Siegfried Wendt
Universität Kaiserslautern

Die Begriffswelt des Formalen

1. Der Zweck des Formalen

Es kommt immer wieder vor, daß jemand, der mit einer Entscheidung eines Verwaltungsbeamten oder Richters nicht einverstanden ist, den Vorwurf erhebt, die Entscheidung sei ausschließlich unter ”rein formalen” Gesichtspunkten gefällt worden und andere Gesichtspunkte seien gar nicht in Betracht gezogen worden. Damit will der Opponent sagen, daß einfach ”stur” nach dem Wortlaut der Gesetze oder Vorschriften verfahren worden sei. Ob hier der Opponent das Wort *formal* in der gleichen oder einer ähnlichen Bedeutung verwendet, wie sie auch in der Fachsprache der Mathematik, der Informatik oder der Philosophie gilt, ist eine Frage, die der Leser beantworten können sollte, nachdem er die folgenden Betrachtungen gelesen hat.

Die formalen Systeme entstanden in der Logik, also dem Zweig der Mathematik, in dem die Grundlagen des Beweisens untersucht werden. Sie sind ein Ergebnis des Bemühens der Logiker, mehr ”Objektivität” in die Kunst des Beweisens hineinzubringen. Ein mathematischer Beweis zu einer Behauptung b ist eine endliche Folge a_1, a_2, a_3, ...a_n von Aussagen, die von Mathematikern als ”wahr” anerkannt werden; die letzte Aussage a_n ist entweder gleich der Behauptung b, womit die Wahrheit von b bewiesen ist, oder gleich deren Verneinung, womit die Behauptung b widerlegt ist. Für jede einzelne Aussage a_i in der Folge a_1 bis a_n wird die Anerkennung ihrer Wahrheit auf eine von zwei möglichen Weisen begründet: Entweder stammt a_i aus einem apriori gegebenen Repertoire anerkannt wahrer Aussagen, oder aber die Wahrheit von a_i folgt logisch zwingend aus der Wahrheit der voranstehenden Aussagen a_1 bis a_{i-1}. Während die Vorgabe eines Repertoires anerkannt wahrer Aussagen als Voraussetzung jeglichen Beweisens von den Logikern ohne weiteres akzeptiert werden kann – denn wo sollte sonst die wahre Aussage a_1 herkommen? –, ist für einen Logiker keineswegs jede Schlußfolgerung akzeptabel, die andere Leute für logisch zwingend halten. Das Bemühen der Logiker nach mehr Objektivität war deshalb darauf gerichtet, genau zu definieren, was eine akzeptable logische Schlußfolgerung sei.

Da die größtmögliche Objektivität sicher erreicht wäre, wenn es gelänge, das logische Schließen auf einen reinen Formalismus zu reduzieren, entwickelte man die formalen Systeme. Jede Aussagenkette a_1 bis a_{i-1} in einer Beweiskette wird nur noch als Symbolfolge betrachtet, deren Interpretation außer Betracht bleiben kann, wenn die Schlußregeln formalisiert sind. Die Schlußregeln sind genau dann formalisiert, wenn jede Aussage a_i, die als Schlußfolgerung aus den voranstehenden Aussagen a_1 bis a_{i-1} akzeptiert werden soll, nach diesen Regeln ausschließlich unter Bezug auf die Form und nicht auf die Interpretation der Aussagen a_1 bis a_{i-1} gewonnen werden kann.

Ein solches formales System ist also ein geregeltes Spiel mit Formen – man könnte auch Muster oder Figuren sagen –, das auch von jemandem gespielt werden kann, für den die Formen bedeutungslos sind. Dennoch ist ein formales System nur dann von praktischem Interesse, wenn die Formen von jemandem interpretiert werden können; dies muß jedoch nicht der Spieler sein.

Die Erkenntnis, daß der Spieler und der Interpretierende unterschiedliche Personen sein dürfen, bildet den Ausgangspunkt für die weitere Erkenntnis, daß der Spieler gar kein Mensch sein muß, sondern eine Maschine sein kann. Von der Maschine wird nur verlangt, daß sie aus gegebenen Formen nach einfachen Regeln neue Formen erzeugen kann. Bei der Festlegung des Formenrepertoires wird man zweckmäßigerweise die Mustererkennungsfähigkeiten des jeweiligen Spielers berücksichtigen. Im Falle eines menschlichen Spielers wird man ein Repertoire von Schrift– und Formelzeichen wählen, dessen Elemente durch einfaches Hinsehen erkannt werden können. Im Falle eines maschinellen Spielers wird man dagegen als

Formen binäre Codewörter wählen, die man eins–zu–eins den Schrift– und Formelzeichen zuordnen kann. Binäre Codewörter lassen sich technisch leicht durch unterschiedliche physikalische Sachverhalte repräsentieren – z.B. als Lochmuster auf einem Papierstreifen oder einer Karte, als Spannungsverteilung auf einem Leitungsbündel oder als Magnetisierungsverteilung auf einer Magnetplatte. Durch geeignete Gestaltung des technischen Apparates kann man erreichen, daß die nach physikalischen Gesetzmäßigkeiten entstehende zeitliche Folge solcher Repräsentationen von Codewörtern dem gewünschten formalen Formenspiel entspricht. Jeder Computer ist ein solcher Apparat, worin ein Formenspiel abläuft, welches vom menschlichen Betreiber des Computers, nicht aber vom Computer selbst interpretiert wird.

> Der Zweck des Formalen ist die vollständige Objektivierung eines Teilbereichs geistiger Tätigkeit. Durch die Formalisierung wird die geistige Tätigkeit aufgeteilt in ein objektives, d.h. mechanisierbares und damit jederzeit nachvollziehbares Formenspiel und in die separate Interpretation der Formen.

Diese Objektivierung ist zum einen von großem Nutzen in der Mathematik, und zum anderen ist sie die Grundlage der Gestaltung von Computern und ihrer Programmierung.

2. Das Prinzip des Formalen

Im Abschnitt 1 wurde ein Formenspiel als ein Vorgang definiert, bei dem Formbausteine nach Regeln zu Formen zusammengesetzt werden und aus gegebenen Formen nach Regeln neue Formen erzeugt werden, wobei man bewußt auf jegliche Interpretation der Formen verzichtet. In Bild 1 ist schematisch ein Formenspiel mit vier Spielschritten dargestellt: Zu Beginn des Spiels ist eine initiale Formenmenge vorgegeben. Ob es mehrere alternativ vorgebbare Initialmengen gibt oder ob für jedes Spiel immer die gleiche Initialmenge genommen werden muß, wird durch die Spielregeln festgelegt.

