen.

3.1 SimpleStructuredQL

Diese Anfragesprache ist bestimmt für „*Benutzer, die einen einheitlichen Zugriff auf einfach strukturierte Internet-Dokumente wünschen*" (WILL01, S. 110). Es handelt sich um eine mehrsortige Logiksprache mit 2-stelligen Prädikatensymbolen und Quantoren für Individuenvariablen. Anfragen werden als λ-Ausdrücke mit gebundenen Individuen- und Prädikatenvariablen definiert.

Die Anfragesprache SimpleStructuredQL hat eine Reihe von Mängeln:

a) Individuenkonstanten werden mit Sorten identifiziert (S. 111).

b) Der Ausdruck $(\lambda P^{s1,\ldots,sn})F$ ist für $n > 2$ nicht erklärt, da im Alphabet der Anfragesprache nur Prädikatenvariaben der Sorten (s_{i1}, s_{i2}) vorgesehen sind (S. 112). Bei der Interpretation $\Im$ des λ-Ausdrucks (S. 114) wird die Relation rel nicht definiert.

c) Im Widerspruch zur Voraussetzung der Zweistelligkeit wird die 3-stellige Prädikatenkonstante Near der Sorte (string, int, string) eingeführt (S. 112). Es wird nicht gezeigt, wie diese Prädikatenkonstante bedeutungsgleich durch eine 2-stellige Prädikatenkonstante ersetzt werden könnte.

d) Die 2-stelligen Prädikatenkonstanten $\neq$ (ungleich), $<$ (kleiner), $>$ (größer), $\leq$ (kleiner-gleich) und $\geq$ (größer-gleich) werden für die Sorten (int, int) und (date, date) eingeführt. Beim Formelaufbau werden jedoch nur Prädikatenkonstanten der Sorte (s1, s2) mit s1 $\neq$ s2 berücksichtigt (S. 112), und bei der Definition der Interpretation $\Im$ werden sie fehlerhaft mit der Sorte (s1, s2) indiziert (S. 113).

e) Für die folgenden 2-stelligen Prädikatenkonstanten der Sorte (string, string) fehlt die Definition einer extensionalen Interpretation (vgl. Carnap 1958 S. 42 ff. bzw. 1954 S. 42 ff.): Upper, Lower, Stem, Fuzzy, Soundex, Syn, NT, BT und PT (S. 113). Sie werden beim Aufbau der Syntax lediglich als Notation für die folgenden Prädikate benutzt: „in Großbuchstaben", „in Kleinbuchstaben", „Stammform ", „rechtschreibähnlich", „phonetisch ähnlich", „synonym", „Unterbegriff", „Oberbegriff" und „Vorzugsbegriff".

f) Es fehlt eine Verbindung von Thesaurus (Wörterbuch) und Dokumentenbestand. Daher kann nur getrennt in einem Thesaurus und einem Dokumentenbestand gesucht werden. Terminologiebasiertes Retrieval wird zwar diskutiert (Kap. 6), aber beim Sprachaufbau ausgelassen (Lücke gegenüber Reiner 1991b).

3.2 SimpleLinkQL

Diese Anfragesprache soll „*einen Zugriff auf einfache link-strukturierte Dokumente*" ermöglichen (WILL01, S. 119). Im selben Absatz heißt es: „*Dokumente werden als einfach (unstrukturiert) vorausgesetzt.*" Gemeint ist wohl der Zugriff auf einen link-strukturierten Dokumentenbestand. Und weiter: „*Beziehungen sind gerichtet.*". Richtig ist, daß Beziehungen nicht notwendig symmetrisch sind und durch gerichtete Graphen repräsentiert werden können.

Bei der Anfragesprache handelt es sich um eine einfache Logiksprache mit einer 2-stelligen Prädikatenkonstanten und Quantoren für Individuenvariablen. Anfragen werden als λ-Ausdrücke mit gebundenen Individuenvariablen definiert.

Es wird vorausgesetzt, daß der Dokumentenbestand die Struktur eines gerichteten Graphen hat. Schlingen, d.h. Wege, die aus einer einzigen Kante mit identischem Anfangs- und Endknoten bestehen, sind zugelassen. Sie repräsentieren Dokumente mit einem Verweis (link) auf sich selbst (dessen Sinn nicht erklärt wird).

Der Aufbau der Anfragesprache SimpleLinkQL ist im wesentlichen korrekt, ihre Anwendung auf Beispiele enthält jedoch Unstimmigkeiten:

Bei den fünf Beispielfragen (Kap. 4.2.3) entspricht dem natürlichsprachlichen Satz

Das Dokument $unit_2$ steht zum Dokument $unit_1$ in einer Beziehung.

