

**Simon C. Dik:**

**Functional Grammar in Prolog: An Integrated Implementation for English, French, and Dutch. (Natural Language Processing 2) Berlin:** Mouton de Gruyter, 1992. 264 pp.; DM 118,- DM.

Das Buch ist eine Dokumentation zum Programmsystem ProfGlot, einem auf LPA Prolog abgestimmten 'multilingual natural language processor' mit Parsing-, Generierungs-, Übersetzungs- und Inferenzfunktionen (@1991 Amsterdam Linguistic Software). Die Programmoberfläche und die beiliegende Dokumentation präsentieren sich im gleichen Stil wie die ebenfalls von Amsterdam Linguistic Software vertriebene interaktive Einführung in LPA Prolog (@Logic Programming Associates Ltd, London) ProCourse (Dik/Kahrel1991).

Das Begleitbuch zu ProfGlot gliedert sich in folgende Teile (in Klammern jeweils Kapitel/Seiten): Einführung in ProfGlot (1/13), in Prolog (2/5-17) und in die Funktionale Grammatik [FG] nach Dik 1989 (3/19-30); Beschreibung des Programmsystems ProfGlot im Überblick (4-5/31-44) und nach Einzelmodulen und Funktionen (6-15/45-234); Anmerkungen, Literaturangaben, Verzeichnis der Prolog-Prädikate, Stichwort- und Namensregister (255/264). Die Präsentation ist professionell und das Layout ansprechend, es ist eines jener Bücher, die einen gewissen Anreiz zum Blättern und Lesen ausstrahlen, wenn man sie in Händen hält.

Bei der Bestellung für die Institutsbibliothek fragten wir uns, ob denn wohl die Programm-Diskette mit geliefert würde. Die Verlagsankündigung schwieg sich darüber aus - Vergessen oder Politik? - eine klare Aussage wäre jedenfalls wünschenswert gewesen. Wir kamen zu dem Schluß, daß, angesichts des stolzen Preises und der Tat-

sache, daß eine Softwarebeschreibung ohne die beschriebene Software zu wenig nutzen würde, das Programm enthalten wäre. Erste Enttäuschung: es war es nicht.

Wir bestellten ProfGlot bei Amsterdam Linguistic Software (es kostet ein Vielfaches des Buchpreises). Inzwischen machte ich mich an die Lektüre der gut 250 recht kostbaren Seiten (so etwa DM 0.43 jede) des Druckwerks, von der fidelen Hoffnung getragen, daß das Buch, wenn es schon bona fide - ohne Programm-Diskette verkauft wird, es auch einem Nur-Leser - bonae voluntatis - von Wert und Nutzen sein müsse. Es läßt und liest sich tatsächlich gut an: in gewohnt verständlicher Sprache und mit der typischen Dik'schen Suggestivität werden einige "Elemente von Prolog" und Grundlagen einer Implementation der FG in Prolog vorgestellt. Alles sieht prima aus: Prolog ist einfach und gut, FG ist einfach und gut, sie sind wie füreinander gemacht, und ihre Verbindung ist selbstverständlich auch einfach und das Beste überhaupt.

Das Grundkonzept von ProfGlot folgt den theoretischen Prämissen in Dik 1989. Das Programm ist der Versuch einer (Teil-)Modellierung der Kompetenz eines (mehrsprachigen) natürlichsprachlichen Sprechers ("M.NLU" - s.auch Dik 1990). Der modulare Aufbau spiegelt den der Theorie mit ihrer Ausrichtung auf 'the construction of linguistic expressions in a quasi-productive, bottom-up way' (Dik 1989:54) wider. Generator- und Parsing-Komponente funktionieren nach einheitlichen Regeln (vgl. Dik 1989:51); der Parser ist "eine Art invertierter Generator" mit dem 'a sentence is analysed by considering how it could be formed' (33). Die Übersetzung auf der Ebene von 'underlying clause structures' (die, daran sei erinnert, aus einfachen und abgeleiteten Prädikatstrukturen (basic and derived predicates), i.e. lexi-

kalischen Einträgen bestehen) funktioniert nach dem theoretisch postulierten Prinzip, daß 'translation will have to be reconstructed in much the same way as is done in a bilingual dictionary' (Dik 1989:86).

Der klare, modulare Aufbau der FG, die konsequente Formalisierung in Verbindung mit einer stets überzeugenden Nähe zur Sprachrealität, die Tatsache, daß auch das durchgehend eingehaltene Bemühen um allgemeine typologische Adäquatheit (fast) nirgends zu ungläubigen konstruktiven Klammern führt, erweckte in mir, wie vorher schon, die Erwartung, daß eine Computer-Implementierung der FG besonders adäquat, inspirierend und erfolgreich werden würde. Zweite Enttäuschung: sie wurde es nicht.

