Paul Koop

Algorithmic Recursive Sequence Analysis

Algorithmic Structuralism:

Formalizing Genetic Structuralism:

An Attempt to Contribute to Making Genetic Structuralism Falsifiable

Abstract:
A method for the analysis of discrete finite character strings is presented. Postmodern social philosophy is rejected. A naturalistic sociology with falsifiable models for action systems is approved. The algorithmic recursive sequence analysis (Aachen 1994) is presented with the definition of a formal language for social actions, a grammar inducer (Scheme), a parser (Pascal) and a grammar transducer (Lisp).

Algorithmic Recursive Sequence Analysis (Aachen 1994) is a method for analyzing finite discrete character strings.

Ndiaye, Alassane (Role-taking as a user modeling method: global anticipation in a transmutable dialogue system 1998) and Krausse, CC, & Krueger,FR (Unknown signals 2002) published equivalent methods. It is ingenious to simply think something simple.

Since the beginning of the 21st century, the construction of grammars from given empirical character strings has been discussed in computational linguistics under the heading of grammar induction (Alpaydin, E. 2008: Maschinelles Lernen, Shen, Chunze 2013: Effiziente Grammatikinduktion, Dehmer (2005) Strukturelle Analyse, Krempel 2016: Netze, Karten, Irrgärten). Mit sequitur definieren Nevill-Manning und Witten (Nevill-Manning Witten 1999: Identifying Hierarchical Structure in Sequences: A linear-time algorithm 1999) define a grammar induction for the compression of character strings. Graphs, grammars and transformation rules are of course just the beginning. Because a sequence analysis is only complete when, as in algorithmic recursive sequence analysis, at least one grammar can be specified for which a parser identifies the sequence as well-formed,

with which a transducer can generate artificial protocols that are equivalent to the empirical sequence under investigation and to which an inductor can produce at least one equivalent grammar. Gold (1967) formulated the problem in response to Chomsky (1965).

Algorithmic structuralism is consistent, empirically proven, Galilean, naturalistic, Darwinian and a nuisance for deeply hermeneutic, constructivist, postmodernist and (post)structuralist social philosophers. I welcome heirs who continue the work or seek inspiration.

A social action is an event in the possibility space of all social actions. The meaning of a social action is the set of all possible subsequent actions and their probability of occurrence. The meaning does not have to be understood interpretively, but can be reconstructed empirically. The reconstruction can be proven or falsified by probation tests on empirical protocols.

From the mid-1970s to the present, irrationalist or anti-rationalist ideas have become increasingly prevalent among academic sociologists in America, France, Britain, and Germany. The ideas are referred to as deconstructionism, deep hermeneutics, sociology of knowledge, social constructivism, constructivism, or science and technology studies. The generic term for these movements is (post)structuralism or postmodernism. All forms of postmodernism are anti-scientific, anti-philosophical, anti-structuralist, anti-naturalist, anti-Galilean, anti-Darwinian, and generally anti-rational. The view of science as a search for truths (or approximate truths) about the world is rejected.

The natural world plays little or no role in the construction of scientific knowledge. Science is just another social practice that produces narratives and myths no more valid than the myths of pre-scientific epochs.

One can observe the subject of the social sciences as astronomy observes its subject. If the object of the social sciences eludes direct access or laboratory experiments like celestial objects (court hearings, sales talks, board meetings, etc.), the only thing that remains is to observe it purely physically without interpretation and to record the observations purely physically. The protocols could of course also be interpreted without reference to physics, chemistry, biology, evolutionary biology, zoology, primate research and life science. This unchecked interpretation is then called astrology when observing the sky. In the social sciences, this unchecked interpretation is also called sociology. Examples are constructivism (Luhmann), systemic doctrines of salvation, postmodernism, poststructuralism, or theory of communicative action (Habermas). Rule-based agent models have therefore previously worked with heuristic rule systems. These control systems have not been empirically proven. As in astrology, one could of course also create computer models in sociology, which, like astrological models, would have little empirical explanatory content. Some call this socionics. However, the protocols can also be interpreted taking into account physics, chemistry, biology, evolutionary biology, zoology, primate research and life science and checked for empirical validity. The observation of celestial objects is then called astronomy. In the social sciences one could speak of socionomy or sociomatics. That's actually sociology. This would not result in big

world views, but as in astronomy, models with a limited range that can be empirically tested and can be linked to evolutionary biology, zoology, primate research and life science. These models (differential equations, formal languages, cellular automata, etc.) allow the deduction of empirically testable hypotheses, so they would be falsifiable. Such socionomy or sociomatics does not yet exist. I would prefer formal languages as model languages for empirically proven rule systems. Because rule systems for court hearings or sales talks, for example (models with limited range, multi-agent systems, cellular automata) can be modeled with formal languages rather than with differential equations.

Algorithmic structuralism is an attempt to help translate genetic structuralism (without omission and without addition) into a falsifiable form and to enable empirically proven systems of rules. The Algorithmically Recursive Sequence Analysis is the first systematic attempt at a naturalistic and computer-based formulation of genetic structuralism as a memetic and evolutionary model. The methodology of Algorithmic Recursive Sequence Analysis is Algorithmic Structuralism. Algorithmic structuralism is a formalization of genetic structuralism. Genetic structuralism (Oevermann) assumes an intention-free, apsychic possibility space of algorithmic rules that structure the pragmatics of well-formed chains of events in text form (Chomsky, McCarthy, Papert, Solomon, Lévi-Strauss, de Saussure, Austin, searle). Algorithmic structuralism is an attempt to make genetic structuralism falsifiable. Algorithmic structuralism is Galilean and as incompatible with Habermas and Luhmann as Galileo was with Aristotle. Of course, one can try to remain compatible with Luhmann or

Habermas and to algorithmize Luhmann or Habermas. All artefacts can be algorithmized, for example astrology or chess. And one can model normative agents of distributed artificial intelligence, cellular automata, neural networks and other models with heuristic protocol languages and rules. This is undoubtedly theoretically valuable. So there will be no sociological theoretical progress. A new sociology is sought that models the replication, variation, and selection of social replicators stored in artifacts and neural patterns. This new sociology will be just as incompatible with Habermas or Luhmann as Galileo could be with Aristotle. And their basic theorems will be as simple as Newton's laws. Just as Newton operationally defined the terms motion, acceleration, force, body and mass, this theory will algorithmically and operationally define the social replicators, their material substrates, their replication, variation and selection and secure them through sequence analysis. Social structures are linguistically coded and based on a digital code. We are looking for syntactic structures of a culture-encoding language. But this will not be a philosophical language, but a language that encodes and creates society. This language encodes the replication, variation, and selection of cultural replicators. On this basis, normative agents of distributed artificial intelligence, cellular automata, neural networks and other models will then be able to use protocol languages and rule systems other than heuristics in order to simulate the evolution of cultural replicators.

Algorithmic structuralism moves thematically in the border area between computer science and sociology. Algorithmic structuralism assumes that social reality itself (wetware, world 2) is not capable of calculation. In its reproduction and transformation, social reality leaves traces that are purely physical and semantically unspecific (protocols,

hardware, world 1). These traces can be understood as texts (discrete finite character strings, software, world 3). It is then shown that an approximation of the transformation rules of social reality (latent structures of meaning, rules in the sense of algorithms) is possible by constructing formal languages (world 3, software). This method is the Algorithmic Recursive Sequence Analysis. This linguistic structure drives the memetic reproduction of cultural replicators. This algorithmically recursive structure is of course not (sic!) compatible with Habermas and Luhmann. Galileo is not compatible with Aristotle either!

Through the production of readings and the falsification of readings, the system of rules is generated informally, sequence by sequence. The informal rule system is translated into a K-system. A simulation is then carried out with the K-System. The result of the simulation, a terminal, finite character string, is statistically compared with the empirically verified trace.

This does not mean that subjects in any sense of meaning follow rules in the sense of algorithms. Social reality is directly accessible only to itself. The inner states of the subjects are completely inaccessible. Statements about these inner states of subjects are derivatives of the found latent structures of meaning, rules in the sense of algorithms. Before an assumption about the inner state of a subject can be formulated, these latent structures of meaning, rules in the sense of algorithms, must first be validly determined as a space of possibility of meaning and meaning. Meaning does not mean an ethically good, aesthetically beautiful or empathetically comprehended life, but an intelligible connection, rules in the sense of algorithms.

The latent structures of meaning, rules in the sense of algorithms, diachronically generate a chain of selection nodes (parameter I), whereby they synchronously generate the selection node t+1 from the selection node t at time t (parameter II). This corresponds to a context-free formal language (K-systems), which generates the selection node t+1 from the selection node at time t by applying production rules.

Each selection node is a pointer to recursively nested K-systems. It is possible to zoom into the case structure like with a microscope. The set of K-Systems form a Case Structure Modeling Language "CSML".

The approximation can be brought as close as you like to the transformation of social reality. The productions are assigned dimensions that correspond to their empirically secured pragmatics/semantics. Topologically, they form a recursive transition network of discrete, nonmetric sets of events over which an algorithmic rule system works.

K-systems K are formally defined by an alphabet ($A := \{a_1, a_2, \ldots a_n\}$), all words above the alphabet (A*), production rules (p) the occurrence measure h (pragmatics/semantics) and an axiomatic first character string ($k_0 k_1 k_2 \ldots$):

$$K := (A, P, k_0)$$
$$A := \{a_1, a_2, \ldots a_n\}$$
$$p := A \rightarrow A$$
$$p(a_i) \in P$$
$$p := A \times H \times A$$
$$H := \{h \mid 0 \leq h \leq 100 \wedge h \in N\}$$
$$k_0 \in A^* \wedge k_i \in A$$

The appearance dimension h can be expanded in terms of game theory (cf. Diekmann).

Starting from the axiom k_0 , a K-system produces a character string $k_0k_1k_2$... by applying the production rule p to the character i of a string:

$$a_{i+1} := P(a_i)$$
$$k_i := a_{i-2}a_{i-1}a_i$$
$$k_{(i+1)} := a_{i-2}a_{i-1}a_i P(a_i)$$

A rigorous measure of the reliability of the assignment of the interacts to the categories (provisional Formative since in principle it can be approximated ad infinitum) is the number of assignments made by all interpreters (cf. MAYRING 1990, p.94ff, LISCH/KRIZ1978, p.84ff). This number then has to be normalized by relativizing the number of performers. This coefficient is then defined with:

$$R_{ars} := \frac{N * Z}{\sum_{i=1}^{N} I_i}$$

N:= Number of interpreters

Z:= Number of totally matching assignments

Ii:=Number of assignments of the interpreter Ii

An example session under clisp with the K-system for sales calls:

The example is the result of extensive sequence analyzes of salesin 1994, 1995 and 1996. Large amounts of traces of sales and purchase interactions were secured: tape records of interactions in retail and markets. A selection of these protocols were transcribed and subjected to extensive objective hermeneutic interpretation. A transcript from this selection was then subjected to a complex, complete algorithmic recursive sequence analysis. All work was extensively documented and fully summarized. (The documents will be made available in full on request.)

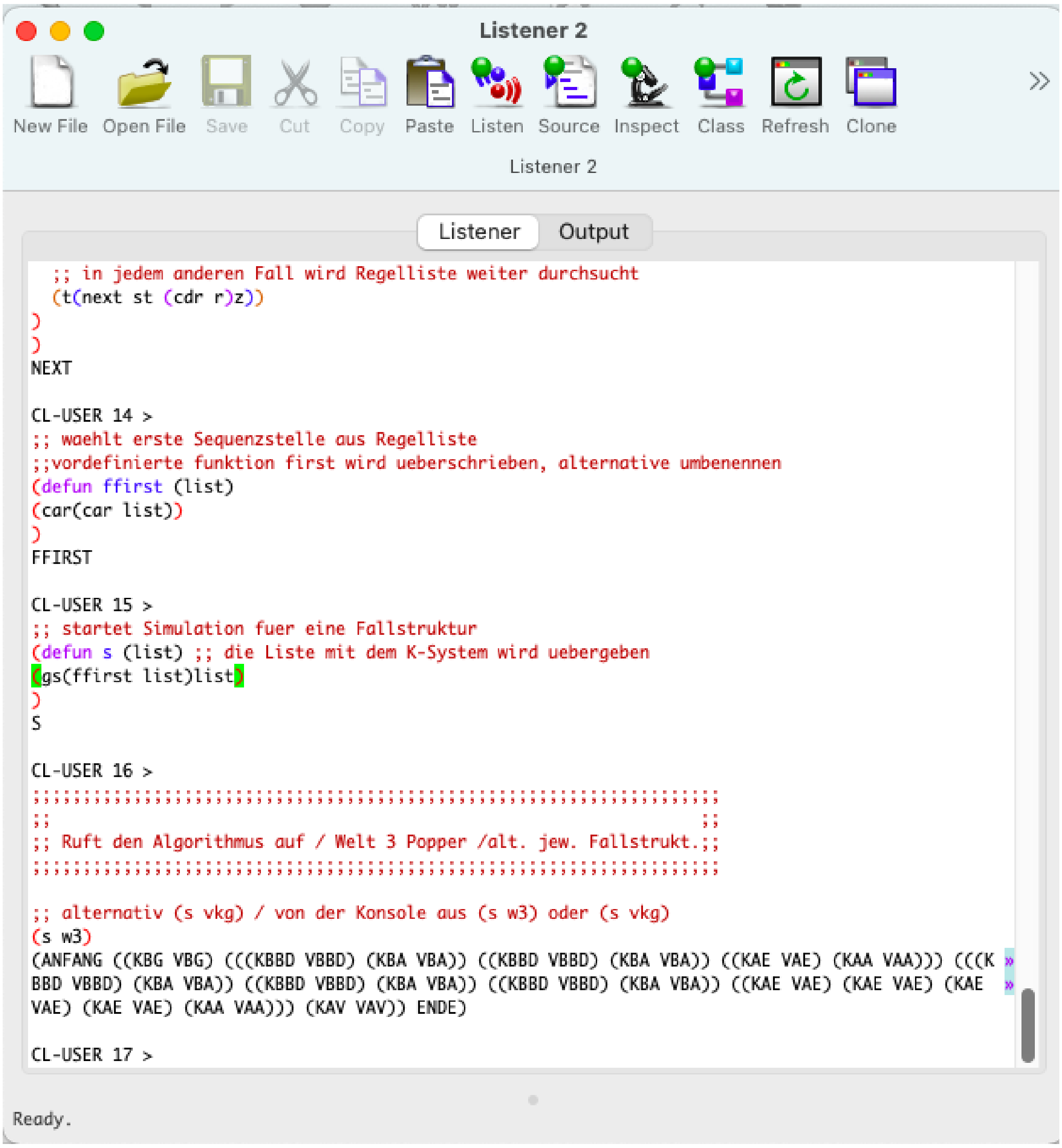
```
[3]> (s vkg)
((KBG VBG) (((KBBD VBBD) (KBA VBA)) ((KAE VAE) (KAA VAA)))
 (((KBBD VBBD) (KBA VBA)) ((KBBD VBBD) (KBA VBA)) ((KBBD VBBD) (KBA VBA))
  ((KAE VAE) (KAA VAA)))
 (KAV VAV))
[4]> (s vkg)
((KBG VBG)
 (((KBBD VBBD) (KBA VBA)) ((KAE VAE) (KAE VAE) (KAE VAE) (KAE VAE) (KAA VAA)))
 (((KBBD VBBD) (KBA VBA)) ((KAE VAE) (KAE VAE) (KAA VAA)))
 (((KBBD VBBD) (KBA VBA)) ((KAE VAE) (KAA VAA)))
 (((KBBD VBBD) (KBA VBA)) ((KBBD VBBD) (KBA VBA)) ((KAE VAE) (KAA VAA)))
 (KAV VAV))
[5]> (s vkg)
((KBG VBG)
 (((KBBD VBBD) (KBA VBA)) ((KBBD VBBD) (KBA VBA))
  ((KAE VAE) (KAE VAE) (KAA VAA)))
 (((KBBD VBBD) (KBA VBA)) ((KBBD VBBD) (KBA VBA)) ((KBBD VBBD) (KBA VBA))
  ((KBBD VBBD) (KBA VBA)) ((KAE VAE) (KAA VAA)))
 (((KBBD VBBD) (KBA VBA)) ((KAE VAE) (KAA VAA)))
 (((KBBD VBBD) (KBA VBA)) ((KBBD VBBD) (KBA VBA)) ((KAE VAE) (KAA VAA)))
 (((KBBD VBBD) (KBA VBA)) ((KAE VAE) (KAE VAE) (KAA VAA)))
 (((KBBD VBBD) (KBA VBA)) ((KBBD VBBD) (KBA VBA)) ((KAE VAE) (KAA VAA)))
 (KAV VAV))
[6]> _
```