(irreführend) die Formel

$$\mathrm{Conn}(\mathrm{unit}_1, \mathrm{unit}_2).$$

Nach Hilbert & Ackermann 1972 (3. Kap.) müßte die Formel lauten:

$$\mathrm{Conn}(\mathrm{unit}_2, \mathrm{unit}_1).$$

Bei Frage 1 wird nach allen Dokumenten gefragt, zu denen (vgl. Abb. 36) das Dokument UNIT_1 nicht in Beziehung steht. Wird UNIT_1 durch UNIT_9 ersetzt, so ist das Dokument selbst die Antwort. Es fehlt eine Erklärung, warum das Dokument UNIT_9 als einziges Dokument mit sich selbst in Beziehung steht.

Bei Frage 3 wird nach allen Dokumenten gefragt, zu denen ein benanntes Dokument nicht in Beziehung steht. Bei großen Dokumentenbeständen, wie sie im

Internet vorkommen, dürften Fragen mit der (absoluten) Negation Effizienzprobleme verursachen. Hier hätte sich eine Erörterung mit Bezug auf Kap. 2 (Abschnitt „Anfragesprachen“) angeboten.

Bei Frage 5 wird nach Dokumentenpaaren gefragt, die miteinander in Beziehung stehen, sofern sie über ein Dokument (als Zwischenstation) miteinander in Beziehung stehen (lokale Transitivität). Die Auswertung ist fehlerhaft, da nur das Paar $\langle$ UNIT_1, $\mathrm{UNIT}_7\rangle$ als Antwort geliefert wird, es fehlt das Paar $\langle \mathrm{UNIT}_6$, $\mathrm{UNIT}_9\rangle$.

3.3 LinkQL

Diese Anfragesprache kann als Erweiterung der Sprache SimpleLinkQL angesehen werden. Sie ist konzipiert für „*Benutzer, die einen Zugriff auf link-strukturierte Dokumente wünschen*“ (WILL01, S. 122). Gemeint ist ein link-strukturierter Dokumentenbestand. Es werden Prädikate eingeführt, die in einem Dokumentenbestand, der die Struktur eines gerichteten Graphen hat, folgende Anfragen ermöglichen sollen: Anfragen

- nach Pfaden (Wegen) bestimmter Länge zwischen zwei benannten Dokumenten
- nach Nachfolgern einer bestimmter Stufe zu einem benannten Dokument
- nach Nachfolgern bis zu einer bestimmten Stufe zu einem benannten Dokument
- nach Vorgängern einer bestimmten Stufe zu einem benannten Dokument
- nach Vorgängern bis zu einer bestimmten Stufe zu einem benannten Dokument
- nach Geschwistern eines benannten Dokuments
- nach Dokumenten auf einem Pfad (Stationen) zwischen zwei benannten Dokumenten.

Die Anfragesprache LinkQL hat eine Reihe vom Mängeln:

a) Beim Aufbau der Syntax (S. 122) werden Individuensymbole für Stufen von Nachfolgern bzw. Vorgängern eingeführt, ohne sie als Argumente in atomaren Formeln oder bei der Interpretation $\Im$ der nichtlogischen Symbole einzubeziehen.

b) Die Prädikatenkonstanten Path (Pfad), $\downarrow$ (Nachfolger einer Sufe), $\Downarrow$ (Nachfolger bis zu einer Stufe), $\uparrow$ (Vorgänger einer Stufe), $\Uparrow$ (Vorgänger bis zu einer Stufe), $\Leftrightarrow$ (Geschwister) und θ (Stationen) werden bei der Interpretation $\Im$ der nichtlogischen Symbole nicht berücksichtigt, sondern es wird versucht, ihre Interpretation mit Hilfe der Prädikatenkonstanten Conn durch logische Äquivalenzen zu bewerkstelligen (S. 123). –
Zur Interpretation $\Im$ der n-stelligen Prädikatenkonstanten Path wird die folgende Aussage benutzt:

Mod $\Im$ $\text{Path}^{Conn}(x_1, ..., x_n)$ gdw_{Df}

Mod $\Im$ Conn(x_1, x_2) und Mod $\Im$ Conn(x_2, x_3) und ... und
Mod $\Im$ Conn(x_{n-1}, x_n) und $\Im(x_1) \neq \Im(x_2) \neq ... \neq \Im(x_n)$. $(\star)$