Im ersten Spielschritt wird der Spieler nach eigenem Ermessen aus der Menge der gegebenen Formenkonstruktionsregeln eine auf die Initialmenge A_1 anwendbare Regel auswählen, die beschreibt, wie man auf der Grundlage der Initialmenge durch formale Konstruktion eine neue Form F_1 gewinnen kann. Am Ende des ersten Spielschritts hat man als Ausgangssituation für den zweiten Spielschritt eine Formenmenge A_2, die neben den initialen Formen zusätzlich noch die erste erzeugte Form F_1 enthält. Im zweiten Spielschritt wird aus der Menge der gegebenen Konstruktionsregeln eine auf die Formenmenge A_2 anwendbare Regel ausgewählt und angewendet, usw.. Vor jedem Spielschritt wird geprüft, ob die erreichte Formenmenge A_k bereits das formale Kriterium für das Spielende erfüllt. Dieses Kriterium muß neben den Konstruktionsregeln als Spielregel vorgegeben sein.

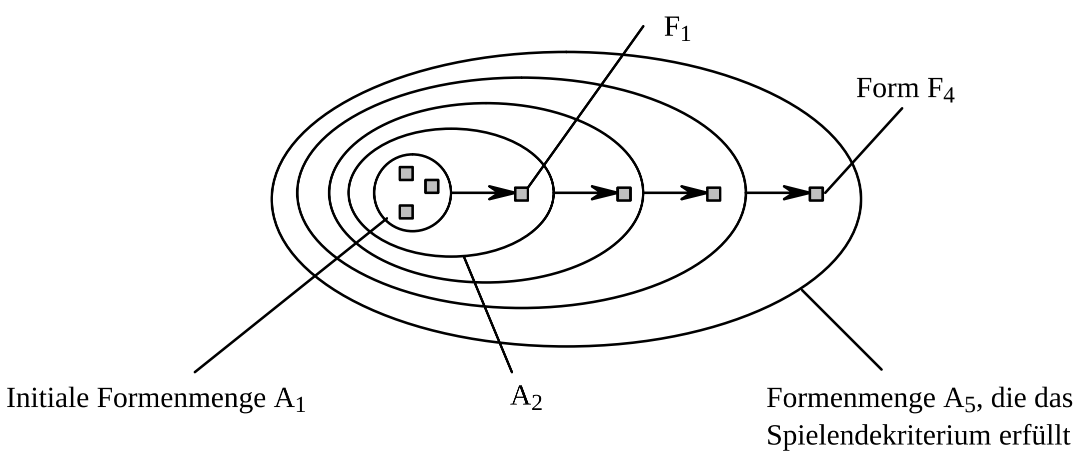


Bild 1 Formenspiel als Erzeugung einer Folge von Formenmengen

Zur Veranschaulichung der abstrakten Begriffe werden in Bild 2 zwei Beispiele von Formenspielen vorgestellt [Wendt–91]. Das Bild zeigt zwei verschiedene Repertoires von flächenhaften Formbausteinen, aus denen – wie beim Kacheln einer Wand – zusammengesetzte Formen gebildet werden können. Wegen der anschaulichen Darstellbarkeit wurden hier flächenhafte Formbausteine als Beispiele gewählt. Der Begriff des Formbausteins ist jedoch keinesfalls auf die Zweidimensionalität beschränkt.

Das Repertoire in Beispiel 1 enthält nur einen einzigen Bausteintyp, aber aus Bausteinen dieses Typs können zweidimensionale Formen aufgebaut werden, denn es gibt vier unterschiedliche Verbindungsarten zwischen zwei Bausteinen: Weißer Riegel ins weiße Feld, weißer Riegel ins schwarze Feld, schwarzer Riegel ins weiße Feld, schwarzer Riegel ins schwarze Feld. Dagegen enthält das Repertoire in Beispiel 2 zwei Bausteintypen, aber aus Bausteinen dieser Typen können nur eindimensionale Formen aufgebaut werden, da jeder Baustein jeweils nur einen Verbindungsriegel und eine dazu passende Aussparung besitzt. Die Menge K ist gleich der Menge der mit dem jeweiligen Typenrepertoire von Formbausteinen machbaren zusammenhängenden endlichen Formstrukturen.

Für die Beispiele in Bild 2 wurden keine Spielendekriterien festgelegt. In beiden Beispielen ist jeweils eine feste Initialmenge für den Spielbeginn vorgegeben.

In allen Fällen, in denen es nur eine einzige Initialmenge gibt, bezeichnet man die Formen in der Initialmenge als *Axiome*. Dabei muß die Bedingung der *Unabhängigkeit* erfüllt sein, daß kein Axiom aus den anderen Axiomen über einmalige oder mehrmalige Anwendungen von Ableitungsregeln gewonnen werden kann.

Man beachte, daß hier der Begriff des Axioms in einer abstrakteren Bedeutung verwendet wird als sie jahrhundertelang in der Mathematik üblich war. Im Kontext des Formalen sind Axiome lediglich bestimmte Formstrukturen; im traditionellen mathematischen Sprachgebrauch dagegen sind Axiome bestimmte Aussagen, die per Definition, d.h. ohne Beweis als wahr zu betrachten sind und aus denen alle damit implizierten wahren Aussagen formal abgeleitet werden können. Im Abschnitt 3.2 wird dieser Sonderfall von Axiomen, die als mathematische Aussagen interpretiert werden können, näher betrachtet.

In den Beispielen in Bild 2 sind die Formen gar nicht interpretierbar, so daß selbstverständlich hier in den Spielregeln nur die Form, aber nicht die Interpretation herangezogen werden kann. Aber auch in solchen Fällen, wo die Formen interpretierbar sind, darf in den Spielregeln – also in den Ableitungsregeln und im Spielendekriterium – die Interpretation nicht benutzt werden.

Unter den formalen Systemen haben nur wenige Sonderfälle eine praktische Bedeutung. Dies sind insbesondere diejenigen, bei denen nur eindimensionale Formen, also Folgen von Bausteinen vorkommen wie im Beispiel 2 in Bild 2. Auf eindimensionale Formen beschränkt sich nun die weitere Betrachtung.

In der Praxis geht es selbstverständlich nicht um reine Spielereien mit bedeutungslosen Formen, sondern um interpretierbare Symbolfolgen. Die betrachteten Symbolfolgen sind dann meist als Ausdrücke, Aussagen, Fragen oder Anweisungen interpretierbar. Im Abschnitt 3 werden Beispiele vorgestellt.