Paul Koop K-System VKG transducer sales pitch in Lisp

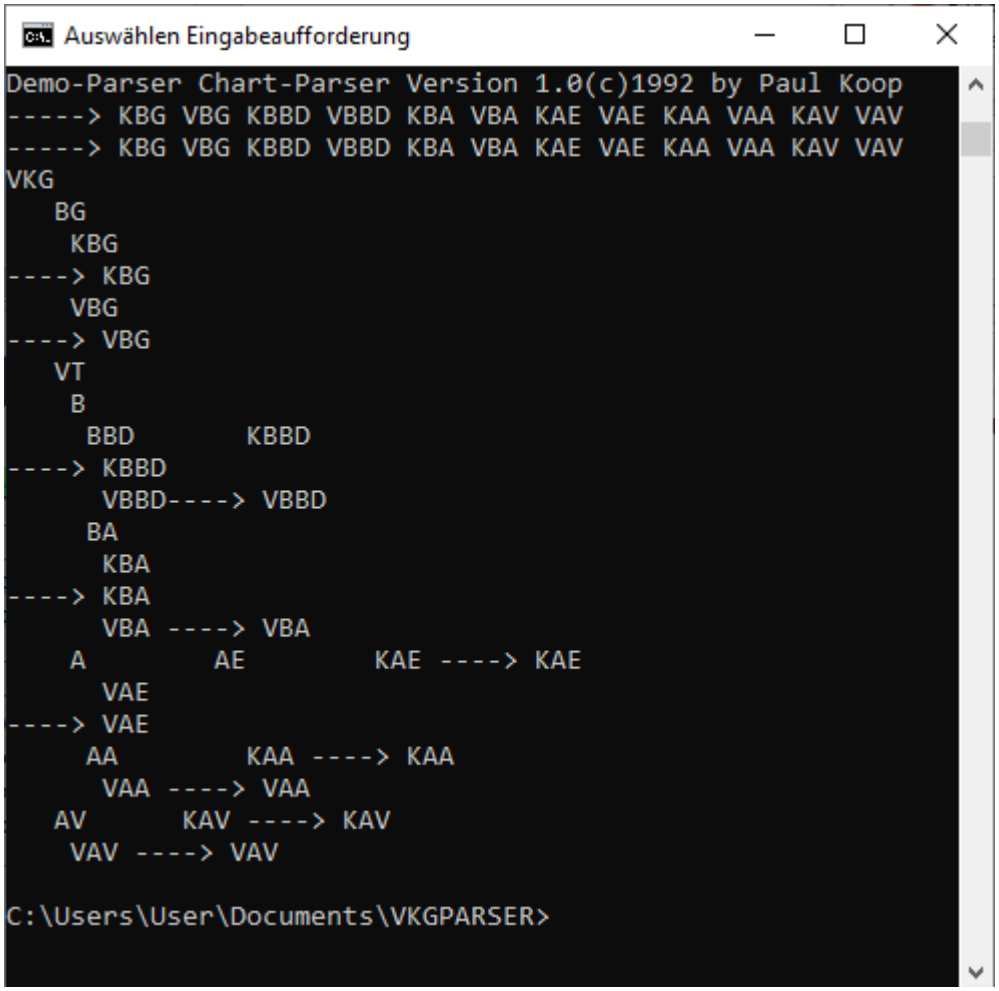
```
;; Korpus
   (define korpus (list 'KBG 'VBG 'KBBd 'VBBd 'KBA 'VBA 'KBBd 'VBBd
                        'KBBd 'VBBd 'KBA 'VBA 'KBBd 'VBBd 'KBA 'VBA 'KAE
                        'VAE 'KAE 'VAE 'KAA 'VAA 'KAV 'VAV);

;; Lexikon
   (define lexikon (vector 'KBG 'VBG 'KBBd 'VBBd 'KBA 'VBA 'KAE 'VAE
                           'KAA 'VAA 'KAV 'VAV)) ;; 0 - 12

➤ (transformationenZaehlen korpus)
➤ (grammatikerstellen matrix)
(KBG -> . VBG)
(VBG -> . KBBd)
(KBBd -> . VBBd)
(VBBd -> . KBBd)(VBBd -> . KBA)
(KBA -> . VBA)
(VBA -> . KBBd)(VBA -> . KAE)
(KAE -> . VAE)
(VAE -> . KAE)(VAE -> . KAA)
(KAA -> . VAA)
(VAA -> . KAV)
(KAV -> . VAV)
➤ (matrixausgeben matrix)
010000000000
001000000000
000400000000
001030000000
000003000000
002000100000
000000020000
000000101000
000000000100
000000000010
000000000001
000000000000
➤
```

	0	1	2	3	4	5	6	7	8	9	10	11
0	-	1										
1	-		1									
2	-			4								
3	-		1	-	3							
4	-		-			3						
5			2				1					
6	-		-				-	2				
7	-	-	-	-	-	-	1	-	1			
8	-									1		
9	-										1	
10	-	-	-	-	-	-	-	-	-	-	-	1
11												

Paul Koop K-System VKG inductor session Scheme



Paul Koop K-System VKG transducer session with Lisp



Paul Koop K-System VKG PARSER session at the console (Created with Object Pascal)

The characters of the character string have no predefined meaning. Only the syntax of their combination is theoretically relevant. It defines the case structure. The semantic interpretation of the signs is solely an interpretive achievement of a human reader. In principle, a visual interpretation (which can be animated) is also possible, for example for the automatic synthesis of film sequences.

A human reader can interpret the characters:

sales	talks VKG
sales	VT
requirement	B
conclusion	A
greeting	BG
required	Bd
requirement argument	BA
final objections	AE
sale	AA
farewell	AV
prepended K	customer
prepended V	seller

1	(setq vkg '(((s bg)100(s vt)) ((s vt)50(s vt)) ((s vt)100(s av))))	parameters II
2	(setq av '((kav 100 vav)))	parameters II
3	(setq bg '((kbg 100 vbg)))	parameters II
4	(setq vt '(((sb)50(sb)) ((sb)100 (sa))))	Parameter II

5	(setq a '(((s ae)50(s ae)) ((s ae)100(s aa))))	Parameter II
6	(setq b '(((s bbd) 100 (s ba))))	Parameter II
7	(setq aa '((kaa 100 vaa)))	Parameter II
8	(setq ae '((kae 100 vae)))	Parameter II
9	(setq ba '((kba 100 vba)))	Parameter II

10	(setq bbd '((kbbd 100 vbbd)))	Parameter II
11	(defun gs (sr) (cond ((equal s nil)nil) ((atom s)(cons s(gs(next sr(random 100))r))) (t (cons(eval s)(gs(next sr(random 100))r)))))	Parameter I
12	(defun next (srz) (cond ((equal r nil)nil) ((and(<=z(car(cdr(car r))))) (equal s(car(car r))))(car(reverse(car r)))) (t(next s (cdr r)z))))	Parameter I
13	(defun first (list) (car(car list)))	Parameter I
14	(defun s ( ) (setq protocol(gs(first vkg)vkg)))	Parameter I

It was a reliability coefficient of

$$R_{ars} = \frac{2*35}{118} = 0.59$$

Correlations

		Kodierer1	Kodierer2
Kodierer1	Correlation		,59
	Sig.		,09
Kodierer2	Correlation	,59	
	Sig.	,09	

Test Statistics

	Kodierer1	Kodierer2
Chi-Square	2,60	2,00
df	6	5
Asymp. Sig.	,86	,85

measured.

However, social reality itself is not capable of calculation and is only accessible to itself at the moment of transformation.

Humanities, constructivist and postmodern approaches are methodologically foreign to me. I left Mead, Parsons, Weber, Simmel, Mannheim/Scheler, Berger/Luckmann, Maturana, Varela, Habermas and Luhmann behind me. Albert, Axelrod, Esser, Diekmann, Troitzsch, Popper, Brezinka, Rössner, Dawkins, Dennett, Hofstadter, Rucker, Blackmore convince me more. Personally, I prefer a linguistic evolutionary perspective and the associated modeling of cultural replicators with formal languages. From the discrete structure of matter emerges the linguistic structure of biological evolution and the linguistic structure of cultural replicators. I therefore prefer an algorithmic structuralism.

Attachments

1. Sequence Analysis
2. Categories
3. Grammar Transducer (Lisp)
4. Grammar Parser (Pascal)
5. Grammar Inducer (Scheme)

Erklärung von Protokollzeichen:

laanngsaam:	gedehntes Sprechen
LANGSAM:	lautes Sprechen
A: langsamm ┐	
	gleichzeitig
B: hallo ┘	
:	Forscher
M:	Mann
F:	Frau
m:	Junge
f:	Mädchen
K:	Käufer
V:	Verkäufer
P:	einfach Person
(Kommentare):	Kommentar
...	Pause

*** Anfang Text ***
blablabla
*** Ende Text ***

1.*** Anfang Text4 ***

Paraphrase:
Ein Anfang eines Textes wird markiert.[40]
Intention:
Der Leser soll wissen, daß hier ein neuer Text beginnt.
Regulative Regel:
-
Konstitutive Regel:
Ein Text ist nur ein lesbarer Text, wenn er Anfang und Ende hat.

2.: Markt, 11.00 Uhr (Aachen, 28.06.94, Gemüsestand)

Paraphrase:
Der Text bezieht sich auf ein Ereignis auf einem Markt, an einem bestimmten Ort zu einer bestimmten Zeit.
Intention:
Der Text erhält eine historische und funktionale Einordnung. Die Ereignisse spielen sich im Rahmen der sozialen Institution Markt ab. Diese Institution führt an einem Ort Angebot und Nachfrage zusammen. Verkäufer stellen sich und ihre Ware vor. Käufer äußern ihre Kaufwünsche. Güter werden gegen Geld in Höhe ihres Preises getauscht.
Regulative Regel:
Nur zahlungsfähige Kunden erhalten Ware.
Konstitutive Regel:
In einer arbeitsteiligen Gesellschaft gibt es ohne Markt und ohne Zahlungsmittel keinen Güteraustausch.

3. (unverständlich)

Paraphrase:
Das Gespräch ist unverständlich. Es läuft aber schon. Der weitere Text bezieht sich auf ein schon laufendes Gespräch.
Intention:
Es soll deutlich werden, daß ein laufendes Gespräch verfolgt wird.
Regulative Regel:
-
Konstitutive Regel:
-

4.FK1.1: Hör ens, ich nehm ein paar Champignons mit.

Paraphrase:
Eine Käuferin spricht eine andere Person in der Du-Form an und teilt ihr mit, daß sie einige Champignons mitnimmt.
Intention1:

40 Das die Interpretation hier beginnt, wird einen objektiven Hermeneuten erstaunen. Und er hat ja auch aus seiner Perspektive Recht. Der Anfangsmarker des Textes hat nichts mit der protokollierten Handlung zu tun. Denn gegen alle Beteuerungen ist der objektive Hermeneut immer versucht, irgendwie hinter dem Text die Handlung zu retten. der algorithmische Deskriptionist aber weiß genau, daß er die Handlung nicht mehr erreichen kann. Also Angehörige der Welt 2 ist die Handlung entgültig verloren. Dem algorithmischen Deskriptionisten bleibt nur der Text. Der Text aber, als Angehöriger der Welt 3 ist vollständig zu interpretieren.

Die Käuferin kennt die andere Person, die über Champignons verfügt, persönlich, und glaubt, von diesen Champignons einige nehmen zu dürfen.
Regulative Regel1:
Von Freunden kann man Gefälligkeiten erwarten, weil man ihnen auch Gefälligkeiten gewährt.
Konstitutive Regel1:
Gefälligkeiten sind Merkmal von Freundschaften.
Intention2:
Die Käuferin will die Qualität der Champignons prüfen, um gegebenenfalls später zu kaufen.
Regulative Regel2:
Käufer können von Verkäufern die Möglichkeit erwarten, die Qualität der Ware zu prüfen.
Konstitutive Regel2:
Ein Kauf ist nur ein richtiger Kauf, wenn man nicht die Katze im Sack kaufen muß.
Intention3:
Die Käuferin beabsichtigt, ein paar Champignons zu kaufen.
Regulative Regel3:
Um eine Ware zu erhalten, muß man einen Kaufwunsch äußern.
Konstitutive Regel3:
Einem Kauf geht der Kaufwunsch notwendig voraus.

5.FV1.1: Eh, braune?
Paraphrase:
Die Verkäuferin möchte wissen ob die Käuferin braune oder helle Champignons nehmen will.
Intention1:
Der Wunsch, Champignons einfach zu nehmen, könnte akzeptiert werden.
Regulative Regel1:
siehe 4.FK1.1
Konstitutive Regel1:
siehe 4.FK1.1
Intention2:
Der Wunsch, Champignons zu probieren, wird akzeptiert.
Regulative Regel2:
siehe 4.FK1.1
Konstitutive Regel2:
siehe 4.FK1.1
Intention3:
Der Wunsch, Champignons zu kaufen, wird akzeptiert.
Regulative Regel3:
siehe 4.FK1.1
Konstitutive Regel3:
siehe 4.FK1.1

6.FK1.2: Ne helle.
Paraphrase:
Die Käuferin teilt mit, daß sie helle Champignons wünscht.
Intention1:
Die Käuferin veranlaßt die Verkäuferin, ihr helle Champignons einfach zu geben.

Regulative Regel1:
siehe 4.FK1.1
Konstitutive Regel1:
siehe 4.FK1.1
Intention2:
Die Käuferin veranlaßt die Verkäuferin, sie helle Champignons probieren zu lassen.
Regulative Regel2:
siehe 4.FK1.1
Konstitutive Regel2:
siehe 4.FK1.1
Intention3:
Die Käuferin veranlaßt die Verkäuferin, ihr helle Champignons zu verkaufen.
Regulative Regel3:
siehe 4.FK1.1
Konstitutive Regel3:
siehe 4.FK1.1

7.FV1.2: Helle.
Paraphrase:
Die Verkäuferin bestätigt, daß die Käuferin helle Champignons wünscht.
Intention1:
Die Verkäuferin will der Käuferin helle Champignons einfach geben.
Regulative Regel1:
siehe 4.FK1.1
Konstitutive Regel1:
siehe 4.FK1.1
Intention2:
Die Verkäuferin will der Käuferin helle Champignons zum Probieren überlassen.
Regulative Regel2:
siehe 4.FK1.1
Konstitutive Regel2:
siehe 4.FK1.1
Intention3:
Die Verkäuferin will der Käuferin helle Champignons verkaufen.
Regulative Regel3:
siehe 4.FK1.1
Konstitutive Regel3:
siehe 4.FK1.1

8.FK1.3: mhmh.
Paraphrase:
Die Käuferin bestätigt, daß sie helle Champignons wünscht.
Intention1:
Die Käuferin veranlaßt die Käuferin, ihr helle Champignons einfach zu geben.
Regulative Regel1:
siehe 4.FK1.1
Konstitutive Regel1:

siehe 4.FK1.1
Intention2:
Die Käuferin veranlaßt die Verkäuferin, sie helle Champignons probieren zu lassen.
Regulative Regel2:
siehe 4.FK1.1
Konstitutive Regel2:
siehe 4.FK1.1
Intention3:
Die Käuferin veranlaßt die Verkäuferin, ihr helle Champignons zu verkaufen.
Regulative Regel3:
siehe 4.FK1.1
Konstitutive Regel3:
siehe 4.FK1.1

9. (unverständlich)
Paraphrase:

-
Intention1:
-
Regulative Regel1:
-
Konstitutive Regel1:
-

10.FK1.4: Meinen se nich.
Paraphrase:
Die Käuferin hat in die Sie- Form gewechselt und fragt ob die Verkäuferin helle oder braune Champignons besser findet.
Intention1:
Sie will die Ware einfach nehmen (siehe 4.FK.1.1). Diese Lesart wird unplausibel, wenn man berücksichtigt, daß die Käuferin in die Sie- Form gewechselt ist. Die Lesart kann aber noch nicht ausgeschlossen werden.
Regulative Regel1:
siehe 4.FK.1.1
Konstitutive Regel1:
siehe 4.FK.1.1
Intention2:
Die Ware soll getestet werden (siehe 4.FK.1.1).
Regulative Regel2:
siehe 4.FK.1.1
Konstitutive Regel2:
siehe 4.FK.1.1
Intention3:
Die Ware soll gekauft werden (siehe 4.FK.1.1).
Regulative Regel3:
siehe 4.FK.1.1
Konstitutive Regel3:
siehe 4.FK.1.1
Intention4:

Die Käuferin ist unsicher und bittet um Rat, welche Champignons sie nehmen soll.
Regulative Regel4:
Käufer kaufen keine Ware, sondern Problemlösungen, die Käuferin bittet um eine Problemlösung durch die Verkäuferin.
Konstitutive Regel4:
Ein Verkaufsgespräch ist nur ein Verkaufsgespräch, wenn der Verkäufer um Rat gefragt werden kann.