Die Aussage $(\star)$ ist als Definition unvollständig, sie könnte zu einer induktiven Definition nach der Zahl der Knoten (Kanten) ergänzt werden. Schwerwiegender ist, daß die Aussage $(\star)$ bei beliebigen Interpretationen $\Im$ keine Äquivalenz darstellt. Weiterhin werden durch die Bedingung $\Im(x_1) \neq \Im(x_n)$ Zyklen ausgeschlossen, im Widerspruch zur Voraussetzung „*Die Bestände können Zyklen enthalten.*"(S. 122)

c) Die Unstimmigkeiten bei der Definition der Interpretation für die Prädikatenkonstante Path kommen als Folgefehler bei der Interpretation der Prädikatenkonstanten für „Nachfolger", „Vorgänger"und „Stationen" zum Tragen, da diese mit Hilfe der Prädikatenkonstanten Path erklärt werden. Ein weiterer Definitionsfehler besteht darin, daß im Definiens Ausdrücke der Form $\text{Path}^{Conn}(x_\alpha, ..., x_i, ... x_\beta)$ auftreten, wobei die freien Variablen x_i, i=1, ..., s-1 (bzw. n) nicht im Definiendum vorkommen (vgl. Carnap 1958, S. 62 ff. bzw. 1973, S. 54 ff.).

d) Bei der Definition des Prädikates „Geschwister" wird zugelassen, daß Geschwisterknoten mehrere Vaterknoten haben können. Weiterhin liegt ein Definitionsfehler bei der Interpretation $\Im$ der Prädikatenkonstanten θ vor, da die freie Variable x_γ zwar im Definiens aber nicht im Definiendum vorkommt.

e) Bei den Beispielfragen (S. 124-128), deren Auswertung wegen ungültiger Äquivalenzen irrig ist, wird die Länge eines Pfades der Zahl der Zwischenknoten gleichgesetzt. Die Länge eines Pfades ist jedoch bei gerichteten Graphen definiert als Zahl der Kanten zwischen Anfangs- und Endknoten (vgl. Aho et al. S. 198) bzw. als Knotenzahl minus 1 (Schmidt & Ströhlein 1993, S. 105).

f) Der in Kap. 2.3 angegebene graphentheoretische Rahmen wird überschritten, weil Graphen mit mehreren assoziierten Relationen (den Prädikatenkonstanten $Conn_i$ entsprechend) zugelassen werden.

3.4 Nested&LinkQL

Diese Anfragesprache soll „*einen Zugriff auf geschachtelte und link-strukturierte Dokumente*" ermöglichen. „*Geschachtelte Dokumente werden mit den Funktionen aufgebaut (vgl. Abiteboul, Beeri (1995))*" (WILL01, S. 128). Die Arbeit von Abiteboul und Beeri fehlt im Literaturverzeichnis. Für die Schachtelung in Dokumenten und die Verknüpfung von Teildokumenten werden gerichtete azyklische Graphen als Struktur zugrundegelegt (S. 131).
Es wird eine mehrsortige Logiksprache mit n-stelligen Funktionssymbolen, ein- und zweistelligen Prädikatensymbolen sowie Prädikatenprädikaten (für Eigenschaften von Relationen) aufgebaut. Eine Besonderheit sind Prädikate für „*Teile, Behälter und Geschwister von Dokumenten*". Anfragen werden als λ-Ausdrücke mit gebundenen Individuen- und Prädikatenvariablen definiert.

Die Anfragesprache Nested&LinkQL hat eine Reihe von Mängeln:

a) Es werden Individuensymbole für Stufen von Teilen und für Stufen von Behältern eingeführt. Die Symbole treten weder als Argumente von Prädikatensymbolen auf noch werden sie bei der Interpretation $\Im$ berücksichtigt.