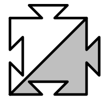
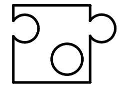
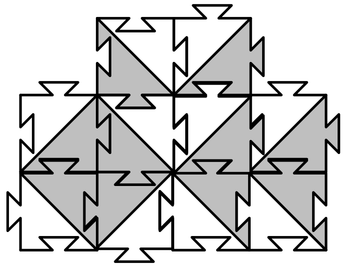
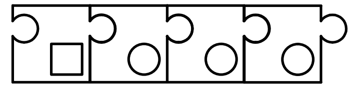
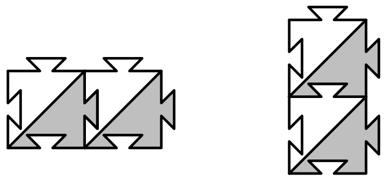
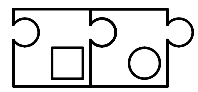
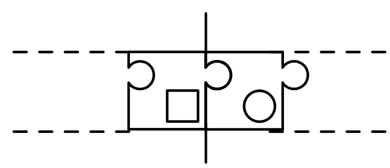
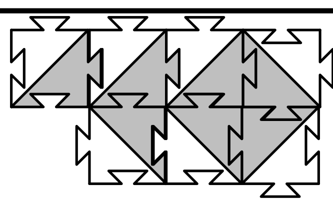
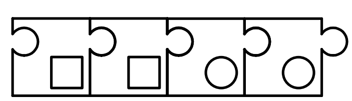
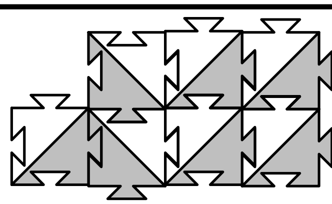
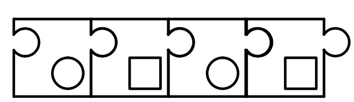
	Beispiel 1	Beispiel 2
Typenrepertoire der Formbausteine		
Beispiel einer zusammenge–setzten Form (Element von K)		
Vorgegebene Initialformen (Axiome)		
Regeln	(1) Aus der Kopie einer Initialform für den ersten Spielschritt bzw. von F_k für den (k+1)–ten Spielschritt kann man eine Form F_1 bzw. F_{k+1} bauen, indem man an diese Kopie einen Baustein derart anfügt, daß die beim Anschluß beteiligten Riegel jeweils in die Felder ihrer Farbe greifen. (2) Aus der Kopie einer Form F_k, bei der irgendwo am Rand ein auf zwei Bausteine verteiltes Dreieck seine Hypothenuse nach außen streckt, erhält man eine weitere Form, indem man an diese Hypothenuse mit zwei Bausteinen genau ein gleiches Dreieck parallelverschoben noch einmal anbaut.	Aus einer Kopie der Initialform für den ersten Spielschritt bzw. von F_k für den (k+1)–ten Spielschritt kann man eine Form F_1 bzw. F_{k+1} bauen, indem man diese Kopie an der Stelle  auftrennt und an der Trennstelle die Kopie einer in A_1 bzw. A_{k+1} enthaltenen Form einbaut.
Beispiel einer ableitbaren Form F_k (Beispiel 1: k=4 Beispiel 2: k=1)		
Beispiel einer Form, die zu K gehört, aber nicht ableitbar ist		

Bild 2 Zwei Beispiele von Formenspielen

3. Anwendungsfelder

Es werden drei Anwendungsfelder betrachtet, wobei die Reihenfolge der Betrachtung durch eine schrittweise Einschränkung der Vielfalt von Möglichkeiten der Zuordnung von Bedeutungen zu den Formen bestimmt ist.

Im Abschnitt 3.1 wird nur angenommen, daß die Formen als Sprachgebilde interpretiert werden können. Zur Vielfalt der zuordenbaren Bedeutungen gehören in diesem Fall beliebige Ausdrücke, Aussagen, Anweisungen oder Fragen.

Im Abschnitt 3.2 ist die Vielfalt der zuordenbaren Bedeutungen auf Aussagen über beliebige abstrakte Welten eingeschränkt.

Im Abschnitt 3.3 schließlich geht es nur noch um Aussagen über einen bestimmten Typ abstrakter Welten, nämlich die Welt der Verhaltensmodelle für diskrete Systeme.

3.1 Formale Zeichenfolgesprachen

3.1.1 Formale Syntax

Wenn man nach der Strukturierung von Sprachgebilden fragt, dann wird man sofort an grammatikalische Regeln denken, denen die Sätze in natürlichen Sprachen gehorchen müssen. Es ist das Ziel dieses Abschnitts, den Begriff der Grammatik formal zu erfassen, damit er mit künstlichen Sprachen in Verbindung gebracht werden kann. Die künstlichen Sprachen dienen dazu, Information durch solche Formen zu repräsentieren, die von technischen Systemen, also insbesondere von Computern, als Formen eines gewünschten Formenspiels behandelt werden können. Da der Mensch dieses Formenspiel interpretieren kann, realisiert er auf diese Weise eine Informationsverarbeitung mit technischen Mitteln.

Man kann davon ausgehen, daß es in einer natürlichen Sprache nur endlich viele unterschiedliche Wörter gibt, selbst wenn man alle verschiedenen Formen mit demselben Wortstamm getrennt zählt. Jeder Satz ist eine endliche Folge von Vorkommen von Wörtern aus dem endlichen Wortrepertoire. Aus der unendlichen Menge K aller kombinatorisch bildbaren endlichen Wortfolgen aus dem endlichen Wortrepertoire wird durch die Grammatik eine echte Teilmenge L – man assoziiere dazu das engliche Wort *language* – abgegrenzt, die alle sog. *syntaktisch korrekten* und keine anderen Wortfolgen enthält. In dieser Teilmenge wird die Wortfolge

”Mozart starb in Wien.”

enthalten sein, nicht aber die Wortfolge

”Starb in gelbes Mozart und.”

Bei den praktisch relevanten formalen Sprachen ist die Menge L jeweils eine unendliche Menge. In diesen Fällen kann also die sog. *Syntax*, d.h. die Abgrenzung von L in K nicht dadurch angegeben werden, daß alle Elemente von L explizit aufgezählt werden. Zur Erfassung der Syntax von Symbolfolgen haben sich die *Grammatiken* als spezielle *axiomatische Systeme* durchgesetzt.