Die Beschreibung der Lexikonmodule (mit Basisprädikaten Englisch / Französisch / Niederländisch) beginnt mit der Bemerkung, daß es sich nur um ein geschränkte Beispiele handele. Warum, fragt man sich, diese Beschränkung schon beim einfachsten (basic!) Teil? Wäre es nicht interessanter gewesen, sich auf ein oder zwei Sprachen zu beschränken und sich dafür an einer tiefergehenden, auch in ihrer Problematik diskussionswerten Implementation zu versuchen?

Ein Beispiel (aus dem Englischen) für eine einfache Verbprädikatstruktur und ihre faktische Deklaration als Prolog-Prädikat (basic predicate frame verbal):

```
giveV(x1:<anim>(x1)»AG(x2)
GO(x3:<anim>(x3)» REC
(Dik 1989:54)
```

- entspricht:

```
bpredv(eng, [[give],[act],[[anim],t,[ag]],
[[inanim],t,[pt]],[[anim],t,[rec]]]) (48).
```

Die dreistellige FG-Prädikatstruktur ist als zweites Argument in ein Prolog-Prädikat integriert, das aus einer Sprachenangabe als erstem Argument und einer Liste mit drei Unterlisten besteht, von denen die letzte sich wiederum aus drei Listen mit je drei Elementen zusammensetzt. Die Liste an zweiter Argumentstelle

enthält, jeweils als Liste mit einem Element, die Zitierform ([give]) und den denotierten semantischen Typ (state of affairs (SoA): [act]) und die Liste mit Argumentrollen (semantic functions: [ag], [pt], [rec]) und Selektionsbeschränkungen ([anim], [inanim]) - abweichend von der Theorie (vgl. Dik 1989:76) nicht als einstellige Prädikate (<anim>(xi)).

Die Listenstruktur wird nur knapp erläutert, warum die FG-Prädikatstruktur so und nicht anders umgesetzt wurde, wird nicht erklärt, ebensowenig Status und Funktion des 't' in den Argument-Unterlisten (es kann in LPA-Prolog jedenfalls keine Variable sein). Nominale und adjektivische Prädikate sind - mutatis mutandis - ebenso deklariert.

Die semantische Beschreibung der Substantivprädikate stützt sich auf wenige marker ([hum], [masc], [fern] ...), deren Status, Auswahl, Erweiterbarkeit nicht angesprochen wird. Sie sind angeblich 'selfexplanatory' (46); einige Redundanzregeln werden durch Implikationsmechanismen realisiert (70).

Bei den Adjektiven werden grammatische ([grad<able>]), FG-relevante ([eval <uative>]) und semantische ([vert<ical>D Kategorien (47) als 'features' angegeben. Es werden vereinzelte 'meaning postulates' ohne weitere Begründung von Auswahl und Zusammenhang angeführt. Daneben werden einige semantische und grammatische Besonderheiten in sog. 'idioms' und 'paradigms' festgehalten. Die immer wieder beschworene Klarheit, Explizitheit, Gründlichkeit, die, ein Grundstein im Sinnfundament der Computerlinguistik, durch die Implementierung linguistischer Theorien als lauffähige Programme erzwungen würde, bleibt aus. Der Autor meint selbst, daß 'a more extensive and principled system of such restructions must no doubt be set up' (47). Wir warten gespannt auf Teil II - Titel: 'Real Functional Grammar in Prolog'?

Im längsten Einzelkapitel (9/69-108) des Buches wird das Kernstück des Systems, der 'universelle Generator' Uni Gen beschrieben, der für eine der auswählbaren

Sprachen (teilweise auf andere Sprachen übertragbare) 'fully specified underlying dause structures' (69) nach Maßgabe von veränderbaren random-Einstellungen generiert. Die allgemeinste Regel, fully -specified\_dause(L,X) verzweigt zu 'dause-', 'proposition'- und 'predication'-Bildungsregeln gemäß dem stratifikationellen Aufbau der FG. Der Aufbau wird, wie in der Theorie, von unten nach oben beschrieben. In den folgenden Kapiteln (10-12/109-167) werden die 'expression rules' zur Morphologie der Konstituenten, Wortstellung und zu sandhi-Erscheinungen (vor allem im Französischen) eingeführt. In den letzten drei Kapiteln (13-16; 169-234) werden das Parser, Übersetzungs- und Inferenzmodul beschrieben, alle, wie der Generator, leicht hyperbolisch als "universal" bezeichnet: UniPar, UniTra, UniLog. Letzteres baut auf (deklarierten) Hyponymiebeziehungen zwischen Lexemen (rose flower), Bedeutungs-postulaten (kiss touch [wiih the lips]) und satzlogischen Theoremen (p (p)) auf.

Die ständige Herausforderung an denjenigen/diejenige, der/die dieses Buch durcharbeitet, ist die, daß er/sie fast alles erraten oder aus der Theorie zu erschließen versuchen muß. Erklärt in irgendeinem interessanten Sinne des Wortes wird so gut wie gar nichts, Beispiele für die Anwendung der Prolog-Regeln mit Instanzierung der Variablen fehlen in Kap. 10 fast und in Kap. 9 völlig. Die in ProfGlot deklarierten Regeln werden, wenn überhaupt, nur äußerst knapp erläutert, so knapp, daß sie weder für einen wenig geübten Prolog-Benutzer wie mich, noch für einen besser trainierten Kenner verständlich sind (s. die Rezension von Patrick Saint-Dizier in Computational Linguistics 19/4 (1993) 695 f.).