b) Die Interpretation $\Im$ der Formeln $\bigtriangledown_m$(t,x) (Teile von t der Stufe m), $\bigtriangleup_m$(t,x) (Behälter von t der Stufe m) und $\Diamond$(t,x) (Geschwister von t) ist jeweils mehrfach fehlerhaft. Die Ausdrücke werden wie atomare Formeln behandelt, obwohl sie als logisch äquivalent zu Formeln mit Quantoren eingeführt werden. Außerdem sind Ausdrücke mit Quantoren für Funktionsvariablen in der Syntax der Sprache gar nicht definiert. –
Bei der Interpretation $\Im$ der nichtlogischen Symbole werden die Prädikatenkonstenten $\bigtriangledown$ (Teile einer Stufe), $s\bigtriangledown$ (Teile bis zu einer Stufe), $\bigtriangleup$ (Behälter einer Stufe), $s\bigtriangleup$ (Behälter bis einer Stufe) und $\Diamond$ (Geschwister) ausgelassen (Symbolik: $s\bigtriangledown$ und $s\bigtriangleup$ anstelle von schwarzen Dreiecken in WILL01). Stattdessen wird versucht, ihre Interpretation $\Im$ durch logische Äquivalenzen mit Hilfe von Funktionen zu definieren. –
Zur Interpretation $\Im$ der 2-stelligen Prädikatenkonstanten $\bigtriangledown_m$ werden die folgenden Aussagen benutzt (Versuch einer induktiven Definition):

Mod $\Im$ $\bigtriangledown_m$(t, x) gdw$_{Df}$
Mod $\Im$ $(\exists f)(\exists x_1)...(\exists x_n)$ $\bigtriangledown_{m-1}$(t, f(x_1, ..., x, ..., x_n))

Mod $\Im$ $\bigtriangledown_1$(t, x) gdw$_{Df}$
Mod $\Im$ $(\exists f)(\exists x_1)...(\exists x_n)$ Equal(t, f(x_1, ..., x, ..., x_n))

Die Definition der Interpretation $\Im$ von $\bigtriangledown_m$ ist irrig (gilt mit gleicher Begründung für $s\bigtriangledown_m$, $\bigtriangleup_m$, $s\bigtriangleup_m$ und $\Diamond$): Zum einen sind beliebige Terme im Definiendum unzulässig (vgl. Carnap 1958, S. 62 ff. bzw. 1973, S. 56 ff.). Z.B. könnte der Term t mit der Individuenvariablen x identisch sein oder (undefinierte) Funktionssymbole enthalten. Zum anderen ist zur Einführung der Funktionsvariablen im Definiens ein Existenz- und Eindeutigkeitsbeweis für Funktionen erforderlich (vgl. Carnap 1958, S. 71 ff. bzw. 1973, S. 71 ff.). Ein solcher Beweis kann bei gerichteten azyklischen Graphen G = (N, C) nicht gelingen, da weder die assoziierte Relation C noch ihre Transponierte C^T i.a. Funktionen sind, d.h. es gilt weder $C^TC \subset I$ noch $CC^T \subset I$ mit I als Identitätsrelation (vgl. Schmidt & Ströhlein 1993, S. 54 bzw. 1989, S. 56/57).

c) Die Beispielfragen (S. 133-138) beziehen sich auf einen Spezialfall mit umkehrbar eindeutigen Abbildungen und sind daher nur bedingt relevant. Die Relation CONN wird als symmetrisch vorausgesetzt (S. 135), d.h. es gilt CONN^T = CONN. Ein Graph mit der assoziierten Relation CONN hat jedoch Zyklen – im Widerspruch zur Voraussetzung der Zyklenfreiheit (S. 131).

3.5 StructuredQL

Diese Anfragesprache soll „*eine universelle Suche in strukturierten Dokumenten*“ (WILL01, S. 138) ermöglichen. Die bislang vorgestellten vier Anfragesprachen sollen zu einer einzigen Anfragesprache vereinigt werden. Das Prädikat „universell“ wird nicht expliziert, so daß die Zielsetzung vage bleibt.

Es wird eine mehrsortige Logiksprache mit n-stelligen Funktions- und Prädikatensymbolen, Prädikatenprädikaten und Quantoren für Individuen-, Funktions- und Prädikatenvariablen aufgebaut. Anfragen werden als λ-Ausdrücke mit gebundenen Individuen- und Prädikatenvariablen definiert.

Die Anfragesprache StructuredQL hat eine Vielzahl von Mängeln. Zu den Mängeln der vier Einzelsprachen treten Mängel hinzu, die durch ihre Zusammenführung entstehen. Im einzelnen:

a) Indizes für Stufen werden als Individuensymbole eingeführt, die weder als Argumente von Funktions- noch von Prädikatensymbolen auftreten und auch nicht interpretiert werden.

b) Im Widerspruch zur allgemeinen Voraussetzung der Zweistelligkeit (S. 110) werden 3-stellige Prädikatenkonstante (Near und Θ) und eine n-stellige Prädikatenkonstante (Path) verwendet. Diese Prädikatenkonstanten können nicht als Einsetzungsinstanzen von 2-stelligen Prädikatenvariablen auftreten (nur solche sind zugelassen).