Der hier vorgestellte formale Grammatikbegriff geht auf den schon lange bekannten Begriff der Grammatik natürlicher Sprachen zurück. Dort gibt es grammatikalische Begriffe wie *Satz, Objekt, Prädikat, Attribut, adverbiale Bestimmung des Ortes* usw. zur Klassifikation von Wortfolgen. Die Wörter für diese grammatikalischen Begriffe können sowohl als Elemente in Formen zur Symbolisierung grammatikalischer Strukturen als auch als Wörter in grammatikalisch strukturierten Sätzen vorkommen. Man betrachte hierzu das zweimalige Vorkommen des Wortes ”Subjekt” im folgenden Beispiel:

Grammatikalisch strukturierter Satz:	Jeder vollständige Satz	enthält	ein Subjekt.
Grammatikalische Struktur des Satzes:	*Subjekt*	*Prädikat*	*Objekt*

In diesem Beispiel ist offensichtlich ”ein Subjekt” gar nicht das Subjekt, sondern das Objekt des Satzes. Die Notwendigkeit, zwischen den Elementen zur Erfassung grammatikalischer Strukturen und den Elementen in den grammatikalisch strukturierten Sprachgebilden unterscheiden zu müssen, führt in der Definition des formalen Grammatikbegriffs zur Unterscheidung zwischen den sogenannten *Superzeichen* und den sogenannten *Terminalen*. Jedes Superzeichen symbolisiert eine Klasse von Terminalfolgen. In natürlichen Sprachen sind die betrachteten Terminalfolgen ganze Sätze oder grammatikalisch abgrenzbare Satzteile; die Terminale sind hier die Wörter in den Sätzen.

Es ist durchaus sinnvoll, neben den reinen Terminalfolgen auch reine Superzeichenfolgen und Mischfolgen aus Terminalen und Superzeichen zu betrachten. Im folgenden Beispiel sind die Superzeichen die kursiv geschriebenen Wörter:

Satz ⇒ *Subjekt Prädikat Objekt* ⇒ *Subjekt* aß *Objekt* ⇒ Herr Müller aß ein Schnitzel.

Nicht nur hinsichtlich der reinen Terminalfolgen, sondern auch hinsichtlich der Superzeichenfolgen und der Mischfolgen muß die Frage nach der Verträglichkeit mit den grammatikalischen Regeln definiert sein.

Grammatikalisch korrekte Folge:	*Subjekt* aß *Objekt*.
Grammatikalisch inkorrekte Folge:	*Prädikat* aß *Objekt*.

In der folgenden Betrachtung wird die Menge aller hinsichtlich einer gegebenen Grammatik korrekten Folgen jeweils G genannt; die Benennung soll auf Grammatik hinweisen.

Grammatiken werden heute üblicherweise in einer Form definiert und klassifiziert, wie sie von Chomsky[1] eingeführt wurde. Durch eine *Grammatik* wird eine Terminalfolgenmenge L als echte Teilmenge einer Symbolfolgenmenge G beschrieben, wobei G durch ein axiomatisches System mit nur einem einzigen Axiom definiert wird. Das Symbolrepertoire für G enthält neben den Terminalen noch die *Superzeichen*. Die Menge L ist genau diejenige Teilmenge von G, die alle Symbolfolgen enthält, in denen nur Terminale vorkommen.

Das axiomatische System zur Definition von G ist ein Formenspiel mit den folgenden Merkmalen: Es gibt nur ein einziges Axiom, und dieses ist eine Symbolfolge der Länge eins, die aus einem Superzeichen besteht. In natürlichen Sprachen ist dieses Axiom üblicherweise das Superzeichen *Satz*. Das Spielendekriterium besagt, daß ein Spiel zu Ende gekommen ist, wenn eine Terminalfolge erreicht wurde. Jede Ableitungsregel beschreibt eine Möglichkeit, aus einer Symbolfolge, die keine reine Terminalfolge ist, dadurch eine neue Symbolfolge zu gewinnen, daß man ein Superzeichen durch einen bestimmten Symbolfolgenabschnitt ersetzt. Dabei kann die Ableitungsregel einen sogenannten *Kontext* angeben, in dem das zu ersetzende Superzeichen stehen muß, damit es entsprechend der Regel ersetzt werden darf. Die Ableitungsregeln haben also die Form in Bild 3.

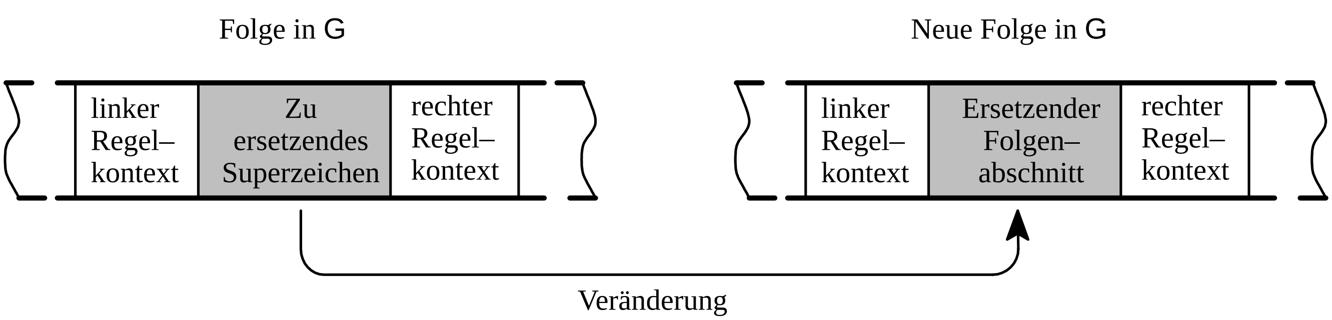


Bild 3 Form der Regeln einer Grammatik

Grundsätzlich ist die Möglichkeit nicht ausgeschlossen, daß der ersetzende Folgenabschnitt auch die leere Folge mit der Länge null sein darf. Aber diese Möglichkeit wird hier nicht weiter betrachtet, d.h. die folgende Betrachtung bleibt auf die sog. *längenmonotonen Grammatiken* beschränkt, bei denen in keiner Regel die neue Folge kürzer ist als diejenige, aus der sie durch Ersetzung eines Superzeichens gewonnen wird.

1) Noam Chomsky, geb. 1928, hat als Professor für Linguistik am Massachusetts Institute of Technology (bei Boston, USA) ab 1955 grundlegende linguistische Theorien entwickelt.