11.FV1.3: Ja is ejal, se sinn beide frisch.
Paraphrase:
Die Verkäuferin hat die Frage als Bitte um Rat verstanden. Sie gibt den Rat, daß in bezug auf die Frische der Ware, das Merkmal helle Champignons, vom Merkmal braune Champignons nicht zu unterscheiden ist, und daher dem Zufall überlassen werden kann.
Intention1:
Der Rat unterstützt den Wunsch der Käuferin, die Ware einfach zu nehmen.
Regulative Regel1:
siehe 4.FK.1.1
Konstitutive Regel1:
siehe 4.FK.1.1
Intention2:
Der Rat unterstützt den Wunsch, die Ware zu testen.
Regulative Regel2:
siehe 4.FK.1.1
Konstitutive Regel2:
siehe 4.FK.1.1
Intention3:
Der Rat unterstützt den Wunsch, die Ware zu kaufen.
Regulative Regel3:
siehe 4.FK.1.1
Konstitutive Regel3:
siehe 4.FK.1.1
Intention4:
Die Verkäuferin ist langsam genervt. Sie will die Käuferin los werden und überlegt nicht lange, sondern täuscht einen echten Ratschlag vor, um möglichst schnell zum Kaufabschluß zu kommen.
Regulative Regel4:
Der Laden muß weiterlaufen.
Konstitutive Regel4:
In schwierigen Situationen kann die Kommunikation nur aufrecherhalten werden, wenn man flexibel genug ist, auch einmal zu "mogeln".

12.FK1.5: Oder, wie is et denn mit, mit, eh
13.FV1.4: Die können se länger liejen lassen.
Paraphrase 12.FK1.5:
Die Käuferin ist sich tatsächlich in ihrer Entscheidung unsicher geworden. Sie scheint einen neuen Wunsch gebildet zu haben.

Intention1 12.FK1.5:
Es ist nun entgültig unplausibel geworden, daß sie einen anderen, als einen Kaufwunsch hat. Um einen anderen, als einen Kaufwunsch jetzt noch aufrecht zu erhalten, müßte die Käuferin eine sehr hoch gestellte Persönlichkeit, oder die Verkäuferin extrem gelassen, oder aber die Käuferin pathologisch unsensibel sein. Alle Lesarten, die sich nicht auf einen anderen als einen Kaufwunsch beziehen werden jedenfalls ab hier aufgegeben.
Regulative Regel1 12.FK1.5:
Ein Kunde darf in Grenzen einen Kaufwunsch im Kaufgespräch ungestraft ändern
Konstitutive Regel1 12.FK1.5:
Nur Verkaufsgespräche, die flexibel genug sind, auf Unsicherheiten von kaufwilligen Käufern zu reagieren, sind erfolgreich.
Paraphrase 13.FV1.4:
Die Verkäuferin liefert einen konkreten Grund für die Wahl von hellen Champignons nach. Sie rät, zu berücksichtigen, daß helle Champignons länger gelagert werden können.
Intention1 13.FV1.4:
Die Verkäuferin will einen echten Rat geben.
Regulative Regel1 13.FV1.4:
Verkäuferinnen müssen auf Unsicherheiten der Kunden eingehen.
Konstitutive Regel1 13.FV1.4:
Verkaufsgespräche sind auch Beratungsgespräche.
Intention2 13.FV1.4:
Die Verkäuferin will ihren vorgetäuchten Rat wasserdicht gegen Aufdeckung absichen.
Regulative Regel2 13.FV1.4:
Je "wasserdichter" eine Täuschung ist, um so geringer ist das Risiko, entdeckt zu werden.
Konstitutive Regel2 13.FV1.4:
Die Qualität einer Täuschung wächst mit ihrer "Wasserdichtigkeit".

14.FK1.6: Neh, aber Pfifferlinge.
Paraphrase:
Die Käuferin setzt 13.FK1.5 fort und fragt nach Pfifferlingen.
Intention1:
Die Käuferin will der Verkäuferin signalisieren, daß sie Pfifferlinge kaufen will.
Regulative Regel1:
Ein Käufer muß seinen Kaufwunsch signalisieren.
Konstitutive Regel1:
Ohne Kaufwunsch kein Kauf.

15.FV1.5: Ah, die sinn super.
Paraphrase:
Die Verkäuferin stellt begeistert fest, daß die

Pfifferlinge ganz toll sind.
Intention1:
Die Verkäuferin preist die Qualität der Pfifferlinge an und will so Kaufbestätigung geben.
Regulative Regel1:
Verkäuferinnen müssen zu einem Verkaufsabschluß kommen.
Konstitutive Regel1:
Nur Verkaufsgespräche mit Verkaufsabschluß sind Verkaufsgespräche.
Intention2:
Die Verkäuferin ist froh, daß die Käuferin nicht mehr zwischen hellen und braunen Champignons schwankt und will die Käuferin auf die Pfifferlinge festlegen, um zu einem Abschluß zu kommen.
Regulative Regel2:
Verkäuferinnen müssen zu einem Verkaufsabschluß kommen.
Konstitutive Regel2:
Nur Verkaufsgespräche mit Verkaufsabschluß sind Verkaufsgespräche.

16. (unverständlich)

17.FK1.7: Kann ich die denn in Reissalat tun?
Paraphrase:
Die Käuferin fragt, ob sie die Pfifferlinge in Reissalat mitverarbeiten kann.
Intention1:
Die Käuferin möchte sich Reissalat mit Pilzen machen und deshalb wissen, ob das mit Pfifferlingen geht.
Regulative Regel1:
Käufer dürfen im Verkaufsgespräch fragen stellen.
Konstitutive Regel1:
Verkaufsgespräche sind nur Verkaufsgespräche, wenn Fragen der Käufer zugelassen sind.

18. (Unverständlich)

19.FK1.8: Brauch ich nich abzukochen oder was?
Paraphrase:
Die Käuferin fragt, ob sie die Pfifferlinge abkochen muß, bevor sie die Pfifferlinge im Reissalat verarbeiten kann.
Intention1:
Die Käuferin beabsichtigt, Pfifferlinge im Reissalat zu verarbeiten, möchte aber wissen, ob es gefährlich ist, die Pfifferlinge im Reissalat unbehandelt zu verarbeiten.
Regulative Regel1:
Käufer dürfen im Verkaufsgespräch Fragen stellen.
Konstitutive Regel1:
Verkaufsgespräche sind nur Verkaufsgespräche, wenn Fragen der Käufer zugelassen sind.
Intention2:
Die Käuferin beabsichtigt, Pfifferlinge im Reissalat zu verarbeiten, möchte aber wissen, ob der Reissalat mit abgekochten Pfifferlingen besser schmeckt.
Regulative Regel2:
Käufer dürfen im Verkaufsgespräch Fragen stellen.
Konstitutive Regel2:
Verkaufsgespräche sind nur Verkaufsgespräche, wenn Fragen der Käufer zugelassen sind.

20.FV1.6: Ehh, roh, doch müssen se en bischen in de Pfanne tun.
Paraphrase:
Die Verkäuferin sagt, daß Pfifferlinge vor der Verarbeitung in Reissalat mit Hitze behandelt werden müssen.
Intention1:
Die Verkäuferin will die Käuferin warnen, die Pfifferlinge vor der Verarbeitung in Reissalat abzukochen.
Regulative Regel1:
Sind mit der Ware Gefahren verbunden, die dem Käufer nicht offensichtlich sind, muß der Verkäufer auf die Gefahr aufmerksam machen.
Konstitutive Regel1:
-
Intention2:
Die Verkäuferin will der Käuferin einen Zubereitungstip

geben.
Regulative Regel2:
Verkäufer verkaufen keine Ware, sondern "helfen" dem Käufer bei "Problemlösungen".
Konstitutive Regel2:
-

21.FK1.9: Tuh ich.
Paraphrase:
Die Käuferin sagt, daß sie die Pfifferlinge in die Pfanne tun wird.
Intention1:
Die Käuferin quittiert den Rat der Verkäuferin.
Regulative Regel1:
-
Konstitutive Regel1:
-

22.FV1.7: Klein bischen.
Paraphrase:
Die Verkäuferin rät, die Pfifferlinge ein wenig in der Pfanne zu erhitzen.
Intention1:
Die Verkäuferin bekräftigt ihren Rat.
Regulative Regel1:
-
Konstitutive Regel1:
-

23. (unverständlich)

24.FK1.10:Die kann ich aber, ehm, in en Reissalat tun.
Paraphrase:
Die Käuferin will erneut wissen, ob sie die Pfifferlinge in den Reissalat tun kann.
Intention1:
Die Käuferin will wissen, ob sie die Pfifferlinge nach der Behandlung wirklich in Reissalat verarbeiten kann. Sie ist unsicher, entweder weil sie fortgeschrittenen Alters ist, oder weil sie tatsächlich psychisch instabil ist.
Regulative Regel1:
Auf Fragen älterer oder psychisch instabiler Personen antwortet man geduldigt, besonders, wenn sie zahlungsfähige Kunden sind.
Konstitutive Regel1:
Ältere oder psychisch instabile Personen dürfen auch "fünf mal" nachfragen.

25.FV1.8: Ja, datt is kein Problem, se müssen so nur...
Paraphrase:
Die Verkäuferin bestätigt erneut, daß es kein Problem ist, die Pfifferlinge behandelt in Reissalat zu verarbeiten.
Intention1:
Die Verkäuferin hat die Käuferin als ältere oder psychisch

instabile Person akzeptiert und läßt sich geduldig auf Wiederholungen ein.
Regulative Regel1:
Auf Fragen älterer oder psychisch instabiler Personen antwortet man geduldig, besonders, wenn sie zahlungsfähige Kunden sind.
Konstitutive Regel1:
Ältere oder psychisch instabile Personen dürfen auch "fünf mal" nachfragen.

26.FK1.11:bischen, ja
Paraphrase:
Die Käuferin sagt, daß sie die Pfifferlinge ein wenig in der Pfanne erwärmen wird.
Intention1:
Die Käuferin bestätigt, daß sie die Verkäuferin verstanden hat.
Regulative Regel1:
-
Konstitutive Regel1:
-

27.FV1.9: Bischen in eh, nitt wie de Champignons, die tuh ich ej auch roh erein.
Paraphrase:
Die Verkäuferin sagt, daß sie, im Gegensatz zu den Champignons, die Pfifferlinge nur erwärmt in einen Reissalat tun würde.
Intention1:
Die Verkäuferin hat die Käuferin entgültig als nicht "voll zurechnungsfähig" eingestuft und ermahnt sie durch ein Gegenbeispiel, die Pfifferlinge auch wirklich nur erwärmt zu verarbeiten.
Regulative Regel1:
Psychisch instabile Kunden werden wie Kinder als beschränkt geschäftsfähig behandelt, um wenigstens noch einen Verkaufsabschluß rechtfertigen zu können.
Konstitutive Regel1:
Ein Kaufvertrag kann nur mit einem Kunden zustandekommen, der mindestens beschränkt geschäftsfähig erscheint.

28.FK1.11:eh ja.
Paraphrase:
Die Käuferin sagt, daß sie, im Gegensatz zu Champignons , die Pfifferlinge erwärmen und dann erst in den Reissalat tun wird.
Intention1:
Die Käuferin bestätigt den Rat der Verkäuferin.
Regulative Regel1:
-
Konstitutive Regel1:
-
Intention2:
Die Käuferin merkt, daß sie wie ein kleines Kind behandelt

wird und will den Rat der Verkäuferin unkommentiert quittieren.
Regulative Regel2:
Eine erwachsene Person wird nur akzeptiert, wenn sie Zweifel an ihrer Zurechnungsfähigkeit zurückweist.
Konstitutive Regel2:
Nur wer voll zurechnungsfähig ist, wird als erwachsen akzeptiert.

29.FV1.10:Hundert ne?
Paraphrase:
Die Verkäuferin fragt, ob sie hundert Gramm Pfifferlinge abwiegen soll.
Intention1:
Die Verkäuferin will endlich zum Verkaufsabschluß kommen.
Regulative Regel1:
-
Konstitutive Regel1:
-
Intention2:
Die Verkäuferin merkt, daß die Käuferin ihre eigene Zurechnungsfähigkeit eingefordert hat und quittiert dies mit einer sachlichen Frage der der gewünschten Menge Pfifferlinge.
Regulative Regel2:
Die Zurechnungsfähigkeit eines anderen kann nicht ungestraft in Zweifel gezogen werden, wenn die andere Person zurechnungsfähig ist.
Konstitutive Regel2:
Erwachsene Personen fordern gegenseitig Zurechnungsfähigkeit ein.

30.FK1.12:Ja bitte. Watt krisch ich denn noch hier?
Paraphrase:
Die Käuferin sagt, das sie hundert Gramm Pfifferlinge will und fragt, was sie denn an diesem Stand noch bekommen kann.
Intention1:
Die Menge Pfifferlinge wird bestätigt und die Verkäuferin nach weiteren aktuellen Angeboten gefragt.
Regulative Regel1:
-
Konstitutive Regel1:
-
Intention2:
Die Menge Pfifferlinge wird bestätigt und die Käuferin fragt sich selbst, ob sie an diesem Verkaufsstand weitere Kaufbedürfnisse befriedigen kann.
Regulative Regel2:
-
Konstitutive Regel2:
-

31.FV1.11:Waldbeeren? Hab ich auch schonn.
Paraphrase:

Die Verkäuferin macht auf ihr Waldbeerangebot aufmerksam und betont, daß sie neben anderen Waren auch schon diese Ware anbieten kann.

Intention1:

Die Verkäuferin macht ein weiteres Kaufangebot und bekräftigt die Qualität ihres Angebotes mit dem Hinweis, daß sie saisonbedingt noch knappe Ware schon im Angebot hat. Sie ist eine gute Verkäuferin an einem hervorragenden Verkaufsstand.

Regulative Regel1:

-

Konstitutive Regel1:

Käufer kaufen besonders gerne bei guten Verkäufern an hervorragenden Verkaufsständen.

32.FK1.13:(unverständlich) Wie ist es denn mit Erdbeeren?

Paraphrase:
Die Käuferin fragt, ob die Verkäuferin Erdbeeren im Angebot hat.
Intention1:
Die Käuferin möchte keine Waldbeeren, sondern Erdbeeren.
Regulative Regel1:
-
Konstitutive Regel1:
-

33. (Unverständlich)

34.FK1.14:Watt hann se denn sons noch?

Paraphrase:
Die Käuferin fragt nach weiteren Angeboten.
Intention1:
Die Käuferin hat ihren Wunsch nach Erdbeeren aufgegeben, zum Beispiel, weil keine Erdbeeren im Angebot sind oder die angebotenen Erdbeeren im Vergleich zu ihrem Preis ihren Vorstellungen nicht mehr entsprechen. Unsicher, ob ein konkretes Angebot gegeben ist, fragt sie zunächst nach Alternativen.
Regulative Regel1:
-
Konstitutive Regel1:
-

35.FV1.12:Hann se denn keine Lust auf Himbeeren? Oder Johannisbeeren, hab ich auch schonn.

