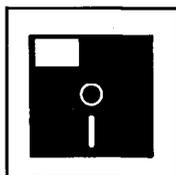
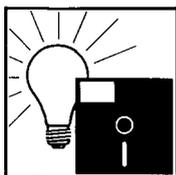


Überzeugen konnten dabei vor allem die Praktiker. Diese konnten jedoch nicht leugnen, zu einem endgültigen Urteil bezüglich der Effektivität des PC-Einsatzes am Richterarbeitsplatz noch nicht fähig zu sein. In Erinnerung bleibt mir die kritische Bemerkung des Stuttgarter OLG-Richters Nack, selbst Verfasser von zwei Programmen zum Thema Leasing – Abrechnung gekündigter Finanzierungs-Leasingverträge – und Kredit – Berechnung bei der Sittenwidrigkeits-Prüfung von Ratenkreditverträgen –, der darauf hinwies, daß ein Anwender seiner Programme stets in der Lage sein müsse, die zugrundeliegenden Rechenoperationen nachzuvollziehen (Stichwort „Das Urteil aus dem Computer gibt es nicht“). Bemerkenswert ist auch das vorläufige Resümee des Karlsruher OLG-Richters Zöller: Bei einem Versuch im Lande Baden-Württemberg räumten fast alle beteiligten Richter ein, der PC habe ihnen keine Zeitersparnis gebracht. Hergeben wollte ihn aber auch keiner der Kollegen mehr.

Insgesamt blieb wohl bei den meisten Teilnehmern ein sehr positiver Gesamteindruck. Die gegebenen Anregungen sollten jetzt so schnell wie möglich am „heimatlichen Arbeitsplatz“ umgesetzt werden.



 Parsing

In'jur-pc 3/90 (S. 505 – 512) und jur-pc 4/90 (S. 548 – 551) stellte Jürgen Oechsler „Gedanken zur Programmierung wissenschaftlicher Systeme in TurboProlog“ vor. Bei der Aufteilung des Beitrags in zwei Teile ist in der Mitte ein Abschnitt ausgefallen (wofür der Herausgeber die Verantwortung trägt, M.H.). Im folgenden wird dieser Passus ergänzend nachgetragen. Glücklicherweise behandelt der ausgefallene Abschnitt ein zusammenhängendes Thema, so daß er auch so lesbar bleibt. Es dürfte sich um eine Ausnahme zu „Murphy's Law“ handeln, das bei einem derartigen Versehen normalerweise eine völlige Zerstörung des Kontextes postulieren würde.

Parsing

Jürgen Oechsler

In Prolog ist grundsätzlich die Programmierung einer Schnittstelle möglich, die dem Benutzer die Eingabe in natürlicher Sprache erlaubt⁸. Der damit assoziierte Problembereich wird oft mit dem Parsing identifiziert, einer Teildisziplin der Künstlichen Intelligenz (vom englischen to parse, grammatisch zerlegen).

Die Konstruktion einer natürlich-sprachlichen Schnittstelle ist allerdings nicht unbedingt auf grammatische Analyse angewiesen; im Gegenteil zeigt sich die semantische Aufarbeitung menschlicher Sprache der grammatischen Untersuchung teilweise überlegen. Kehren wir daher kurz zum Ausgangsbeispiel zurück und prüfen es zur Abwechslung semantisch:

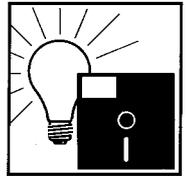
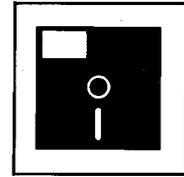
„Hat Berta gegen Anton einen Anspruch auf Zahlung des Kaufpreises aus § 433 II BGB?“

Die einzelnen Nomina dieses Satzes stehen zueinander in semantischer Verbindung, wie sich durch ein einfaches Prädikat aufweisen läßt:

verbindung (Bezugspunkt, Beziehung, Bezugspunkt)
 verbindung („Berta“, „gegen“, „Anton“)⁹
 verbindung („Berta“, „hat“, „Anspruch“)
 verbindung („Anspruch“, „auf“, „Zahlung des Kaufpreises“)
 verbindung („Zahlung des Kaufpreises“, „aus“, „§ 433 II BGB“)
 usw.

⁸ Vgl. dazu Kinnebrock (o. Fn. 2), S. 88ff; Schiödt (o. Fn. 2), S. 79 ff.; Turbo Prolog 2.0, Referenz, S. 375 ff; Weiskamp/Hengl (o. Fn. 2), S. 127ff.