Nun soll die allgemeine Definition des Grammatikbegriffs durch ein Beispiel veranschaulicht werden. Es wird das Beispiel 2 aus Bild 2 betrachtet, anhand dessen bereits die Begriffe des formalen Systems und des axiomatischen Systems veranschaulicht wurden. Eine Grammatik hierzu sieht wie folgt aus:

Terminalrepertoire	=	{ ○ , □ }
Superzeichenrepertoire	=	{ S }
Einziges Axiom:		S
Ableitungsregeln:	(1)	S ⇒ □ ○
	(2)	S ⇒ □ S ○

Bild 4 zeigt die zu dieser Grammatik gehörenden beiden Mengen G und L. Die Regeln enthalten keine Kontextangaben, denn das Superzeichen S steht auf der linken Seite der Regeln alleine, und dies bedeutet, daß das Superzeichen S immer ersetzt werden darf unabhängig davon, wo es in einer Folge angetroffen wird.

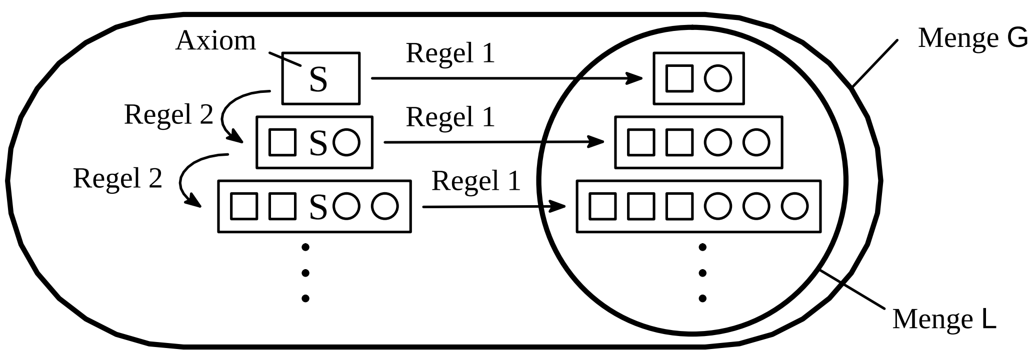


Bild 4 Mengen zu einem Grammatik–Beispiel

Weil die Symbolfolgen in der Menge L durch Anwendung der Ableitungsregeln aus dem Axiom abgeleitet werden, läßt sich zu jeder Symbolfolge in L mindestens ein sogenannter *Ableitungsbaum* angeben, der genau beschreibt, wie die Ableitung erfolgen kann. Eine Grammatik ist *eindeutig*, wenn es zu jeder Symbolfolge in L jeweils nur einen Ableitungsbaum gibt, wenn also diese Folge nur auf eine einzige Art und Weise aus dem Axiom abgeleitet werden kann; andernfalls ist die Grammatik *mehrdeutig*.

Bild 5 zeigt einen Ableitungsbaum über einem arithmetischen Ausdruck. Dieser Baum ergibt sich aufgrund der folgenden eindeutigen Grammatik für arithmetische Ausdrücke, in denen Addition und Multiplikation einziffriger Zahlen in beliebiger Klammerschachtelung vorkommen können:

Repertoire der Terminale = { 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , + , ∗ , (,) }

Repertoire der Superzeichen = { A, S, P, F, Z }
mit A für <u>A</u>usdruck und <u>A</u>xiom.
S für <u>S</u>ummandenform
P für <u>P</u>rodukt
F für <u>F</u>aktor
Z für <u>Z</u>ahl

Regeln:			
	(1) A ⇒ S	(5) P ⇒ F ∗ F	(9) Z ⇒ 1
	(2) A ⇒ S + A	(6) P ⇒ F ∗ P	(10) Z ⇒ 2
	(3) S ⇒ Z	(7) F ⇒ Z	⋮ ⋮
	(4) S ⇒ P	(8) F ⇒ (S + A)	
			(17) Z ⇒ 9