Paraphrase:
Die Verkäuferin bietet Himbeeren oder Johannisbeeren an. Sie weist darauf hin, daß sie trotz saisonbedingter Knappheitz beide Angebote machen kann.
Intention1:
Die "Sache mit den Erdbeeren" ist ungünstigt für das Ansehen. Die Verkäuferin will diese Scharte durch ein attraktives Angebot ausmerzen.
Regulative Regel1:
"Blöffen ist alles".
Konstitutive Regel1:
"Wer zögert, wird ausgetrixt".

36.FK1.15:Ja. (Pause) Nehm werr beides eins.

Paraphrase:
Die Käuferin sagt, daß sie sowohl Himbeeren, als auch Johannisbeeren nehmen will.
Intention1:
Die Käuferin will das günstige Angebot annehmen und gleich beide Sorten nehmen.
Regulative Regel1:
-
Konstitutive Regel1:
-

Intention2:
Die Käuferin will sich nicht erklären und nimmt ausweichend beide Angebote an.
Regulative Regel:

-
Konstitutive Regel:
-

37.FV1.13:Johannisbeeren is a Pfund, die können se auch noch länger verwahren.
Paraphrase:
Die Verkäuferin macht darauf aufmerksam, daß die Johannisbeeren im Pfund verpackt sind und länger aufbewahrt werden können.
Intention1:
Die Verkäuferin bedauert, daß sie Johannisbeeren nur im Pfund anbieten kann und beabsichtigt, diesen Mangel mit dem Hinweis auf ihre bessere Haltbarkeit auszugleichen.
Regulative Regel1:
-
Konstitutive Regel1:
-

38. (Pause, unverständlich)

39.FK1.16:Dann habb ich, ‚jlaub ich, alles fürr ze Hause.
Paraphrase:
Die Käuferin stellt fest, daß sie alles, was sie für Daheim kaufen wollte, erworben hat.
Intention1:
Die Käuferin will signalisieren, daß sie ihre Kaufwünsche an diesem Verkausstand befriedigt hat und nun den Kauf beenden möchte
Regulative Regel1:
Wer einen Kauf beenden will, muß das auch sagen.
Konstitutive Regel1:
Ein Kaufhandlung endet an einem bestimmten Punkt.

40.FV1.14:Joh, bis Übbermo‚jen, näh.
Paraphrase:
Die Verkäuferin bestätigt das Ende und verweist auf den übermorgigen Tag.
Intention1:
Die Verküferin will die Käuferin verabschieden.
Regulative Regel1:
Ein Verkäufer ist freundlich.
Konstitutive Regel1:
Verkäufer sind freundlich.
Intention2:
Die Verkäuferin bestätigt das Ende und verweist darauf, daß die Einkäufe bis übermorgen ausreichen, sie will Kundenbindung erreichen.
Regulative Regel2:
Verkäufer versuchen Kundenbindung zu erreichen

Konstitutive Regel2:
Kundenbindung erhöht den Erfolg eines Verkaufsgespräches.

41.FK1.17:neh.(Pause) Kuck mal, der junge Mann muß für Euch sorgen.
Paraphrase:
Die Käuferin macht auf eine männliche Person, die jünger als sie selbst ist, aufmerksam, die aus der Sicht der Käuferin eine für die Verkäuferin positive Handlung ohne erwartbare Gegenleistung vollzieht.
Intention1:
Die Käuferin will auf ein sie verblüffendes Ereignis hinweisen.
Regulative Regel1:
Man läßt andere an unerwarteten Beobachtungen teilhaben.
Konstitutive Regel1:
Es erhöht den Lebensgenuß, mit anderen Erfahrungen zu teilen.

42.FV1.15:Ja, damit uns de, de , eh
Paraphrase:
Die Verkäuferin bestätigt und sucht nach einer Begründung für die dem Interpreten unbekannte Handlung.
Intention1:
Am Ende des Verkaufsgespräches möchte die Verkäuferin die gute Atmosphäre aufrechterhalten und geht darum auf den Hinweis der Käuferin ein.
Regulative Regel1:
Verkäufer sind freundlich.
Konstitutive Regel1:
Verkäufer sind freundlich.

43.FK1.18:Ja
Paraphrase:
Die Käuferin bestätigt.
Intention1:
Die Käuferin will die Verkäuferin ermutigen, ihre Erklärung für die Handlung des jungen Mannes auszusprechen.
Regulative Regel1:
Man sollte im Gespräch bleiben, man weiß nie, was es einem nützt.
Konstitutive Regel1:
Nur wer im Gespräch bleibt, ist dabei.

44.FV1.16:(unverständlich) damit uns ett Jehirrn nett ahfängt zu koche.
Paraphrase:
Die Verkäuferin erklärt, daß die Handlung des jungen Mannes verhindert, die Gehirne der Verkäufer zum Kochen zu bringen.
Intention1:
Sie will die Handlung des jungen Mannes (vielleicht das Aufspannen eines Sonnensegels) erklären.
Regulative Regel1:

Handlungen mussen erklärt werden können.
Konstitutive Regel1:
Unerklärliche Handlungen darf es nicht geben, weil sie Unsicherheit verursachen.

45.FK1.19:So.
Paraphrase:
Die Käuferin sagt: So.
Intention1:
Die Käuferin will wieder zur Sache kommen.
Regulative Regel1:
Verkaufgespräche müssen zum Ende gebracht werden.
Konstitutive Regel1:
Endlose Verkaufsgespräche zerstören der Fortgang der Dinge.

46.FV1.17:Sechzig, vier Mark sechzig, acht Mark sechzig, zwölf Mark un fünfzig.
Paraphrase:
Die Verkäuferin rechnet den Preis aus und errechnet ein Ergebnis von zwölf Mark und fünfzig Pfennig.
Intention1:
Die Verkäuferin will auch zur Sache kommen und tut dies mit der Benennung des Endbetrages.
Regulative Regel1:
Verkäufer fordern einen Preis in der Währung des Landes.
Konstitutive Regel1:
Verkaufsgespräche sind nur erfolgreich abgeschlossen, wenn es zur Zahlung kommt.

47.FK1.20:Du kriss die Tür nich zu.
Paraphrase:
Die Käuferin bringt ihr Erstaunen über die Summe zum Ausdruck.
Intention1:
Die Käuferin will sagen, daß ihr der Preis zu hoch erscheint.
Regulative Regel1:
Käufer versuchen, den Preis zu drücken.
Konstitutive Regel1:
Marktpreise entstehen im Wechselspiel zwischen Forderung und Zahlungsbereitschaft.
Intention2:
Die Käuferin will sagen, daß sie erstaunt ist, wieviel sie gekauft hat.
Regulative Regel2:
-
Konstitutive Regel2:
-

48.FV1.18:Zwölf Mark un Fünfzisch. (Pause) Ich weiß, ich bin heut wieder unverschämt...
Paraphrase:
Die Verkäuferin wiederholt den Preis und bezeichnet sich

als unverschämt.
Intention1:
Die Verkäuferin bekräftigt ihre Forderung, schickt aber hinterher, daß der Preis tatsächlich hoch sei.
Regulative Regel1:
-
Konstitutive Regel1:
-
Intention2:
Die Verkäuferin bekräftigt ihre Forderung und ironisiert den Einwand der Käuferin.
Regulative Regel2:
-
Konstitutive Regel2:
-

49.FK1.21:Ja.
Paraphrase:
Die Käuferin sagt: ja
Intention1:
Die Käuferin bestätigt den Preis.
Regulative Regel1:
- Ein Kaufvertrag kommt zustande durch Preiseinigung.
Konstitutive Regel1:
- Über Preise muß man sich einigen.
Intention2:
Die Käuferin bestätigt, daß die Verkäuferin eine unverschämte Preisforderung stellt.
Regulative Regel2:
Man muß sich wehren.
Konstitutive Regel2:
-

50.FV1.19:Aber, aber, aber, eine Mark (unverständlich) noch.
Paraphrase:
Die Verkäuferin beginnt mit einem Einwand, wechselt dann aber über zu einer Restforderung.
Intention1:
Der Einwand ist offensichtlich überflüssig geworden, weil die Käuferin wohl mit der Zahlung begonnen hat, es fehlt aber mindestens noch eine Mark, die von der Verkäuferin von der Käuferin verlangt wird.
Regulative Regel1:
Preise müssen vollständigt bezahlt werden.
Konstitutive Regel1:
Nur nach vollständiger Bezahlung kommt ein Kaufvertrag zustande.

51.FK1.22:Hör ens
Paraphrase:
Die Käuferin fordert die Verkäuferin auf, ihr zuzuhören.
Sie ist wieder in die Du- Form gewechselt.
Intention1:
Die Käuferin kündigt eine wichtige Mitteilung an die

Verkäuferin als Mitmensch an.
Regulative Regel1:
Neuigkeiten teilt man Freunden und Bekannten mit, man bezieht sie ein.
Konstitutive Regel1:
-
Intention2:
Etwas Unangenehmes soll mitgeteilt werden, man fordert aber eine freundschaftliche Gefälligkeit ein.
Regulative Regel2:
-
Konstitutive Regel2:
-

52.FV1.20:Watt müssen se?
Paraphrase:
Die Verkäuferin fragt in der Sie- Form nach, worum es geht.
Intention1:
Die Verkäuferin will wissen, worum es geht und unterstreicht persönliche Distanz.
Regulative Regel1:
-
Konstitutive Regel1:
-

53.FK1.23:Zur eh Barmer, aber ich komm dann, ich komm dann nachher, dann stell ich et unter.
Paraphrase:
Die Käuferin teilt mit, daß sie noch zur Krankenkasse gehen muß und nicht weiß, was sie in der Zwischenzeit mit der gekauften Ware tun soll.
Intention1:
Die Käuferin bittet um Hilfe.
Regulative Regel1:
-
Konstitutive Regel1:
-
54.FV1.21:neh, sons lassen se et hier. Dreizehn, fünfzhen, Zwanzig Mark.
55.FK1.24:Danke.
Paraphrase:
Die Verkäuferin bietet der Käuferin an, die Ware für den Krankenkassenbesuch am Stand zu verwahren und gibt Wechselgeld zurück.
Intention1:
Man beendet den Kauf korrekt und freundlich.
Regulative Regel1:
-
Konstitutive Regel1:
-

56.FV1.22:Bis Übermorgen.
57.FK1.25:Danke schön.
Paraphrase:

Man verabschiedet sich gegenseitig freundlich.
Intention1:
Beide Gesprächspartnerinnen wollen einen mögliche späteren Kontakt offen halten.
Regulative Regel1:
- Verkäufer bieten Kundenbindung an. Käufer halten sich eine Tür auf.
Konstitutive Regel1:
- Verkäufer sind freundlich.

58.FV1.23:Ja
Paraphrase:
Die Verkäuferin bekräftigt den Abschied und Dank der Kundin.
Intention1:
siehe 57.
Regulative Regel1:
siehe 57.
Konstitutive Regel1:
siehe 57.

59.*** Ende Text4 ***
Paraphrase:
Der Text ist beendet
Intention1:
Das Ende des Textes wird markiert.
Regulative Regel1:
Man markiert das Ende eines Textes.
Konstitutive Regel1:
Texte haben eine endliche Länge.

7/10/95

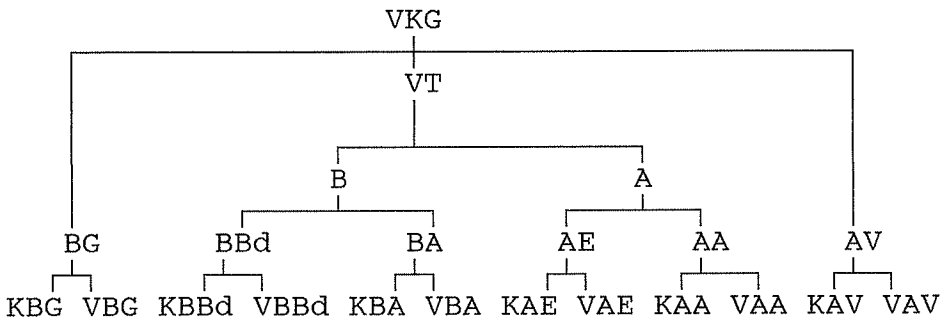
Paul Koop
Josefstraße 152
0241/558369
52080 Aachen

Kodierungsbogen

Algorithmisch rekursive Sequenzanalyse von Verkaufsgesprächen.

Bitte kennzeichnen Sie die Sequenzstellen nach folgendem Schema.
Für Ihre Mithilfe bedanke ich mich[1].

: Typ 1 VKG allgemein



---

[1]

Verkaufgespräch	:= VKG
Verkaufstätigkeit	:= VT
Bedarfsteil	:= B
Abschlußteil	:= A
Begrüßung	:= BG
Bedarf	:= Bd
Bedarfsargumentation	:= BA
Abschlußeinwände	:= AE
Verkaufsabschluß	:= AA
Verabschiedung	:= AV
vorangestelltes K	:= Kunde
vorangestelltes V	:= Verkäufer

1.*** Anfang Text4 ***

2.: Markt, 11.00 Uhr (Aachen, 28.06.94, Gemüsestand)

3. (unverständlich)

KBBd 4.FK1.1: Hör ens, ich nehm ein paar Champignons mit.

KBBd 5.FV1.1: Eh, braune?

KBBd 6.FK1.2: Ne helle.

KBBd 7.FV1.2: Helle.

KBBd 8.FK1.3: mhmh.

9. (unverständlich)

KBBd 10.FK1.4: Meinen se nich.

KBBd 11.FV1.3: Ja is ejal, se sinn beide frisch.

KBBd 12.FK1.5: Oder, wie is et denn mit, mit, eh

KBBd 13.FV1.4: Die können se länger liejen lassen.

KBBd 14.FK1.6: Neh, aber Pfifferlinge.

KBBd 15.FV1.5: Ah, die sinn super.

16. (unverständlich)

KBA 17.FK1.7: Kann ich die denn in Reissalat tun?

18. (Unverständlich)

KBA 19.FK1.8: Brauch ich nich abzukochen oder was?

KBA 20.FV1.6: Ehh, roh, doch müssen se en bischen in de Pfanne tun.

KBA 21.FK1.9: Tuh ich.

KBA 22.FV1.7: Klein bischen.

23. (unverständlich)

KBA 24.FK1.10:Die kann ich aber, ehm, in en Reissalat tun.

KBA 25.FV1.8: Ja, datt is kein Problem, se müssen so nur...

KBA 26.FK1.11:bischen, ja

KBA 27.FV1.9: Bischen in eh, nitt wie de Champignons, die tuh ich ja auch roh erein.

KBA 28.FK1.11:eh ja.

KBA 29.FV1.10:Hundert ne?

KBBd 30.FK1.12:Ja bitte. Watt krisch ich denn noch hier.

KBBd 31.FV1.11:Waldbeeren? Hab ich auch schonn.

KBBd 32.FK1.13:(unverständlich) Wie ist es denn mit Erdbeeren?

33. (Unverständlich)

KBBd 34.FK1.14:Watt hann se denn sons noch?

KBBd 35.FV1.12:Hann se denn keine Lust auf Himbeeren? Oder

Johannisbeeren, hab ich auch schonn.

KBBd 36.FK1.15:Ja. (Pause) Nehm werr beides eins.

VBA 37.FV1.13:Johannisbeeren is a Pfund, die können se auch noch länger verwahren.

38. (Pause, unverständlich)

KBA 39.FK1.16:Dann habb ich, jlaub ich, alles fürr ze Hause.

VBA 40.FV1.14:Joh, bis Übbermojen, näh.

41.FK1.17:neh.(Pause) Kuck mal, der junge Mann muß für Euch sorgen.

42.FV1.15:Ja, damit uns de, de , eh

43.FK1.18:Ja

44.FV1.16:(unverständlich) damit uns ett Jehirrn nett ahfängt zu koche.

KAE 45.FK1.19:So.

VAE 46.FV1.17:Sechzig, vier Mark sechzig, acht Mark sechzig, zwölf Mark un fünfzig.

KAE 47.FK1.20:Du kriss die Tür nich zu.

VAE 48.FV1.18:Zwölf Mark un Fünfzisch. (Pause) Ich weiß, ich bin heut wieder unverschämt...

KAA 49.FK1.21:Ja.

VAA 50.FV1.19:Aber, aber, aber, eine Mark (unverständlich) noch.

KAA 51.FK1.22:Hör ens

VAA 52.FV1.20:Watt müssen se?