c) Für die 2-stelligen Prädikatenkonstanten Equal, $\neq, <, >, \leq$ und $\geq$ werden zum einen die Sorten (s_1, s_2) zugelassen (was unsinnig ist), und zum anderen fehlt eine extensionale Interpretation.

d) Für die folgenden 2-stelligen Prädikatenkonstanten der Sorte (string, string) fehlt eine extensionale Interpretation : Upper, Lower, Stem, Fuzzy, Soundex, Syn, NT, BT und PT.

e) Die Prädikatenkonstanten Path (Pfad), $\downarrow$ (Nachfolger einer Sufe), $\Downarrow$ (Nachfolger bis zu einer Stufe), $\uparrow$ (Vorgänger einer Stufe), $\Uparrow$ (Vorgänger bis zu einer Stufe), $\Leftrightarrow$ (Geschwister) und θ (Stationen) werden bei der Interpretation $\Im$ der nichtlogischen Symbole nicht berücksichtigt, sondern es wird versucht, sie mit Hilfe der Prädikatenkonstanten Conn durch logische Äquivalenzen zu definieren. Bei der Definition der Prädikatenkonstanten Path sind Definiendum und Definiens jedoch nicht äquivalent, weil die Interpretation $\Im$ der Individuensymbole x_i (i=1, ..., n) im Definiens eingeschränkt wird. Die Prädikatenkonstanten $\downarrow, \Downarrow, \uparrow, \Uparrow, \Leftrightarrow$ und θ werden mit Hilfe der Prädikatenkonstanten Path definiert. Dabei werden im Definiens freie Individuenvariablen benutzt, die im Definiendum nicht vorkommen (Definitionsfehler s.o.).

f) Die Prädikatenkonstanten $\bigtriangledown$ (Teile einer Stufe), $s\bigtriangledown$ (Teile bis zu einer Stufe), $\bigtriangleup$ (Behälter einer Stufe), $s\bigtriangleup$ (Behälter bis zu einer Stufe) und $\diamond$ (Geschwister) werden bei der Interpretation $\Im$ der nichtlogischen Symbole nicht berücksichtigt, sondern es wird versucht, sie durch logische Äquivalenzen zu definieren. Im Definiendum werden beliebige Terme zugelassen, im Definiens kommen Formeln mit gebundenen Funktionsvariablen vor, die in der Syntax nicht erklärt sind, und bei der Einführung der Funktionsvariablen fehlt ein Existenz- und Eindeutigkeitsbeweis für Funktionen (Definitionsfehler s.o.).

g) Die Ausdrücke $(\exists f)F$ und $(\forall f)F$ fehlen bei der Definition der Formeln und werden zudem mit Hilfe einer undefinierten Menge $\{\rightarrow\}$ interpretiert (Pseudo-Interpretation).

h) Bei der Interpretation $\Im$ der Formel Prop($P^{s1,s2}$) bleibt $\Im$(Prop) undefiniert.

i) Die graphentheoretischen Voraussetzungen zum Aufbau der Anfragesprachen SimpleStructuredQL, SimpleLinkQL, LinkQL und Nested&LinkQL sind in ihrer Gesamtheit miteinander unverträglich (zyklische vs. azyklische Graphen). Dies führt zu einer widerspruchsvollen Semantik der Anfragesprache StructuredQL.

Unter „Erzielte Ergebnisse" (Kap. 8, S.172) nennt Willenborg als „These 1":

> „*Man kann 3 Gruppen von strukturierten Dokumenten unterscheiden:*
> *a) Einfach strukturierte Dokumente*
> *b) Link-strukturierte Dokumente*
> *c) Geschachtelte Dokumente*
> *Es kann eine Anfragesprache entwickelt werden, mit der es möglich ist, alle 3 Gruppen universell anzufragen und deren jeweilige Vorteile zur verbesserten Recherche zu nutzen.*"

Willenborg präzisiert nicht, was er mit „universell" meint. Seine „universelle Anfragesprache StructuredQL" (S. 167) ist, wie oben dargelegt, eine Zusammenführung syntaktischer und semantischer Fehlkonstruktionen.