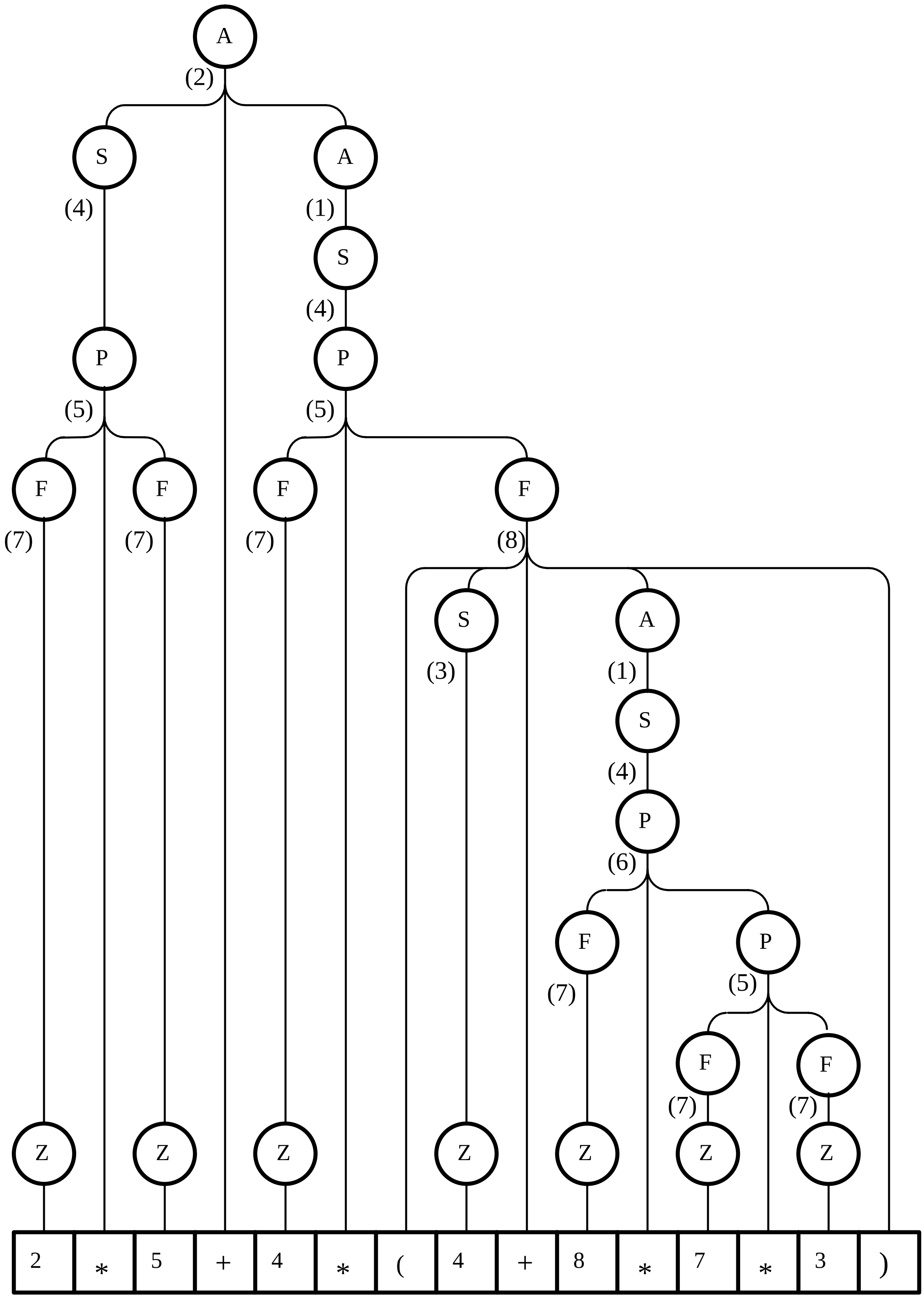


Bild 5 Beispiel eines Ableitungsbaumes

In den Knoten eines solchen Ableitungsbaumes steht jeweils ein Superzeichen, und dieses Superzeichen wird nach unten hin gemäß der durch die Zahlenangabe identifizierten Regel ersetzt. Da die Ableitung mit dem Axiom beginnt, muß im Wurzelknoten das Axiom stehen.

3.1.2 Formale Semantik

Neben der Syntax einer Sprache, also neben der formalen Abgrenzung der Teilmenge L aller interpretierbaren Terminalfolgen in der Menge K aller kombinatorisch möglichen Terminalfolgen muß es selbstverständlich auch noch die sog. Semantik geben. Mit dem Begriff *Semantik* bezeichnet man die Beziehung zwischen einer sprachlichen Form und der ihr zugeordneten Bedeutung, d.h.

> Die Semantik – gleichgültig, ob sie formal ist oder nicht – ist eine Funktion β, die jedem Element von L ein sogenanntes Interpretationsergebnis zuordnet.

Der Buchstabe β wurde gewählt, damit man dazu des Wort ”Bedeutung” assoziieren kann.

Zur nichtformalen, d.h. intuitiven Interpretation einer Terminalfolge muß man nicht nur wissen, was die einzelnen Terminale bedeuten, sondern auch, was es bedeuten soll, wenn bestimmte Terminale nebeneinanderstehen. Da die Möglichkeiten des Nebeneinanderstehens von Terminalen durch die Regeln der Syntax bestimmt werden, müssen die Interpretationsregeln mit den Regeln der Syntax gekoppelt sein. Das bedeutet, daß die Interpretation einer Terminalfolge an ihre syntaktische Struktur, also an den zugehörigen Ableitungsbaum gekoppelt ist.

Eine besonders enge Verbindung zwischen Syntax und Semantik besteht dann, wenn jede Anwendung einer Ersetzungsregel der Grammatik interpretiert werden kann. Da zu jeder Anwendung einer Ersetzungsregel eindeutig ein Superzeichenknoten im Ableitungsbaum gehört – denn durch die Anwendung einer Ersetzungsregel wird ja genau ein Superzeichen ersetzt –, kann in diesem Fall jedem Superzeichenknoten im Ableitungsbaum ein Interpretationsergebnis zugeordnet werden. Unter der Voraussetzung der Längenmonotonie ist jeder Superzeichenknoten Wurzel eines Teilbaums, dessen Blätter einen nichtleeren Terminalfolgenabschnitt bilden; und dieser Terminalfolgenabschnitt ist die Benennung oder Umschreibung dessen, was als Interpretationsergebnis dem Superzeichenknoten zugeordnet ist.

Als Beispiel sei die Terminalfolge in Bild 5 mit dem zugehörigen Ableitungsbaum betrachtet. Jedem Superzeichenknoten in dem Baum kann eindeutig eine natürliche Zahl zugeordnet werden. Man bestimmt diese Zuordnung von den Terminalen ausgehend von unten nach oben. Einem Superzeichenknoten Z ordnet man jeweils die Zahl zu, die durch das unmittelbar unter dem Z–Knoten hängende Terminal benannt ist. Diese den Z–Knoten zugeordneten Zahlen ordnet man auch den unmittelbar über den Z–Knoten liegenden S– oder F–Knoten zu. Das unveränderte Weiterreichen eines Interpretationsergebnisses von einem Superzeichenknoten zu einem unmittelbar darüberliegenden erfolgt in diesem Beispiel immer dann, wenn in der von oben nach unten führenden Ersetzungsregel einfach ein Superzeichen durch ein anderes ersetzt wird. Wenn dagegen durch die von oben nach unten führende Ersetzungsregel unter anderem als Terminal ein arithmetisches Verknüpfungssymbol, also ein + oder ein $\ast$ eingebracht wird, dann gewinnt man die dem oberen Knoten zuzuordnende Zahl durch Addition bzw. Multiplikation der beiden Zahlen, die den zugehörigen beiden unteren Knoten zugeordnet sind. Auf diese Weise erhält man letztlich die der Wurzel des gesamten Baumes zuzuordnende Zahl 698, die das Interpretationsergebnis der gesamten Terminalfolge darstellt.

In Bild 6 ist die Interpretationsvorschrift zu der Grammatik, die dem Baum in Bild 5 zugrundeliegt, in einer speziellen Formelschreibweise dargestellt. Darin wird die Symbolfolge $\beta(X)$ als Abkürzung für ”Interpretationsergebnis des Superzeichenknotens X” verwendet.

Da man hier zu jeder Ersetzungsregel einer Grammatik eine Interpretationsregel hinzugibt, spricht man von einer *attributierten Grammatik*.

Für das Verständnis ist es sehr wichtig, daß man sich der Unterschiede bezüglich der Verwendung der Symbole in den beiden Spalten in Bild 6 bewußt wird. Alle in der Syntax–Spalte vorkommenden Zeichen – außer der Numerierung und dem Ersetzungspfeil – stellen Formbausteine dar, denn hier werden die Regeln des Formenspiels beschrieben. Wenn in dieser Spalte ein + oder ein $\ast$ vorkommt, handelt es sich lediglich um einen Formbaustein, der an einer bestimmten Stelle in eine Form einzubauen ist. Wenn dagegen in der Semantikspalte ein + oder ein $\ast$ vorkommt, dann soll tatsächlich addiert oder multipliziert werden. Die Superzeichen, die in der Syntax–Spalte nur Formbausteine sind, werden in der Semantik–Spalte als Variable im Argument der Bedeutungszuordnungsfunktion β verwendet. Der Wertebereich zur Belegung dieser Variablen ist jeweils die Menge aller Terminalfolgenabschnitte, die man aus den entsprechenden Superzeichenknoten ableiten kann.