KAA 53.FK1.23:Zur eh Barmer, aber ich komm dann, ich komm dann nachher, dann stell ich et unter.

VAA 54.FV1.21:neh, sons lassen se et hier. Dreizehn, fünfzhen, Zwanzig Mark.

KAA 55.FK1.24:Danke.

VAV 56.FV1.22:Bis Übermorgen.

KAV 57.FK1.25:Danke schön.

VAV 58.FV1.23:Ja

59.*** Ende Text4 ***

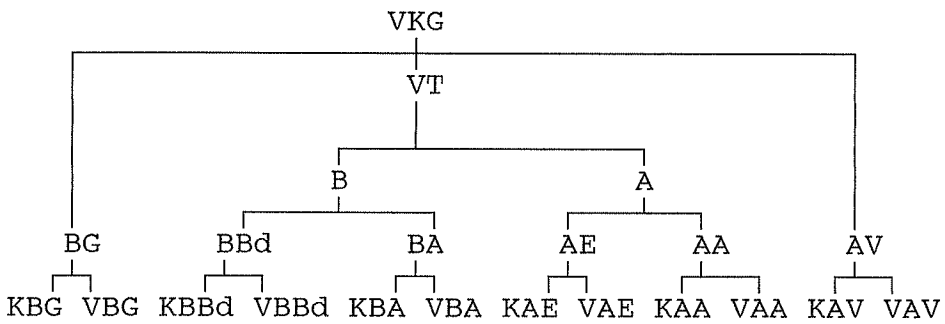
Paul Koop
Josefstraße 152
0241/558369
52080 Aachen

Kodierungsbogen

Algorithmisch rekursive Sequenzanalyse von Verkaufsgesprächen.

Bitte kennzeichnen Sie die Sequenzstellen nach folgendem Schema.
Für Ihre Mithilfe bedanke ich mich[1].

: Typ 1 VKG allgemein



---

[1]

Verkaufgespräch	:= VKG
Verkaufstätigkeit	:= VT
Bedarfsteil	:= B
Abschlußteil	:= A
Begrüßung	:= BG
Bedarf	:= Bd
Bedarfsargumentation	:= BA
Abschlußeinwände	:= AE
Verkaufsabschluß	:= AA
Verabschiedung	:= AV
vorangestelltes K	:= Kunde
vorangestelltes V	:= Verkäufer

1.*** Anfang Text4 ***

2.: Markt, 11.00 Uhr (Aachen, 28.06.94, Gemüsestand)

3. (unverständlich)

4.FK1.1: Hör ens, ich nehm ein paar Champignons mit.

5.FV1.1: Eh, braune?

6.FK1.2: Ne helle.

7.FV1.2: Helle.

8.FK1.3: mhmh.

9. (unverständlich)

10.FK1.4: Meinen se nich.

11.FV1.3: Ja is ejal, se sinn beide frisch.

12.FK1.5: Oder, wie is et denn mit, mit, eh

13.FV1.4: Die können se länger liejen lassen.

14.FK1.6: Neh, aber Pfifferlinge.

15.FV1.5: Ah, die sinn super.

16. (unverständlich)

17.FK1.7: Kann ich die denn in Reissalat tun?

18. (Unverständlich)

19.FK1.8: Brauch ich nich abzukochen oder was?

20.FV1.6: Ehh, roh, doch müssen se en bischen in de Pfanne tun.

21.FK1.9: Tuh ich.

22.FV1.7: Klein bischen.

23. (unverständlich)

24.FK1.10:Die kann ich aber, ehm, in en Reissalat tun.

25.FV1.8: Ja, datt is kein Problem, se müssen so nur...

26.FK1.11:bischen, ja

27.FV1.9: Bischen in eh, nitt wie de Champignons, die tuh ich ja auch roh erein.

28.FK1.11:eh ja.

29.FV1.10:Hundert ne?

30.FK1.12:Ja bitte. Watt krisch ich denn noch hier.

31.FV1.11:Waldbeeren? Hab ich auch schonn.

32.FK1.13:(unverständlich) Wie ist es denn mit Erdbeeren?

33. (Unverständlich)

34.FK1.14:Watt hann se denn sons noch?

35.FV1.12:Hann se denn keine Lust auf Himbeeren? Oder

Johannisbeeren, hab ich auch schonn.

36.FK1.15:Ja. (Pause) Nehm werr beides eins.

37.FV1.13:Johannisbeeren is a Pfund, die können se auch noch länger verwahren.

38. (Pause, unverständlich)

39.FK1.16:Dann habb ich, jlaub ich, alles fürr ze Hause.

40.FV1.14:Joh, bis Übbermojen, näh.

41.FK1.17:neh.(Pause) Kuck mal, der junge Mann muß für Euch sorgen.

42.FV1.15:Ja, damit uns de, de , eh

43.FK1.18:Ja

44.FV1.16:(unverständlich) damit uns ett Jehirrn nett ahfängt zu koche.

45.FK1.19:So.

46.FV1.17:Sechzig, vier Mark sechzig, acht Mark sechzig, zwölf Mark un fünfzig.

47.FK1.20:Du kriss die Tür nich zu.

48.FV1.18:Zwölf Mark un Fünfzisch. (Pause) Ich weiß, ich bin heut wieder unverschämt...

49.FK1.21:Ja.

50.FV1.19:Aber, aber, aber, eine Mark (unverständlich) noch.

51.FK1.22:Hör ens

52.FV1.20:Watt müssen se?

53.FK1.23:Zur eh Barmer, aber ich komm dann, ich komm dann nachher, dann stell ich et unter.

54.FV1.21:neh, sons lassen se et hier. Dreizehn, fünfzhen, Zwanzig Mark.

55.FK1.24:Danke.

56.FV1.22:Bis Übermorgen.

57.FK1.25:Danke schön.

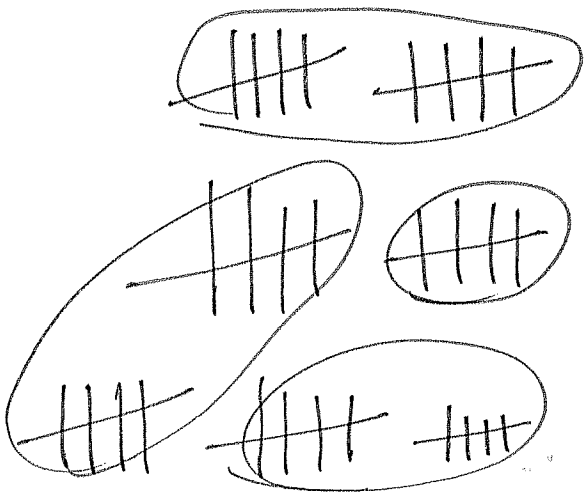
58.FV1.23:Ja

59.*** Ende Text4 ***

48.FV1.18:Zwölf Mark un Fünfzisch. (Pause) Ich weiß, ich bin heut wieder unverschämt...
49.FK1.21:Ja.
50.FV1.19:Aber, aber, aber, eine Mark (unverständlich) noch.
51.FK1.22:Hör ens

52.FV1.20:Watt müssen se?
53.FK1.23:Zur eh Barmer, aber ich komm dann, ich komm dann nachher, dann stell ich et unter.
54.FV1.21:neh, sons lassen se et hier. Dreizehn, fünfzhen, Zwanzig Mark.
55.FK1.24:Danke.
56.FV1.22:Bis Übermorgen.
57.FK1.25:Danke schön.
58.FV1.23:Ja
59.*** Ende Text4 ***

$$\frac{8VV}{0t} = 69'0$$



1.*** Anfang Text4 ***
2.: Markt, 11.00 Uhr (Aachen, 28.06.94, Gemüsestand)
3. (unverständlich)
4.FK1.1: Hör ens, ich nehm ein paar Champignons mit.
5.FV1.1: Eh, braune?
6.FK1.2: Ne helle.
7.FV1.2: Helle.
8.FK1.3: mhmh.
9. (unverständlich)
10.FK1.4: Meinen se nich.
11.FV1.3: Ja is ejal, se sinn beide frisch.
12.FK1.5: Oder, wie is et denn mit, mit, eh
13.FV1.4: Die können se länger liejen lassen.
14.FK1.6: Neh, aber Pfifferlinge.
15.FV1.5: Ah, die sinn super.
16. (unverständlich)
17.FK1.7: Kann ich die denn in Reissalat tun?
18. (Unverständlich)
19.FK1.8: Brauch ich nich abzukochen oder was?
20.FV1.6: Ehh, roh, doch müssen se en bischen in de Pfanne tun.
21.FK1.9: Tuh ich.
22.FV1.7: Klein bischen.
23. (unverständlich)
24.FK1.10:Die kann ich aber, ehm, in en Reissalat tun.
25.FV1.8: Ja, datt is kein Problem, se müssen so nur...
26.FK1.11:bischen, ja
27.FV1.9: Bischen in eh, nitt wie de Champignons, die tuh ich ja auch roh erein.
28.FK1.11:eh ja.
29.FV1.10:Hundert ne?
30.FK1.12:Ja bitte. Watt krisch ich denn noch hier.
31.FV1.11:Waldbeeren? Hab ich auch schonn.
32.FK1.13:(unverständlich) Wie ist es denn mit Erdbeeren?
33. (Unverständlich)
34.FK1.14:Watt hann se denn sons noch?
35.FV1.12:Hann se denn keine Lust auf Himbeeren? Oder Johannisbeeren, hab ich auch schonn.
36.FK1.15:Ja. (Pause) Nehm werr beides eins.
37.FV1.13:Johannisbeeren is a Pfund, die können se auch noch länger verwahren.
38. (Pause, unverständlich)
39.FK1.16:Dann habb ich, jlaub ich, alles fürr ze Hause.
40.FV1.14:Joh, bis Übbermojen, näh.
41.FK1.17:neh.(Pause) Kuck mal, der junge Mann muß für Euch sorgen.
42.FV1.15:Ja, damit uns de, de , eh
43.FK1.18:Ja
44.FV1.16:(unverständlich) damit uns ett Jehirrn nett ahfängt zu koche.
45.FK1.19:So.
46.FV1.17:Sechzig, vier Mark sechzig, acht Mark sechzig, zwölf Mark un fünfzig.
47.FK1.20:Du kriss die Tür nich zu.
48.FV1.18:Zwölf Mark un Fünfzisch. (Pause) Ich weiß, ich bin heut wieder unverschämt...
49.FK1.21:Ja.
50.FV1.19:Aber, aber, aber, eine Mark (unverständlich) noch.
51.FK1.22:Hör ens

Bedarf

52.FV1.20:Watt müssen se?
53.FK1.23:Zur eh Barmer, aber ich komm dann, ich komm dann nachher, dann stell ich et unter.
54.FV1.21:neh, sons lassen se et hier. Dreizehn, fünfzhen, Zwanzig Mark.
55.FK1.24:Danke.
56.FV1.22:Bis Übermorgen.
57.FK1.25:Danke schön.
58.FV1.23:Ja
59.*** Ende Text4 ***

4. FK1.1 Kaufwunsch Champignons

Problemlösung, helfen 10. FK1.4
11. FV1.3

29. FV.10 Problemlösung, helfen
Rat geben, Warenbieten

30. FK1.12
Ware anpreisen
Problem lösen helfen

40. FV1.14

41. FK1.17 Zwischenspiel
45. FK1.19.

46. FV1.17
Preisfindung Zahlung
erfolgreicher Abschluß Verkaufsgespräch
50. FV1.19

51. FK1.22
Wunsch Kundenproblem
& Geldrückgabe
55. FV1.24

56 Dank & freundliche Verabschiedung

1-2 Anfang Text 4 Markt Gemüsestand

3 Kontaktaufnahme (* nicht protokolliert *)

4-30 Bedarfsanalyse & Argumentation

31-40 Bedarfsanalyse & Argumentation

41-44 Zwischenspiel

45-50 Einwandentkräftung

51-55 Service Kaufbestätigung & Kundbindung

56-58 Verabschiedung

59 Ende Text

K: Guten Tag
V: Guten Tag

K: Ich hatte gerne Champignons
V: braune oder helle
K: helle, oder nicht
V: Daß ist egal, die hellen sind haltbarer
K: Nein Pfifferlinge, ~~ka~~
V: Die sind super
K: Kann ich die in Reißsalat tun
V: ja aber erwärmen

K: Was haben sie sonst noch
V: Waldbeeren
K: Erdbeeren
V: Johannis & Himbeeren
K: bordeos

V: 12,50
K: Teuer
V: Unverschämt bin ich

K: ~~12,50~~ 20,
V: 7,50 zurück

K: Danke auf wiedersehen
V: Danke auf wiedersehen

W3_syntax:

W3 → Markt

Struktursyntax

Markt → Verkaufsgespräch

Transsyntax

Verkaufsgespräch → B A

B → G Bd E

A → K V

G → KG VG

Bd → KB VB

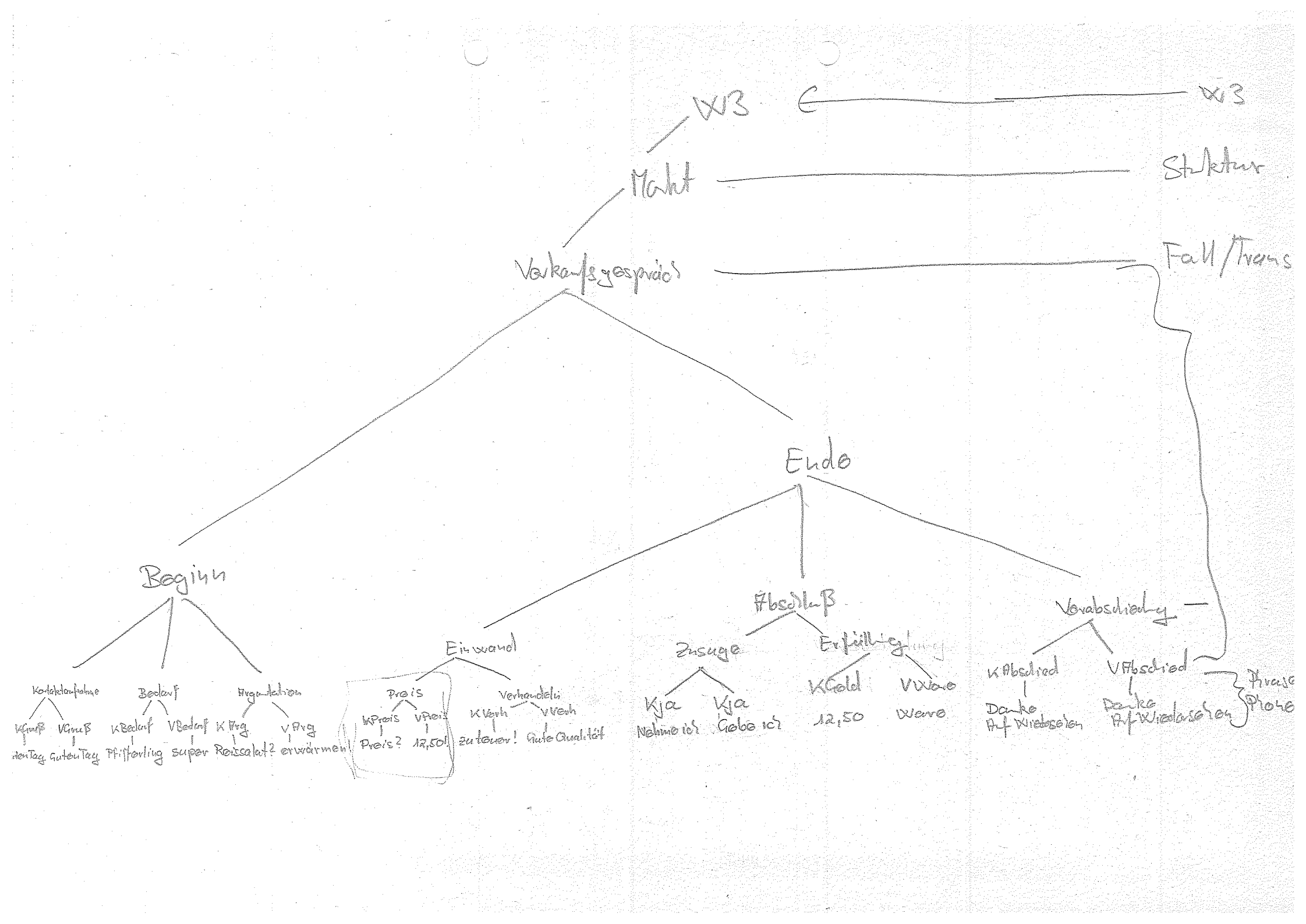
E → KE VE

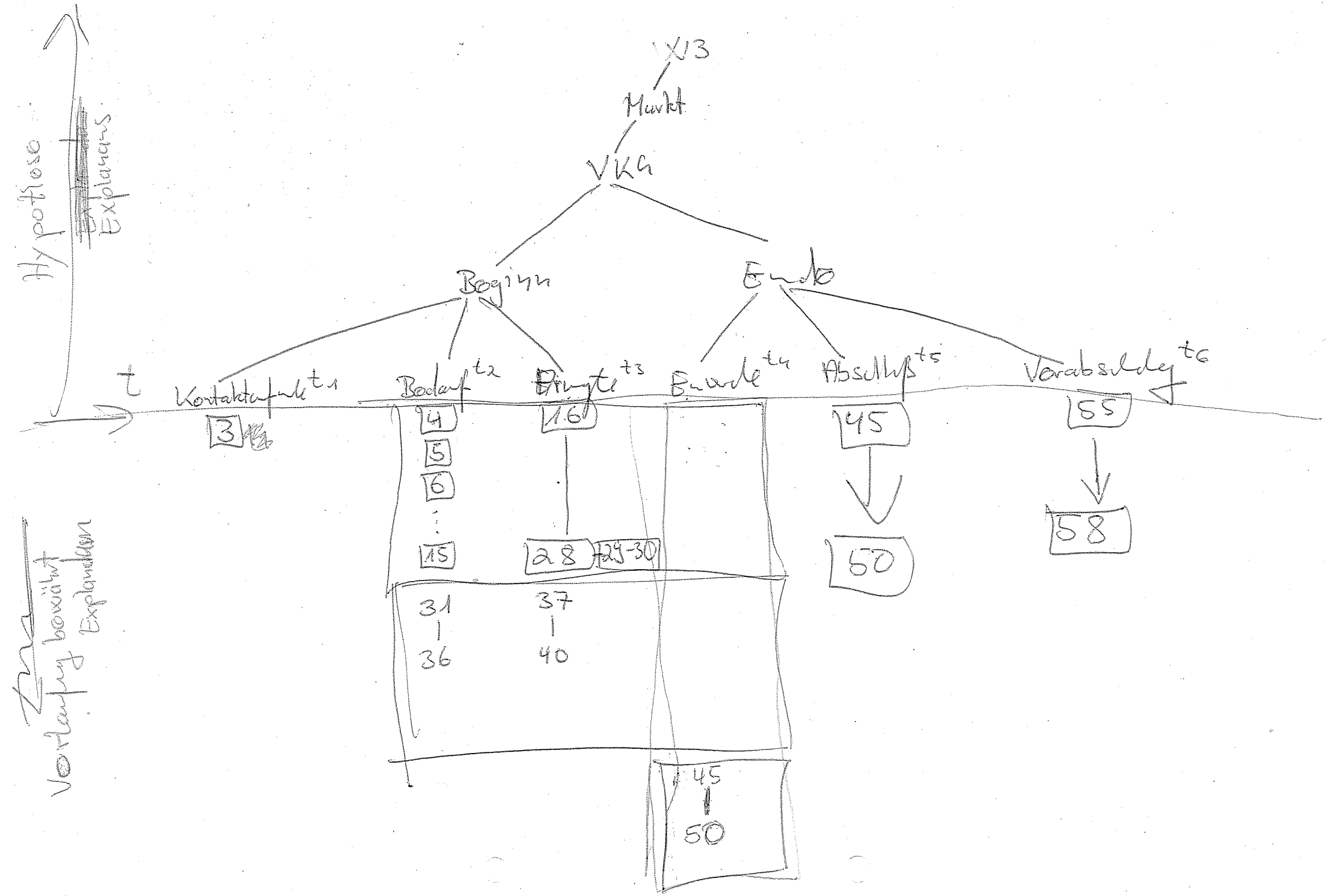
K → KK VK

V → KV VV

Phrase syntax

(x KG VG KB VB KE VE KK VK KV VV x)


WB
WB
Markt
Struktur
Verkaufsgespräch
Fall/Trans
Beginn
Ende
Kontaktaufnahme
Bedarf
Argumentation
KGruß
VGruß
KBedarf
VBedarf
K Frg
V Frg
denTag
GutenTag
Pfifferling
super
Reissalat?
erwärmen
Einwand
Preis
KPreis
VPreis
Preis?
12,50!
Verhandeln
KVerh
VVerh
zu teuer!
Gute Qualität
Abschluß
Zusage
Erfüllung
Kja
Kja
Nehme ich
Gebe ich
KGeld
VWare
12,50
Ware
Verabschiedung
KAbschied
VAbschied
Danke
Auf Wiedersehen
Danke
Auf Wiedersehen
Frage
Phase


Hypothese
Explanans
Vorhersagen bezüglich Explanandum
t
VU3
Markt
VKG
Beginn
Ende
Kontaktaufnahme t1
Bedarf t2
Eingang t3
Erwerb t4
Abschluß t5
Verabschiedung t6
3
4
5
6
15
16
28
29-30
31
36
37
40
45
50
45
50
55
58

VW3 → Markt

Markt → VKG

VKG → Beginn Ende

Beginn → Kontakt Bedarf Angebot

Ende → Einwände Abschluß Versuch

Beginn → GK GV

GK S

S NP VP

NP Art SoNomen

Art Ich

Nom Kaufer

VP V Art

V Gufbo

Art Dich

1-3 Textmarker Anfang

3 Begrüßung & Kontaktaufnahme

4-15 Bedarfsanalyse

16-30 Bedarfsargumente

31-36 Bedarfsanalyse

37-40 Bedarfsargumente

41-44 Ø Zwischenspiel

45-48 Abschlußeinwände

~~49-50 Verkaufsabschluß~~

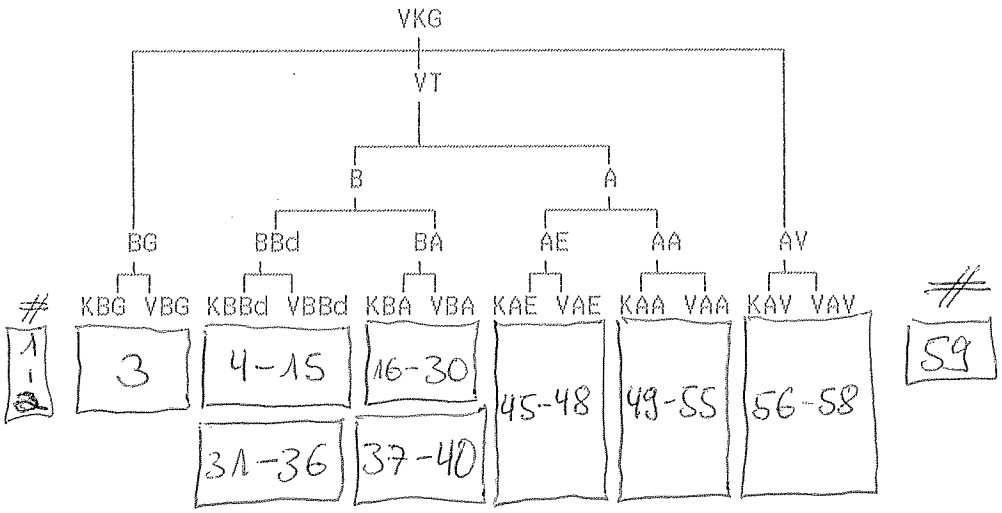
49-55 Verkaufsabschluß

56-58 Verabschiedung

59 Ende

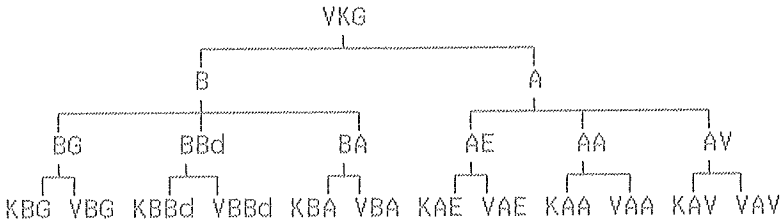
1

Verkaufgespräch	:= VKG
Verkaufstätigkeit	:= VT
Bedarfsteil	:= B
Abschlußteil	:= A
Begrüßung	:= BG
Bedarf	:= Bd
Bedarfsargumentation	:= BA
Abschlußeinwände	:= AE
Verkaufsabschluß	:= AA
Verabschiedung	:= AV
vorangestelltes K	:= Kunde
vorangestelltes V	:= Verkäufer



1

Verkaufgespräch	:= VKG
Verkaufstätigkeit	:= VT
Bedarfsteil	:= B
Abschlußteil	:= A
Begrüßung	:= BG
Bedarf	:= Bd
Bedarfsargumentation	:= BA
Abschlußeinwände	:= AE
Verkaufsabschluß	:= AA
Verabschiedung	:= AV
vorangestelltes K	:= Kunde
vorangestelltes V	:= Verkäufer



1

Verkaufgespräch	:= VKG
Bedarfsteil	:= B
Abschlußteil	:= A
Begrüßung	:= BG
Bedarf	:= Bd
Bedarfsargumentation	:= BA
Abschlußeinwände	:= AE
Verkaufsabschluß	:= AA
Verabschiedung	:= AV
vorangestelltes K	:= Kunde
vorangestelltes V	:= Verkäufer

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Paul Koop M.A. GRAMMATIKINDUKTION empirisch                      ;;
;; gesicherter Verkaufsgespraeche                                   ;;
;;                                                                  ;;
;; Die Simulation wurde ursprunglich entwickelt,                    ;;
;; um die Verwendbarkeit von kontextfreien Grammatiken              ;;
;; fuer die Algorithmisch Rekursive Sequanzanalyse                  ;;
;; zu ueberpruefen                                                  ;;
;; Modellcharakter hat allein der Quelltext.                        ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;  __|__   ___|__    __|__   __|__   __|__   __|__                 ;;
;;  |   |   |    |    |   |   |   |   |   |   |   |                 ;;
;; KBG->VBGKBBd->VBBdKBA->VBAKAE->VAEKAA->VAAKAV-> VAV              ;;
;;                                                                  ;;
;; Die Produktionen --> sind entsprechend ihrer                     ;;
;; emp. Auftrittswahrscheinlichkeit gewichtet                       ;;
;; DIE GRAMMATIK WIRD AUS DEM KORPUS INDUZIERT                      ;;
;; ein Left-to-the-Right-Modell                                     ;;

;;                                                                  ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                  ;;
;; Begruessung           := BG                                      ;;
;; Bedarf                := Bd                                      ;;
;; Bedarfsargumentation  := BA                                      ;;
;; Abschlusseinwaende    := AE                                      ;;
;; Verkaufsabschluss     := AA                                      ;;
;; Verabscheidung        := AV                                      ;;
;; Kunde                 := vorangestelltes K                       ;;
;; Verkaeufer            := vorangestelltes V                       ;;
;;                                                                  ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; Korpus
   (define korpus (list 'KBG 'VBG 'KBBd 'VBBd 'KBA 'VBA 'KBBd 'VBBd 'KBA 'V
BA 'KAE 'VAE 'KAE 'VAE 'KAA 'VAA 'KAV 'VAV));; 0 - 17



   ;; Korpus durchlaufen
    (define (lesen korpus)
     ;; car ausgeben
```

```
    (display (car korpus))
     ;; mit cdr weitermachen
     (if(not(null? (cdr korpus)))
       (lesen (cdr korpus))
       ;;(else)
     )
   )

;; Lexikon
   (define lexikon (vector 'KBG 'VBG 'KBBd 'VBBd 'KBA 'VBA 'KAE 'VAE 'KAA '
VAA 'KAV 'VAV)) ;; 0 - 12



   ;; Index fuer Zeichen ausgeben
    (define (izeichen zeichen)
     (define wertizeichen 0)
     (do ((i 0 (+ i 1)))
      ( (equal? (vector-ref lexikon i) zeichen))
      (set! wertizeichen (+ 1 i))
     )
     ;;index zurueckgeben
     wertizeichen
   )

;; transformationsmatrix
   (define zeile0 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile1 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile2 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile3 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile4 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile5 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile6 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile7 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile8 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile9 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile10 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile11 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile12 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile13 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile14 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile15 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile16 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))
   (define zeile17 (vector 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0))

   (define matrix (vector zeile0 zeile1 zeile2 zeile3 zeile4 zeile5 zeile6
zeile7 zeile8 zeile9 zeile10 zeile11 zeile12 zeile13 zeile14 zeile15 zeile1
```

```
6 zeile17))


   ;; Transformationen zaehlen
      ;; Korpus durchlaufen
   (define (transformationenZaehlen korpus)
     ;; car zaehlen
      (vector-set! (vector-ref matrix (izeichen (car korpus))) (izeichen (c
ar(cdr korpus))) (+ 1 (vector-ref  (vector-ref matrix (izeichen (car korpus
))) (izeichen (car(cdr korpus))))))
     ;; mit cdr weitermachen
      (if(not(null? (cdr (cdr korpus))))
       (transformationenZaehlen (cdr korpus))
       ;;(else)
      )
   )


   ;; Transformation aufaddieren

   ;; Zeilensummen bilden und Prozentwerte bilden


;; Grammatik
   (define grammatik (list '- ))

   ;; aus matrix regeln bilden und regeln in grammatik einfügene
   (define (grammatikerstellen matrix)
    (do ((a 0 (+ a 1)))
        ((= a 12) )(newline)
      (do ((b 0 (+ b 1)))
          ((= b 12))
        (if (< 0 (vector-ref  (vector-ref matrix a) b) )
         (display (cons (vector-ref lexikon a) (cons '-> (vector-ref lexiko
n b))))
         )
      )
    )
   )


  ;; matrix ausgeben
   (define (matrixausgeben matrix)
    (do ((a 0 (+ a 1)))
        ((= a 12) ) (newline)
      (do ((b 0 (+ b 1)))
          ((= b 12))
```

```
      (display (vector-ref  (vector-ref matrix a) b))
    )
  )
)
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Paul Koop M.A. 1994 Sequenzanalyse empirisch                     ;;
;; gesicherter Verkaufsgespraeche                                   ;;
;;                                                                  ;;
;; Die Simulation wurde ursprunglich entwickelt,                    ;;
;; um die Verwendbarkeit von kontextfreien Grammatiken              ;;
;; fuer die Algorithmisch Rekursive Sequanzanalyse                  ;;
;; zu ueberpruefen                                                  ;;
;; Modellcharakter hat allein der Quelltext.                        ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                  ;;
;;                        VKG                                       ;;
;;    _____________________|_____________________                   ;;
;;    |                    |                    |                   ;;
;;    BG------------------>VT------------------>AV                  ;;
;;    |             _______|________            |                   ;;
;;    |             |              |            |                   ;;
;;    |             B------------->A            |                   ;;
;;    |        _____|____       ___|_____       |                   ;;
;;    |        |        |       |       |       |                   ;;
;;    |       BBd----->BA      AE----->AA       |                   ;;
;;  __|__   ___|__    __|__   __|__   __|__   __|__                 ;;
;;  |   |   |    |    |   |   |   |   |   |   |   |                 ;;
;; KBG->VBGKBBd->VBBdKBA->VBAKAE->VAEKAA->VAAKAV-> VAV              ;;
;;                                                                  ;;
;; Die Produktionen --> sind entsprechend ihrer                     ;;
;; emp. Auftrittswahrscheinlichkeit gewichtet                       ;;
;; Waehrend die Kanten des Strukturbaumes ein Top-down-Modell       ;;
;; wiedergeben, bilden die Produktionen                             ;;
;; des Kategoriensystem-Systems (K-System)                          ;;
;; ein Left-to-the-Right-Modell                                     ;;

;;                                                                  ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                  ;;
;; Verkaufsgespraech      := VKG                                    ;;
;; Verkaufstaetigkeit     := VT                                     ;;
;; Bedarfsteil            := B                                      ;;
;; Abschlussteil          := A                                      ;;
;; Begruessung            := BG                                     ;;
;; Bedarf                 := Bd                                     ;;
;; Bedarfsargumentation   := BA                                     ;;
;; Abschlusseinwaende     := AE                                     ;;
```