4 Schlußbemerkungen

Willenborg hat seine Untersuchung über Anfragesprachen für Internet-Informationssysteme in einem frühen Stadium veröffentlicht. Die Darstellung der mathematischen Grundlagen ist fehler- und lückenhaft, sie enthält darüberhinaus Entbehrliches, wie z.B. die volle Stufenlogik, die statt Fundament nur Kulisse ist. Als Ziel hat sich Willenborg gesetzt (S. 110): „Für jede Anfragesprache wird deren Syntax und Semantik exakt spezifiziert." Die Konstruktion der 5 vorgestellten Anfragesprachen ist jedoch überwiegend inkorrekt, insbesondere ist die „universelle Anfragesprache StructuredQL" ein Produkt zahlreicher Fehlgriffe. Daher wird Willenborg noch einen weiten Weg gehen müssen, um sein Ziel zu erreichen.

Literatur

Aho, Alfred A.; Hopcroft, John E.; Ullman, Jeffrey D. (1983): Data Structures and Algorithms, Addison-Wesley, Reading (Massachusetts).

Bollmann, Peter; Konrad, Erhard (1979): Mathematische Modelle von Information-Retrieval-Systemen, in: R. Kuhlen (Hrsg.), Datenbasen-Datenbanken-Netzwerke, Praxis des Information Retrieval, Band 2 (Konzepte von Datenbanken), K.G. Saur, München, S. 277-294.

Carnap, Rudolf (1958): Introduction to Symbolic Logic and its Applications, Dover Publications, New York. Deutsche Ausgabe: Symbolische Logik, Springer-Verlag, Wien 1973 (Nachdruck der 3. Auflage).

Ewert, Gisela; Umstätter, Walther (1997): Lehrbuch der Bibliotheksverwaltung, Hiersemann, Stuttgart.

HERMES, HANS (1972): Einführung in die mathematische Logik – Klassische Prädikatenlogik, Teubner, Stuttgart.

HILBERT, DAVID; ACKERMANN, WILHELM (1972): Grundzüge der theoretischen Logik, 6. Auflage, Springer-Verlag, Berlin.

KONRAD, ERHARD (1982): Modelle für Information Retrieval Systeme, in Tagungsband: Deutscher Dokumentartag 1981 in Mainz, Kleincomputer in Information und Dokumentation (bearb. v. H. Strohl-Goebel), K.G. Saur, München, S. 362-370.

KONRAD, ERHARD; REINER, ULRIKE (1985): Eine semantische Analyse des Information-Retrieval-Systems GRIPS, LIVE-Bericht 2/85 (LIVE: BMFT-Projekt Leistungsbewertung von Information Retrieval Verfahren), Fachbereich Informatik, TU Berlin.

KONRAD, ERHARD; REINER, ULRIKE (1986): Zur Semantik von Anfragesprachen für Dokumentennachweissysteme, in Tagungsband: Deutsche Gesellschaft für Dokumentation (Hrsg.), Deutscher Dokumentartag 1985 in Nürnberg, K.G. Saur, München, S. 180-197.

KUHLEN, RAINER (1991): Hypertext – Ein nicht-lineares Medium zwischen Buch und Wissensbank, Springer-Verlag, Berlin.

MITSCHDÖRFER, PIA; REINER, ULRIKE (1991): Dokumenten-, Fakten- und Erklärungssuchsysteme, in Tagungsband: W. Neubauer; U. Schneider-Brien (Hrsg.), Deutscher Dokumentartag 1990 in Fulda, Frankfurt am Main, S. 559-574.

REINER, ULRIKE (1991a): Die Semantik von Anfragesprachen und ihre Bedeutung für die Informationswissenschaft, in Tagungsband: W. Neubauer; U. Schneider-Brien (Hrsg.), Deutscher Dokumentartag 1990 in Fulda, Frankfurt am Main, S. 31-35.

REINER, ULRIKE (1991b): Anfragesprachen für Informationssysteme, Reihe Informationswissenschaft der DGD, Band 1, Frankfurt am Main.

SALTON, GERARD; MCGILL, MICHAEL J. (1983): Introduction to Modern Information Retrieval, McGraw-Hill, New York.

SCHMIDT, GUNTHER; STRÖHLEIN, THOMAS (1993): Relations and Graphs, Springer-Verlag, Berlin New York. Deutsche Ausgabe: Relationen und Graphen, Springer-Verlag, Berlin 1989.

ULLMAN, JEFFREY D. (1982): Principles of Database Systems, Second Edition, Computer Science Press, Rockville (Maryland).

WILLENBORG, JOSEF (2001): Anfragesprachen für Internet-Informationssysteme, Hochschulschrift, Humboldt-Universität zu Berlin. Online unter der WWW-Adresse http://www.josef-willenborg.de/.