Immer dann, wenn in einer β–Formel mehrere Variable vom gleichen Knotentyp vorkommen, muß man Indizes zur Unterscheidung der Variablen einführen. Die Indizes l und r wurden gewählt, damit man dazu die Wörter links und rechts assoziieren kann, die sich auf den Pfeil in der jeweils zugehörigen Ersetzungsregel beziehen.

Syntax	Semantik
(1) A ⇒ S	(1) $\beta(A) = \beta(S)$
(2) A ⇒ S + A	(2) $\beta(A_l) = \beta(S) + \beta(A_r)$
(3) S ⇒ Z	(3) $\beta(S) = \beta(Z)$
(4) S ⇒ P	(4) $\beta(S) = \beta(P)$
(5) P ⇒ F ∗ F	(5) $\beta(P) = \beta(F_1) \ast \beta(F_2)$
(6) P ⇒ F ∗ P	(6) $\beta(P_l) = \beta(F) \ast \beta(P_r)$
(7) F ⇒ Z	(7) $\beta(F) = \beta(Z)$
(8) F ⇒ (S + A)	(8) $\beta(F) = \beta(S) + \beta(A)$
(9) Z ⇒ 1	(9) $\beta(Z) = 1$
(10) Z ⇒ 2	(10) $\beta(Z) = 2$
.	.
.	.
.	.
(17) Z ⇒ 9	(17) $\beta(Z) = 9$

Bild 6 Attributierte Grammatik zu Bild 5

Die Sprachen, in denen man über Sprachen spricht oder schreibt, werden *Metasprachen* genannt. Die Sprache, über die das Bild 6 Aussagen macht, ist die Sprache zur Formulierung einer bestimmten Art von arithmetischen Ausdrücken. Zur Formulierung von Aussagen über diese Sprache werden in Bild 6 zwei Metasprachen verwendet, nämlich eine Metasprache zur Formulierung der Syntax und eine andere Metasprache zur Formulierung der Semantik. So gehören beispielsweise der Ersetzungspfeil ⇒ zum Terminalrepertoire der einen und das Gleichheitszeichen = zum Terminalrepertoire der anderen Metasprache. Diese beiden Metasprachen sind selbst wieder formale Sprachen, d.h. man kann auch für diese beiden Sprachen wieder jeweils eine Grammatik angeben, die es erlaubt, die syntaktisch korrekten metasprachlichen Ausdrücke in der Menge aller kombinatorisch möglichen Terminalfolgen durch ein Formenspiel abzugrenzen.

Um Aussagen über eine Metasprache machen zu können, benötigt man wieder eine odere mehrere Metasprachen, die man auch ”Metametasprachen” nennen könnte. Die Kette der ”Metameta...metasprachen” endet immer bei einer natürlichen Sprache, die selbst ihre eigene Metasprache ist – man schreibt und spricht in Deutsch über die deutsche Sprache.

Die Frage, ob die zur Beschreibung der Semantik verwendete Metasprache formal ist oder nicht, hat nichts zu tun mit der Frage, ob die Semantik formal ist oder nicht.

> Eine *formale Semantik* liegt vor, wenn die Ergebnisse der Funktion β Formen sind, die in einem Formenspiel gewonnen werden. Das jeweilige Argument von β ist dabei als initiale Form des Formenspiels zu nehmen.

Das Argument von β ist in jedem Falle eine Form, nämlich der zur Wurzel oder einem anderen Knoten im Ableitungsbaum gehörende Terminalfolgenabschnitt. Aber das Ergebnis von β ist selbstverständlich nicht von vornherein eine Form.

Eine formale Semantik ist nur dann brauchbar, wenn sie jedem Paar von Argumenten von β, die für den Menschen unterschiedliche Bedeutung haben, auch unterschiedliche Ergebnisse zurdnet.

Im Beispiel in Bild 6 liefert die Funktion β als Ergebnis jeweils eine natürliche Zahl. Eine solche Zahl ist keine Form, sondern ein mathematisches Individuum, dem zum Zwecke der Kommunikation eine identifizierende Form zugeordnet werden kann. So wird beispielsweise durch jedes Element in der Formenmenge { 4, vier, IV } das gleiche mathematische Individuum identifiziert. Die Darstellung in Bild 6 kann wahlweise als formale oder nichtformale Semantik angesehen werden je nachdem, ob man als Ergebnismenge der Funktion β die Menge $\mathbb{N}$ der natürlichen Zahlen oder eine der Menge $\mathbb{N}$ eins zu eins zugeordnete Menge von Formen festlegt.

Die Regeln der dezimalen Zahlendarstellung definieren eine solche Formenmenge, die sich für die zwischenmenschliche Kommunikation als Norm durchgesetzt hat. Diese Darstellung eignet sich jedoch nicht gut für eine Formalisierung der Semantik der arithmetischen Operationen *Addition* und *Multiplikation*, d.h. im Falle der Dezimaldarstellung ist es nicht ganz leicht, ein Formenspiel zu erfinden, das zu der Form eines Additions– oder Multiplikationsausdrucks die Form des Ergebnisses liefert. Deshalb wird hier eine andere Darstellung für die natürlichen Zahlen gewählt.

Gewählte Zahlendarstellung	Äquivalente Dezimaldarstellung
I	1
II	2
III	3
IIII	4
⋮	⋮

Diese Formenmenge zur Zahlendarstellung kann als äußerst einfache formale Sprache betrachtet werden, deren Grammatik wie folgt aussieht:

Repertoire der Terminale = { I }
Repertoire der Superzeichen = { B } Das Superzeichen B steht für ”Ergebnis von β”.
Regeln:
(1) $\mathrm{B} \Rightarrow \mathrm{I}$
(2) $\mathrm{B} \Rightarrow \mathrm{IB}$

In der Semantik in Bild 6 wird durch die beiden Operatoren + und $\ast$ zum Ausdruck gebracht, daß jeweils zwei Zahlen addiert bzw. multipliziert werden müssen. Für die Formalisierung müssen nun die beiden Operatoren + und $\ast$ als formale Operatoren definiert werden, d.h. es muß zu jedem der beiden Operatoren ein formales Verfahren festgelegt werden, das ausgehend von der Form eines Verknüpfungsausdrucks zur Form des Ergebnisses führt. Die beiden Verfahren lassen sich rekursiv wie folgt definieren (Hinweis: x, y und z sind Variable für Formen.):