Kostproben:

```
(77) Regel:
    predaderpred(L,[[[S,POL],
                    [SE, T, [stand]]],
                    [state],
                    [[SE,t,[zero]]]]) :-
    gamble(predform3), member(POL,[pos,neg]),
    choose_arb_adj(L,[[S],[gradl_],
                    [[SE,t,[zero]]]]),
    term(L,T,SE).
```

Seiten vorher (71), term hingegen 19 Seiten später (96) eingeführt. Darauf fehlt hier ein Hinweis. Es gibt wohl einen Index der Prädikate am Ende, dennoch steht an dieser Stelle der Erklärungszusammenhang für ein später eingeführtes Prädikat aus.

```
(92) Regel: do_all_terms(L,OP,PRED,PREDi) :-
    PRED = [P2,[Pi,[S,T,A],Si],S2],
    list_perform(L,OP,A,Ai),
    list_perform(L,OP,Si,Sii),
    list_perform(L,OP,S2,S22), PRED = [P2, [Pi,
    [S, T ,11],511],522] .
```

Erläuterung: 'We do a certain operation' "OP" on all terms by "list-performing" OP on the arguments, the satellites-1, and the satellites-2' (ohne Kommentar).

Es fällt mir schwer, den Nutzen dieses Buches zu erkennen, und ein Buch, dessen Nutzen ich nicht erkennen kann, finde ich schlecht. Und wenn ein besonders renommierter Wissenschaftler und Autor ein besonders schlechtes Buch herausbringt, das auch noch besonders teuer ist, so ist das besonders ärgerlich. Sei's drum, tröstete ich mich, vielleicht ist ja die tatsächliche Leistung des Programms überzeugender. Dritte Enttäuschung: sie ist es nicht.

Die erste Profglot- Version 2, die wir bekamen, führte auf einem normalen IBM386-PC mit 640 KB konventionellem und (nicht genutzten) 8 MB erweitertem Hauptspeicher zu vielfältigen Speicherüberlaufproblemen. Nach Reklamation bekamen wir eine modifizierte Version, die durch Ausklammerung von Menüs 23 KB mehr Hauptspeicher für die Ausführung der Regeln zur Verfügung stellt. Trotz vieler Haken und Ösen läßt sie sich soweit bedienen, daß man sich mit der linguistischen statt der technischen Funktionalität befassen kann.

Außer den im Buch referierten Sprachen gibt es in der Version 2 Sprachkomponenten für Dänisch, Italienisch, Spanisch/Galizisch (nur Generator) und Japanisch. Bei der Generierung ist es durchaus amüsant zu verfolgen, wie die erzeugten Sätze peu a peu vom leicht deplaziert Komödiantischen ("ehrlich gesagt, eeh, ich kann klugerweise husten"; «franchement,

Das Prädikat choose\_arb\_adj. wird sechs

eeh, sa enfant qui rigole cause du sac meurt probablement) ins stark agrammatisch Unsinnige (" von dat wen wird das buch des sacks das du berührst geschlagen werden?" j «probablement sa enfant qui rigole a. cause du sac du seau qui est donne par nn au professeur moins grand qui embrasse lui-meme meurt) abgleiten. Der Parser kann nicht alle Sätze analysieren, die der Generator vorher erzeugt hat, auch wenn man die Zufallseinstellungen aufeinander abstimmt.

Auch bei der Übersetzung (z.B. Deutsch-Englisch) ist nicht transparent, was wann und warum (nicht) funktioniert: manchmal wird/werden die Übersetzung/en ausgegeben, manchmal nur dause structures, oft auch ein lapidares 'no.' In der Inferenzkomponente erfahren wir, daß (in Auszügen): 'The following sentence: mary kicks john. entails: mary kicks john. john is kicked by mary. that mary does nOt kick john is nOt true. a female person touches a male person. that a person does nOt touch a male person is nOt true. it is nOt true that a person is nOt touched by aperson.' und hunderte ähnlich interessante Dinge.

Am Ende fragen wir uns: Ehrlich gesagt, eeh, kann es sein, daß wir alles in allem etwas veräppelt werden?

### Literatur

**Dik, Simon C. (1989):** The Theory of Functional Grammar. Part1: The Structure of the Clause. Dordrecht / Providence, RI: Foris.

**Dik, Simon C. (1990):** How to build a natural language user, in: M. Hannay / E. Vester (eds.): Working with Functional Grammar. Descriptive and Computational Applications. Dordrecht / Providence, RI: Foris (203-215).

**Dik, Simon C. / Kahrel, Peter (1991):** ProCourse: A Prolog Course for Linguists. Amsterdam: Amsterdam Linguistic Software.

*Nico Weber, Universität Bonn*