⁹ Aus Gründen der Übersichtlichkeit werden jetzt auch die Eigennamen in Stringform geschrieben: „Berta“ statt berta. Ein sachlicher Unterschied besteht jedoch nicht, denn beide Male handelt es sich um Konstanten.



Die Beziehung beschränkt sich erkennbar nicht auf Paarbildungen, sondern die Bezugspunkte einer Verbindung stehen wiederum mit anderen in Kontakt. Ist diese Art Verflechtung aufgrund der Trivialität des Beispiels noch recht gut durchschaubar, so bilden Aussagesätze in der Regel komplizierte Vernetzungen. Als semantische Netze bezeichnet man daher auch die Datentypen, die solche Strukturen auf Programmebene repräsentieren¹⁰. Die „Bezugspunkt“ genannten Objekte innerhalb des Prädikates „verbindung(...)“ werden zu Knoten des Netzes, während dessen Schnüre oder Kanten von den Objekten repräsentiert werden, die im Prädikat „verbindung(...)“ „Beziehung“ heißen. Die Bezugspunkte eines Obersatzes lassen sich verallgemeinern und Unterfälle (Instanzen) dazu bilden:

```
unterfall(Oberbegriff, Instanzen).
unterfall(Anspruchsgrundlage, „§ 433 II BGB“).
unterfall(Anspruchsberechtigter, „Berta“).
unterfall(Anspruchsgegner, „Anton“).
unterfall(Anspruchsziel, „Kaufpreis“).
unterfall(„Kaufpreis“, „§ 433 II BGB“).
usw.
```

Die semantische Prüfung kann dem eigentlich auszuführenden Prädikat leicht vorge-schaltet werden:

```
anspruch__besteht if
verbindung(Anspruchsberechtigter, „gegenüber“, Anspruchsgegner)
and verbindung(Anspruchsberechtigter, „Anspruch auf“,
Anspruchsziel)
and verbindung(Anspruchsziel, „aus“, Anspruchsgrundlage)
and anspruch__auf(Anspruchsziel, Anspruchsberechtigter,
Anspruchsgegner, Anspruchsgrundlage).
```

So ließe sich auch das bekannte Prädikat „anspruch_auf(...)“ mit der Außenwelt in Verbindung setzen. Denn praktisch erfolgt die Bearbeitung so, daß ein weiterer Programmteil, der sogenannte Scanner den eingegebenen Satz in seine Bestandteile zerlegt und der hier in Grundzügen vorgestellte Parser die semantischen Beziehungen analysiert und für die weitere Programmausführung übersetzt.

Nun beschränkt sich die Wortwahl eines möglichen Anwenders natürlich nicht auf: den im semantischen Netz vorgegebenen Sprachschatz: Vielleicht fragt er im Obersatz danach, ob Berta einen Anspruch auf „Übereignung von Zahlungsmitteln“ hat. Die Lösung bietet hier eine Datenbank mit Synonymen, die durch ein einfaches Prädikat strukturiert werden kann:

```
synonym(Instanz, Synonym).
synonym(„Zahlung des Kaufpreises“, „Übereignung von Zahlungsmitteln“).
```

Dieses Prädikat läßt sich leicht bei der Eingabeprüfung einbauen, der Zugriff auf Datenbanken aber wird von Prolog unmittelbar unterstützt¹¹.

Darüber hinaus enthält fast jeder Satz Wörter, die für den Gesamtsinn nicht aussagekräftig sind: „eigentlich“, „hat“, „denn“. Auch führt der Lösungsweg über eine Datenbank, die über ein einfaches Prädikat wie `füllsel(Wort)` innerhalb der Eingabeprüfung für Unterdrückung dieser Wörter sorgt.

Je differenzierter ein System jedoch menschliche Sprache erfassen kann, umso weniger nicht aussagekräftige Worte wird es zu verarbeiten haben. Hier liegen auch die Grenzen des Systems, da die Ausdrucksnuancen natürlicher Sprache nur durch ein sehr eng gewobenes semantisches Netz analysiert werden können. Um einen vollständigen Sachverhalt zu verstehen, muß der Parser vor allem eine erhebliche Bandbreite von Beziehungen zwischen Objekten innerhalb der deutschen Sprache einordnen können. Die Nachzeichnung eines juristischen Obersatzes ist dabei vergleichsweise einfach, weil dieser selbst strukturiert ist. Einem kompletten Sachverhalt, der idiomatische Redewendungen enthält oder bei dem die Satzstellung bedeutungstragend ist, wird ohne Unterstützung durch grammatische Analyse und semantische Grundlagenforschung schwer beizukommen sein.

¹⁰ Weiskamp/Hengl (o. Fn. 2), S. 206ff.; Turbo Prolog 2.0, Referenz S. 381.

¹¹ Dazu sogleich.