```
;; Verkaufsabschluss     := AA                                   ;;
;; Verabscheidung        := AV                                   ;;
;; Kunde                 := vorangestelltes K                    ;;
;; Verkaeufer            := vorangestelltes V                    ;;
;;                                                               ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                               ;;
;; - Die Fallstruktur wird rein physikalisch protokolliert       ;;
;;   mechanisch, magnetisch, optisch oder digital D/A-Wandler    ;;
;;   (interpretationsfreies physikalisches Protokoll)            ;;
;;   z.B. Mikrophonierung, Kinematographie,                      ;;
;;   Optik, Akustik, mechanische, analoge, digitale Technik      ;;
;; - Das Protokoll wird transkribiert                            ;;
;;   (Vertextung, diskrete Ereigniskette,                        ;;
;;    Plausibilitaet, Augenscheinvalidität)                      ;;
;;   Searle, Austin: Sprechakte, Paraphrase, moegl.              ;;
;;   Intentionen, konstitutive, konventionelle Regeln            ;;
;; - Durch Lesartenproduktion und Lesartenfalsifikation          ;;

;;   wird Sequenzstelle fuer Sequenzstelle informell             ;;
;;   das Regelsystem erzeugt                                     ;;
;;   Searle, Austin: Sprechakte, Paraphrase, moegl.              ;;
;;   Intentionen, konstitutive, konventionelle Regeln            ;;
;;   (bei jeder Sequenzstelle werden extensiv Lesarten erzeugt,  ;;
;;    die Lesarten jeder nachfolgenden Sequenzstelle             ;;
;;    falsifizieren die Lesarten der vorausgehenden Sequenzstelle,;;
;;    Oevermann: Sequenzanalyse                                  ;;
;;    das Regelsystem bildet ein kontextfreies Chomskysystem,    ;;
;;    die Ersetzungsregeln sind nach Auftrittswahrscheinlichkeit ;;
;;    gewichtet, die Interkodierreliabilitaet wird bestimmt,     ;;
;;    z.B. Mayring R, Signifikanz z.B. Chi-Quadrat)              ;;
;; - Die Regeln werden in ein K-System uebersetzt                ;;
;;   dabei werden die Auftrittshaeufigkeiten kumuliert           ;;
;;   um den Rechenaufwand zur Laufzeit zu minimieren             ;;
;;   Chomsky: formale Sprachen                                   ;;

;; - Auf einem Computer wird unter LISP eine Simulation gefahren ;;
;;   McCarthy, Papert, Solomon, Bobrow, Feuerzeig
;; - Das Resultat der Simulation, eine terminale Zeichenkette,   ;;
;;   wird in ein Protokoll uebersetzt                            ;;
;; - Das künstlich erzeugte Protokoll wird auf seine Korrelation ;;

;;   mit empirischen Protokollen ueberprueft                     ;;

;; - Bei Bedarf werden Korrekturen am K-System vorgenommen       ;;
```

```
;;   und die Simulation wird wiederholt                               ;;
;;                                                                    ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                    ;;
;; Welt 3 Popper                                                      ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


(setq w3
'(
 (anfang 100 (s vkg)) ;; hier nur Fallstruktur Verkaufsgespraeche
 ((s vkg) 100 ende)
 )
)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                    ;;
;; Kunde teilt Bedarf mit, Verkaeufer spiegelt Bedarf Kunde           ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


(setq bbd
'(
 (kbbd 100 vbbd)
 )
)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                    ;;
;; wechselseitige Bedarfsargumentation nach Bedarfsmitteilung         ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


(setq ba
'(
 (kba 100 vba)
 )
)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                    ;;
;; wechselseitige Einwandsabklaerung                                  ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
(setq ae
'(
(kae 100 vae)
 )
)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                ;;
;; Verkaufsabschluss                                              ;;
;; des Abschlussteils nach den Abschlusseinwaenden                ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


(setq aa
'(
 (kaa 100 vaa)
 )
)


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                ;;
;; Bedarfsteils                                                   ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(setq b
'(
 ((s bbd) 100 (s ba))
 )
)


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                ;;
;; Abschlussteil                                                  ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


(setq a
'(
 ((s ae)50(s ae))
 ((s ae)100(s aa))
 )
)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
;;                                                                ;;
;; Verkaufsteil                                                   ;;
;; im Anschluss an Begruessung                                    ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(setq vt
'(
 ((s b)50(s b))
 ((s b)100(s a))
 )
)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                ;;
;; Begruessung                                                    ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(setq bg
'(
 (kbg 100 vbg)
 )
)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                ;;
;; Verabschiedung                                                 ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


(setq av
'(
 (kav 100 vav)
 )
)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                ;;
;; Verkaufsgespraech                                              ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


(setq vkg
'(
 ((s bg)100(s vt))
 ((s vt)50(s vt))
 ((s vt)100(s av))
 )
```

```
)


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                ;;
;; Algorithmus ueber generativer Struktur                         ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;; Generiert die Sequenz
(defun gs (st r);; Uebergabe Sequenzstelle und Regelliste
(cond

  ;; gibt nil zurück, wenn das Sequenzende ereicht ist
  ((equal st nil) nil)

  ;; gibt terminale Sequenzstelle mit Nachfolgern zurueck
  ((atom st)(cons st(gs(next st r(random 101))r)))

  ;; gibt expand. nichtterm. Sequenzstelle mit Nachfolger zurueck
  (t (cons(eval st)(gs(next st r(random 101))r)))
)
)

;; Generiert nachfolgende Sequenzstelle
(defun next (st r z);; Sequenzstelle, Regeln und Haeufigkeitsmass
(cond

  ;; gibt nil zurueck, wenn das Sequenzende erreicht ist
  ((equal r nil)nil)

  ;; waehlt Nachfolger mit Auftrittsmass h
  (
    (
      and(<= z(car(cdr(car r))))
      (equal st(car(car r)))
    )
   (car(reverse(car r)))
  )

  ;; in jedem anderen Fall wird Regelliste weiter durchsucht
  (t(next st (cdr r)z))
)
)

;; waehlt erste Sequenzstelle aus Regelliste
;;vordefinierte funktion first wird ueberschrieben, alternative umbenennen
```

```
(defun first (list)
(car(car list))
)

;; startet Simulation fuer eine Fallstruktur
(defun s (list) ;; die Liste mit dem K-System wird uebergeben
(gs(first list)list)
)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                ;;
;; Ruft den Algorithmus auf / Welt 3 Popper /alt. jew. Fallstrukt.;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; alternativ (s vkg) / von der Konsole aus (s w3) oder (s vkg)
(s w3)
```

```
PROGRAM parser (INPUT,OUTPUT);
USES CRT;
(************************************************************************
*)
(* Paul Koop Chart Parser VKG
*)
(*
*)
(************************************************************************
*)

  (*----------------------------------------------------------------------
*)
  (* Vereinbarungsteil
*)
  (*----------------------------------------------------------------------
*)

  CONST
    c0              =     0;
    c1              =     1;
    c2              =     2;
    c3              =     3;
    c4              =     4;
    c5              =     5;
   c10              =    10;
   c11              =    11;
   cmax             =    80;
   cwort            =    20;
   CText            :    STRING(.cmax.) = '';
   datei            =    'LEXIKONVKG.ASC';
   blank            =    ' ';

   CopyRight
   =    'Demo-Parser Chart-Parser Version 1.0(c)1992 by Paul Koop';

  TYPE
   TKategorien       = ( Leer, VKG, BG, VT, AV, B, A, BBD, BA, AE, AA,
                         KBG, VBG, KBBD, VBBD, KBA, VBA, KAE, VAE,
                         KAA, VAA, KAV, VAV);


   PTKategorienListe = ^TKategorienListe;
   TKategorienListe  = RECORD
                        Kategorie :TKategorien;
                        weiter    :PTKategorienListe;
                       END;
```

```
PTKante           = ^TKante;
PTKantenListe     = ^TKantenListe;

TKantenListe      = RECORD
                     kante:PTKante;
                     next :PTKantenListe;
                    END;

TKante            = RECORD
                     Kategorie :TKategorien;
                     vor,
                     nach,
                     zeigt     :PTKante;
                     gefunden  :PTKantenListe;
                     aktiv     :BOOLEAN;
                     nummer    :INTEGER;
                     nachkomme :BOOLEAN;
                     CASE  Wort:BOOLEAN OF
                      TRUE :
                          (inhalt:STRING(.cwort.););
                      FALSE:
                          (gesucht :PTKategorienListe;);
                     END;


TWurzel    = RECORD
               spalte,
               zeigt     :PTKante;
              END;

TEintrag    = RECORD
              A,I   :PTKante;
              END;

PTAgenda    = ^TAgenda;
TAgenda     = RECORD
               A,I  :PTKante;
               next,
               back : PTAgenda;
              END;

PTLexElem   = ^TLexElem;
TLexElem    = RECORD
               Kategorie: TKategorien;
               Terminal : STRING(.cwort.);
               naechstes: PTLexElem;
```

```
                END;

  TGrammatik  = ARRAY (.c1..c10.)
                OF
                ARRAY (.c1..c4.)
                OF TKategorien;
 CONST
  Grammatik :      TGrammatik =
             (
              (VKG, BG,      VT,    AV),
              (BG,  KBG,     VBG,   Leer),
              (VT,  B,       A,     Leer),
              (AV,  KAV,     VAV,   Leer),
              (B,   BBd,     BA,    Leer),
              (A,   AE,      AA,    Leer),
              (BBd, KBBd,    VBBd,  Leer),
              (BA,  KBA,     VBA,   Leer),
              (AE,  KAE,     VAE,   Leer),
              (AA,  KAA,     VAA,   Leer)
             );

 nummer :INTEGER = c0;

 (*-----------------------------------------------------------------
*)
 (* Variablen
*)
 (*-----------------------------------------------------------------
*)


 VAR
  Wurzel,
  Pziel       : TWurzel;
  Pneu        : PTKante;

  Agenda,
  PAgenda,
  Paar        : PTAgenda;

  LexWurzel,
  LexAktuell,
  LexEintrag  : PTLexElem;
  Lexikon     : Text;


(*****************************************************************
```

```
*)
(* FUNKTIONEN
*)
(*************************************************************************
*)


  (*-----------------------------------------------------------------------
*)
  (* KantenZaehler
*)
  (*-----------------------------------------------------------------------
*)

  FUNCTION NimmNummer:INTEGER;
   BEGIN
    Nummer := Nummer + c1;
    NimmNummer := Nummer
   END;



(*************************************************************************
*)
(* PROZEDUREN
*)
(*************************************************************************
*)




  (*-----------------------------------------------------------------------
*)
  (* LexikonLesen
*)
  (*-----------------------------------------------------------------------
*)

  PROCEDURE LiesDasLexikon (VAR f:Text;
                            G:TGrammatik;
                            l:PTLexElem);
    VAR
     zaehler :INTEGER;
     z11     : 1..c11;
     z4      : 1.. c4;
     ch      :   CHAR;
```

```
  st5     : STRING(.c5.);

BEGIN
 ASSIGN(f,datei);
 LexWurzel := NIL;
 RESET(f);
 WHILE NOT EOF(f)
  DO
   BEGIN
    NEW(LexEintrag);
    IF LexWurzel = NIL
     THEN
      BEGIN
       LexWurzel := LexEintrag;
       LexAktuell:= LexWurzel;
       LexEintrag^.naechstes := NIL;
      END
     ELSE
      BEGIN
       LexAktuell^.naechstes := LexEintrag;
       LexEIntrag^.naechstes := NIL;
       LexAktuell            := LexAktuell^.naechstes;
      END;
    LexEintrag^.Terminal := '';
    st5 := '';
    FOR Zaehler := c1 to c5
     DO
      BEGIN
       READ(f,ch);
       st5 := st5 + UPCASE(ch)
      END;
    REPEAT
     READ(f,ch);
     LexEintrag^.terminal := LexEintrag^.Terminal + UPCASE(ch);
    UNTIL EOLN(f);
    READLN(f);
    IF st5 = 'KBG**' THEN  LexEintrag^.Kategorie := KBG    ELSE
    IF st5 = 'VBG**' THEN  LexEintrag^.Kategorie := VBG    ELSE
    IF st5 = 'KBBD*' THEN  LexEintrag^.Kategorie := KBBD   ELSE
    IF st5 = 'VBBD*' THEN  LexEintrag^.Kategorie := VBBD   ELSE
    IF st5 = 'KBA**' THEN  LexEintrag^.Kategorie := KBA    ELSE
    IF st5 = 'VBA**' THEN  LexEintrag^.Kategorie := VBA    ELSE
    IF st5 = 'KAE**' THEN  LexEintrag^.Kategorie := KAE    ELSE
    IF st5 = 'VAE**' THEN  LexEintrag^.Kategorie := VAE    ELSE
    IF st5 = 'KAA**' THEN  LexEintrag^.Kategorie := KAA    ELSE
    IF st5 = 'VAA**' THEN  LexEintrag^.Kategorie := VAA    ELSE
    IF st5 = 'KAV**' THEN  LexEintrag^.Kategorie := KAV    ELSE
```

```
      IF st5 = 'VAV**' THEN  LexEintrag^.Kategorie := VAV
     END;
  END;


 (*-----------------------------------------------------------------------
*)
 (* SatzLesen
*)
 (*-----------------------------------------------------------------------
*)

 PROCEDURE LiesDenSatz;
  VAR
   satz:        STRING(.cmax.);
   zaehler:     INTEGER;
  BEGIN
   CLRSCR;
   WRITELN(CopyRight);
   WRITE('-----> ');
   Wurzel.spalte := NIL;
   Wurzel.zeigt  := NIL;
   READLN(satz);
   FOR zaehler := c1 to LENGTH(satz)
    DO satz(.zaehler.) := UPCASE(satz(.zaehler.));
   Satz := Satz + blank;
   Writeln('-----> ',satz);
   WHILE satz <> ''
   DO
   BEGIN
      NEW(Pneu);
      Pneu^.nummer   :=NimmNummer;
      Pneu^.wort     := TRUE;
      NEW(Pneu^.gefunden);
      Pneu^.gefunden^.kante := Pneu;
      pneu^.gefunden^.next  := NIL;
      Pneu^.gesucht         := NIL;
      Pneu^.nachkomme       :=FALSE;
      IF Wurzel.zeigt = NIL
       THEN
        BEGIN
          Wurzel.zeigt := pneu;
          Wurzel.spalte:= pneu;
          PZiel.spalte := pneu;
          PZiel.zeigt  := Pneu;
          pneu^.vor    := NIL;
          Pneu^.zeigt  := NIL;
```

```
          Pneu^.nach   := NIL;
         END
        ELSE
         BEGIN
          Wurzel.zeigt^.zeigt := Pneu;
          Pneu^.vor           := Wurzel.zeigt;
          Pneu^.nach          := NIL;
          Pneu^.zeigt         := NIL;
          Wurzel.zeigt        := Wurzel.zeigt^.zeigt;
         END;
       pneu^.aktiv   := false;
       pneu^.inhalt  := COPY(satz,c1,POS(blank,satz)-c1);
       LexAktuell    := LexWurzel;
       WHILE LexAktuell <> NIL
        DO
         BEGIN
          IF LexAktuell^.Terminal = pneu^.inhalt
           Then
            BEGIN
             pneu^.Kategorie := LexAktuell^.Kategorie;
            END;
          LexAktuell := LexAktuell^.naechstes;
         END;
       DELETE(satz,c1,POS(blank,satz));
      END;
   END;




  (*-----------------------------------------------------------------------
*)
  (* Regel3KanteInAgendaEintragen
*)
  (*-----------------------------------------------------------------------
*)

  PROCEDURE Regel3KanteInAgendaEintragen (Kante:PTKante);
   VAR
    Wurzel,
    PZiel  :TWurzel;
   PROCEDURE NeuesAgendaPaarAnlegen;
    BEGIN
     NEW(paar);
     IF Agenda = NIL
      THEN
       BEGIN
```

```
      Agenda := Paar;
      Pagenda:= Paar;
      Paar^.next := NIL;
      Paar^.back := NIL;
     END
    ELSE
     BEGIN
      PAgenda^.next := Paar;
      Paar^.next    := NIL;
      Paar^.back    := Pagenda;
      Pagenda       := Pagenda^.next;
    END;
  END;

BEGIN
 IF Kante^.aktiv
  THEN
   BEGIN
    Wurzel.zeigt := Kante^.zeigt;
    WHILE wurzel.zeigt <> NIL
     DO
     BEGIN
      IF NOT(wurzel.zeigt^.aktiv)
       THEN
        BEGIN
         NeuesAgendaPaarAnlegen;
         paar^.A := kante;
         paar^.I := wurzel.zeigt;
        END;
     Wurzel.zeigt  := Wurzel.zeigt^.nach
     END
   END
  ELSE
  BEGIN
    PZiel.zeigt  := Kante;
    WHILE NOT(PZiel.zeigt^.Wort)
     DO PZiel.Zeigt := PZiel.Zeigt^.Vor;
    Wurzel.Zeigt    := PZiel.Zeigt;
    Wurzel.Spalte   := PZiel.Zeigt;
    PZiel.Spalte    := Pziel.zeigt;
    WHILE wurzel.spalte <> NIL
     DO
     BEGIN
      WHILE wurzel.zeigt <> NIL
      DO
      BEGIN
       IF wurzel.zeigt^.aktiv
```

```
          AND (Wurzel.zeigt^.zeigt = PZiel.spalte)
          THEN
           BEGIN
            NeuesAGendaPaarAnlegen;
            paar^.I := kante;
            paar^.A := wurzel.zeigt;
           END;
         Wurzel.zeigt  := Wurzel.zeigt^.nach
        END;
        wurzel.spalte  := wurzel.spalte^.vor;
        wurzel.zeigt   := wurzel.spalte;
       END
      END
     END;

  (*-----------------------------------------------------------------------
*)
  (* AgendaAusgabe
*)
  (*-----------------------------------------------------------------------
*)

  PROCEDURE NimmAgendaEintrag(VAR PEintrag:PTAgenda);
   BEGIN
      IF PAgenda = Agenda
      THEN
       BEGIN
        PEintrag := Agenda;
        PAgenda  := NIL;
        Agenda   := NIL;
       END
      ELSE
       BEGIN
        PAGENDA       := PAGENDA^.back;
        PEintrag      := PAgenda^.next;
        PAGENDA^.next := NIL;
       END;
   END;




  (*-----------------------------------------------------------------------
*)
  (* Regel2EineNeueKanteAnlegen
*)
  (*-----------------------------------------------------------------------
```

```
*)

  PROCEDURE Regel2EineNeueKanteAnlegen( Kante     :PTKante;
                                        Kategorie :TKategorien;
                                        Gram      :TGrammatik );
   VAR
     Wurzel             :TWurzel;
     PHilfe,
     PGesuchteKategorie :PTKategorienListe;
     zaehler,
     zaehler2           :INTEGER;

   BEGIN
   Wurzel.zeigt := Kante;
   Wurzel.spalte:= Kante;
   WHILE Wurzel.zeigt^.nach <> NIL
    DO Wurzel.zeigt := Wurzel.zeigt^.nach;
    FOR zaehler := c1 To c11
     DO
      IF  (kategorie = Gram(.zaehler,c1.))
      AND (kategorie <> Leer)
       THEN
       BEGIN
        Gram(.zaehler,c1.) := Leer;
        NEW(pneu);
        Wurzel.zeigt^.nach := pneu;
        pneu^.nummer       := NimmNummer;
        pneu^.vor          := Wurzel.zeigt;
        Pneu^.nach         := NIL;
        Pneu^.zeigt        := wurzel.spalte;
        Wurzel.zeigt       := Wurzel.zeigt^.nach;
        pneu^.aktiv        := true;
        pneu^.kategorie    := kategorie;
        Pneu^.Wort         := false;
        Pneu^.gesucht      := NIL;
        Pneu^.gefunden     := NIL;
        Pneu^.nachkomme    := FALSE;
        FOR zaehler2 := c2 TO c4
         DO
         BEGIN
          IF Gram(.zaehler,zaehler2.) <> Leer
           THEN
            BEGIN
             NEW(PGesuchteKategorie);
             PGesuchteKategorie^.weiter:= NIL;
             PGesuchteKategorie^.Kategorie := Gram(.zaehler,zaehler2.);
             IF Pneu^.gesucht = NIL
```

```
              THEN
               BEGIN
                PHilfe        := PGesuchteKategorie;
                Pneu^.gesucht := PHilfe;
               END
              ELSE
               BEGIN
                PHilfe^.weiter := PGesuchteKategorie;
                PHilfe         := PHilfe^.weiter;
               END
            END
         END;
        Regel3KanteInAgendaEintragen (pneu);
        Regel2EineNeueKanteAnlegen(Wurzel.spalte,
                                   pneu^.gesucht^.kategorie,gram);
      END;
   END;



  (*-----------------------------------------------------------------------
*)
  (* Regel1EineKanteErweiternen
*)
  (*-----------------------------------------------------------------------
*)

  PROCEDURE Regel1EineKanteErweitern(paar:PTAgenda);
   VAR
    PneuHilf,Pneugefneu,AHilf :PTKantenListe;
   BEGIN

   IF paar^.I^.kategorie = paar^.A^.gesucht^.kategorie
    THEN
     BEGIN
      NEW(pneu);
      pneu^.nummer      := NimmNummer;
      pneu^.kategorie   := Paar^.A^.kategorie;
(*-----------------------------------------------------*)
      Pneu^.gefunden := NIL;
      AHilf := Paar^.A^.gefunden;

      WHILE AHilf <> NIL
       DO
       BEGIN
        NEW(Pneugefneu);
        IF Pneu^.gefunden = NIL
```

```
      THEN
       BEGIN
        Pneu^.gefunden := Pneugefneu;
        PneuHilf       := Pneu^.gefunden;
        PneuHilf^.next := NIL;
       END
      ELSE
       BEGIN
        PneuHilf^.next   := Pneugefneu;
        PneuHilf         := PneuHilf^.next;
        PneuHilf^.next   := NIL;
       END;

     Pneugefneu^.kante     := AHilf^.kante;
     AHilf                 := AHilf^.next;
    END;

    NEW(Pneugefneu);
    IF Pneu^.gefunden = NIL
     THEN
      BEGIN
       Pneu^.gefunden := Pneugefneu;
       Pneugefneu^.next := NIL;
      END
     ELSE
      BEGIN
        PneuHilf^.next   := Pneugefneu;
        PneuHilf         := PneuHilf^.next;
        PneuHilf^.next   := NIL;
      END;
    Pneugefneu^.kante    := Paar^.I;
(*-------------------------------------------*)
    Pneu^.wort             := FALSE;
    IF Paar^.A^.gesucht^.weiter = NIL
     THEN Pneu^.gesucht   := NIL
     ELSE Pneu^.gesucht   := Paar^.A^.gesucht^.weiter;
    Pneu^.nachkomme := TRUE;

   IF pneu^.gesucht   = NIL
    THEN Pneu^.aktiv := false
    ELSE Pneu^.aktiv := true;

   WHILE Paar^.A^.nach <> NIL
    DO Paar^.A       := Paar^.A^.nach;

   Paar^.A^.nach     := pneu;
   pneu^.vor         := Paar^.A;
```

```
      pneu^.zeigt       := Paar^.I^.zeigt;
      pneu^.nach        := NIL;

      Regel3KanteInAgendaEintragen (pneu);
      IF Pneu^.aktiv
       THEN Regel2EineNeueKanteAnlegen(Pneu^.zeigt,
                                       pneu^.gesucht^.kategorie,Grammatik);
     END;


   END;
  (*-----------------------------------------------------------------------
*)
  (* SatzAnalyse
*)
  (*-----------------------------------------------------------------------
*)

   PROCEDURE SatzAnalyse;
    BEGIN
    WHILE Agenda <> NIL
    DO
     BEGIN
      NimmAgendaEintrag(Paar);
      Regel1EineKanteErweitern(Paar);
     END;

    END;
  (*-----------------------------------------------------------------------
*)
  (* SatzAusgabe
*)
  (*-----------------------------------------------------------------------
*)

   PROCEDURE GibAlleSatzalternativenAus;
    CONST
     BlankAnz:INTEGER = c2;
    VAR
     PHilf   :PTkantenListe;

    PROCEDURE SatzAusgabe(Kante:PTKante;BlankAnz:INTEGER);
     VAR

     Zaehler:INTEGER;
     PHilf  :PTKantenListe;
     BEGIN
```

```
  FOR Zaehler := c1 TO BlankAnz DO WRITE(blank);

  IF Kante^.kategorie = VKG     THEN WRITELN ('VKG ') ELSE
  IF Kante^.kategorie = BG      THEN WRITELN ('BG  ') ELSE
  IF Kante^.kategorie = VT      THEN WRITELN ('VT  ') ELSE
  IF Kante^.kategorie = AV      THEN WRITE   ('AV  ') ELSE
  IF Kante^.kategorie = B       THEN WRITELN ('B   ') ELSE
  IF Kante^.kategorie = A       THEN WRITE   ('A   ') ELSE
  IF Kante^.kategorie = BBD     THEN WRITE   ('BBD ') ELSE
  IF Kante^.kategorie = BA      THEN WRITELN ('BA  ') ELSE
  IF Kante^.kategorie = AE      THEN WRITE   ('AE  ') ELSE
  IF Kante^.kategorie = AA      THEN WRITE   ('AA  ') ELSE
  IF Kante^.kategorie = KBG     THEN WRITELN ('KBG ') ELSE
  IF Kante^.kategorie = VBG     THEN WRITELN ('VBG ') ELSE
  IF Kante^.kategorie = KBBD    THEN WRITELN ('KBBD') ELSE
  IF Kante^.kategorie = VBBD    THEN WRITE   ('VBBD') ELSE
  IF Kante^.kategorie = KBA     THEN WRITELN ('KBA ') ELSE
  IF Kante^.kategorie = VBA     THEN WRITE   ('VBA ') ELSE
  IF Kante^.kategorie = KAE     THEN WRITE   ('KAE ') ELSE
  IF Kante^.kategorie = VAE     THEN WRITELN ('VAE ') ELSE
  IF Kante^.kategorie = KAA     THEN WRITE   ('KAA ') ELSE
  IF Kante^.kategorie = VAA     THEN WRITE   ('VAA ') ELSE
  IF Kante^.kategorie = KAV     THEN WRITE   ('KAV ') ELSE
  IF Kante^.kategorie = VAV     THEN WRITE   ('VAV ');

  IF Kante^.wort
   THEN
    WRITELN('----> ',Kante^.inhalt)
   ELSE
    BEGIN
    PHilf := Kante^.gefunden;
    WHILE PHilf <> NIL
     DO
      BEGIN
       Satzausgabe(PHilf^.kante,Blankanz+c1);
       PHilf := Philf^.next;
      END
    END
END;

BEGIN
  WHILE Wurzel.zeigt^.vor <> NIL
   DO Wurzel.zeigt := Wurzel.zeigt^.vor;

  WHILE Wurzel.zeigt <> NIL
  DO
  BEGIN
```

```
      IF (Wurzel.zeigt^.kategorie = VKG)
        AND ((NOT(Wurzel.zeigt^.aktiv))
        AND (wurzel.zeigt^.zeigt = NIL))
        THEN
         BEGIN
          WRITELN('VKG');
          PHilf := Wurzel.zeigt^.gefunden;
          WHILE PHilf <> NIL
           DO
            BEGIN
             Satzausgabe(PHilf^.kante,Blankanz+c1);
             PHilf := Philf^.next;
            END
         END;
     Wurzel.zeigt := Wurzel.zeigt^.nach;
     END;

   END;

  (*-----------------------------------------------------------------------
*)
  (* FreigabeDesBenutztenSpeicherplatzes
*)
  (*-----------------------------------------------------------------------
*)

  PROCEDURE LoescheDieListe;
   PROCEDURE LoescheWort(kante :PTKante);
    PROCEDURE LoescheSpalte(kante:PTKante);
     VAR
      Pgefunden :PTKantenListe;
      Pgesucht  :PTKategorienListe;
     PROCEDURE LoescheGesucht(p:PTKategorienListe);
      BEGIN
       IF p^.weiter <> NIL
        THEN LoescheGesucht(p^.weiter);
       IF P <> NIL THEN DISPOSE(P);
      END;
     PROCEDURE LoescheGefunden(Kante:PTKante;p:PTKantenListe);
      BEGIN
       IF p^.next <> NIL
        THEN LoescheGefunden(Kante,p^.next);
       DISPOSE(P);
      END;
     BEGIN(*LoescheSpalte*)
      IF Kante^.nach <> NIL
       THEN LoescheSpalte(kante^.nach);
```

```
      IF (NOT Kante^.nachkomme) AND ((Kante^.gesucht <> NIL)
       AND (NOT Kante^.wort))
       THEN LoescheGesucht(Kante^.gesucht);
      IF Kante^.gefunden <> NIL
       THEN LoescheGefunden(Kante,Kante^.gefunden);
      DISPOSE(Kante)
     END;(*LoescheSpalte*)
    BEGIN(*LoescheWort*)
     IF Kante^.zeigt <> NIL
      THEN LoescheWort(Kante^.zeigt);
    LoescheSpalte(Kante);
    END;(*LoescheWort*)
   BEGIN(*LoescheDieListe*)
    WHILE Wurzel.spalte^.vor <> NIL
     DO Wurzel.spalte := Wurzel.spalte^.vor;
    LoescheWort(Wurzel.spalte);
   END;(*LoescheDieListe*)
(*************************************************************************
*)
(* HAUPTPROGRAMM DES CHART PARSERS
*)
(*************************************************************************
*)

  BEGIN
   Agenda := NIL;
   PAgenda := Agenda;
   LiesDasLexikon(Lexikon,Grammatik,LexWurzel);
   LiesDenSatz;
   WHILE Wurzel.spalte^.vor <> NIL
    DO Wurzel.spalte := Wurzel.spalte^.vor;
   Regel2EineNeueKanteAnlegen(Wurzel.spalte,VKG,Grammatik);
   SatzAnalyse;
   GibAlleSatzalternativenAus;
   LoescheDieListe;
(*************************************************************************
*)
(* ENDE DES HAUPTPROGRAMMS DES CHART PARSERS
*)
(*************************************************************************
*)

  END.
```

Literatur:

Alpaydin, E.: Maschinelles Lernen, 2008

Chomsky, N.: Aspects of the Theory of Syntax, 1965

Dehmer, Matthias: Strukturelle Analyse Web-basierter Dokumente, 2005

Diekmann,A.: Spieltheorie: Einführung, Beispiele, Experimente 2009

Gold, E. Mark: Limiting Recursion, The Journal of Symbolic Logic 30: 28–48. 1965

Gold, E. Mark: Language Identification in the Limit, Information and Control 10: 447–474, 1967

Koop,P.: Über die Entscheidbarkeit der GTG, 1994

Koop,P.: Rekursive Strukturen und Prozesse, 1995

Koop,P.: K-Systeme: Das Projekt ARS, 1994

Koop,P.: Algorithmisch Rekursive Sequenzanalyse, 1996

Koop,P.: Oevermann, Chomsky, Searle, 1994

Krauße, C. C., & Krueger,F.R.: Unbekannte Signale Spektrum Dossier 2/2002

Krempel, Rasmus: Netze, Karten, Irrgärten:Graphenbasierte explorative Ansätze zur Datenanalyse und Anwendungsentwicklung in den Geisteswissenschaften, 2016

Lisch, R., Kriz , J.: Grundlagen und Modelle der Inhaltsanalyse 1978

Mayring, P.: Einführung in die qualitative Sozialforschung 1990

Ndiaye, Alassane: Rollenübernahme als Benutzermodellierungsmethode : globale Antizipation in einem transmutierbaren Dialogsystem 1998

Nevill-Manning Witten: Identifying Hierarchical Structure in Sequences: A linear-time algorithm 1999

Oevermann,U: Die objektive Hermeneutik als unverzichtbare methodologische Grundlage für die Analyse von Subjektivität. Zugleich eine Kritik an der Tiefenhermeneutik, in: Jung,Th., Müller-Dohm, St. (Hg): >>Wirklichkeit<< im Deutungsprozess: Verstehen und Methoden in den Kultur- und Sozialwissenschaften, Frankfurt 1993

Shen, Chunze: EDSI - Effiziente Grammatikinduktion, 2013