Formenverknüpfung durch den Operator + :

$x + y = \; xy$ d.h. man lasse den Formbaustein + weg und hänge die beiden Formen x und y unmittelbar hintereinander

Formenverknüpfung durch den Operator $\ast$:

$$x \ast y = \begin{cases} x & \text{falls } y = \mathrm{I} \\ x + x \ast z & \text{falls } y = \mathrm{I}z \quad \text{(Ausführungsreihenfolge: } \ast \text{ vor } +) \end{cases}$$

Beispiel:	II ✶ III	= II + II ✶ II	Dezimale Darstellung: 2 ✶ 3	= 2 + 2 ✶ 2
		= II + II + II ✶ I		= 2 + 2 + 2 ✶ 1
		= II + II + II		= 2 + 2 + 2
		= II + IIII		= 2 + 4
		= IIIIII		= 6

Unter Beachtung der gegebenen Formalisierung der beiden Operatoren + und ✶ kann für die formale Semantik der arithmetischen Ausdrücke die Tabelle aus Bild 6 fast unverändert übernommen werden. Man muß nur anstelle der Dezimalziffern 1 bis 9 die neuen Formen I bis IIIIIIIII setzen.

Die praktische Bedeutung von formalen Semantiken liegt vor allem im Bereich der verschiedenen Programmiersprachen [Fehr–89]. Als Terminalfolge wird jeweils der in einer Programmiersprache formulierte Programmtext betrachtet, dessen Interpretationsergebnis als Form festgelegt werden soll.

In den Fällen *ergebnisorientierter Programme* stellt das Programm eine Umschreibung einer Funktion f(x) dar, und der Zweck der Programmausführung besteht in der Berechnung des Funktionsergebnisses zu einem gegebenen Argument. In den Fällen *prozeßorientierter Programme* handelt es sich um die Umschreibung eines Prozeßtyps, und der Zweck der Programmausführung besteht im Auftreten eines Prozesses des programmierten Typs; ein solcher Prozeß ist eine zeitlich vollständig oder wegen Gleichzeitigkeit partiell geordnete Menge von Belegungsänderungen von Zustandsvariablen, d.h. von Ausführungen von Anweisungen des Typs $z_{k:=}\delta_k(Z, X)$. Darin ist z_k eine Zustandsvariable, δ_k ist die zu dieser Zustandsvariable gehörende Überführungsfunktion, Z ist die Variable für den jeweils relevanten Teil des Tupels aller Zustandsvariablen (z_1, z_2, z_3, ...), und X ist die Variable für den jeweils relevanten Teil des Tupels aller Eingangsvariablen (x_1, x_2, x_3, ...).

Die formalen Semantiken für Programmtexte lassen sich klassifizieren, wobei sich die Klassifikation daran orientiert, was der interpretierende Mensch zu der Ergebnisform der Funktion β assoziiert.

– Eine *operationale Semantik* liegt vor, wenn die Ergebnisform der Funktion β als Darstellung der Spielregeln oder eines Spielverlaufs eines Formenspiels interpretiert werden kann.
  Im Falle prozeßorientierter Programme kommen nur operationale Semantiken in Frage. Opera–tionale Semantiken sind aber auch brauchbar im Falle ergebnisorientierter Programme, die in einer prozeduralen Sprache formuliert sind.

– Eine *denotationale Semantik* liegt vor, wenn die Ergebnisform der Funktion β als standardi–sierte Beschreibung einer Funktionen f(x) interpretiert werden kann.
  Denotationale Semantiken gehören also immer zu ergebnisorientierten Programmen.
  Auf eine mögliche Unterklassifikation der denotationalen Semantiken wird hier verzichtet.

Die zu Bild 6 assoziierte formale Semantik ist denotational. Die Sprache zur Formulierung arithmetischer Ausdrücke kann als Programmiersprache angesehen werden, in der nur Funktionen f(x) formuliert werden können, die kein variables Argument haben; solche Funktionen sind Konstante. Die angegebene formale Semantik liefert genau eine eindeutige Form pro Konstante – z.B. die Form IIII zur Konstanten 4.

3.2 Kalküle

> Ein Kalkül ist ein Formenspiel, bei dem die Initialformen und alle durch Spielschritte erzeugbaren Formen als Aussagen über Individuen einer abstrakten Welt und ihre Beziehungen zueinander gedeutet werden können. Die abstrakte Welt wird durch die Initialformen festgelegt.

Meistens handelt es sich bei den Formen, die in Kalkülen verwendet werden, um prädikatenlogische Formeln. Zu den Formbausteinen gehören in diesem Fall die beiden Quantorensymbole $\forall$ und $\exists$ sowie die Symbole der Aussagenlogik, also TRUE , FALSE , $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrows$, $\oplus$ usw.. Daneben braucht man Formbausteine, die als Individuenbezeichner, Prädikatsbezeichner, Funktionsbezeichner oder als Bezeichner von Individuenvariablen, Prädikatsvariablen und Funktionsvariablen zu interpretieren sind. Um die Lesbarkeit der Formeln zu erleichtern, wird man für die Bezeichnerformen Normen festlegen. Ein Beispiel solcher Normen wird im Zusammenhang mit Bild 7 vorgestellt.

Die als Aussagen interpretierten Initialformen müssen widerspruchsfrei und voneinander unabhängig sein. *Widerspruchsfreiheit* schließt aus, daß der abstrakten Welt ein bestimmtes Merkmal gleichzeitig zugesprochen und abgesprochen wird. *Unabhängigkeit* schließt aus, daß eine Initialform durch Spielschritte aus den restlichen Initialformen erzeugt werden kann.

Jede derartige abstrakte Welt ist eine mathematische Struktur, also ein Tupel aus disjunkten, nichtleeren Individuenmengen $\mathrm{M_i}$ und – möglicherweise fehlenden – Relationen $\mathrm{R_k}$, deren Faktoren Individuenmengen aus dem Repertoire der $\mathrm{M_i}$ sind.

Diese sehr abstrakte Definition läßt sich leicht durch ein einfaches Beispiel verständlich machen. Es wird die Struktur der natürlichen Zahlen betrachtet, die nur aus einer Menge $\mathrm{M_1}$ und einer Relation $\mathrm{R_1}$ besteht:

$$\text{Struktur der natürlichen Zahlen} = (\mathrm{M_1}, \mathrm{R_1})$$

$$\begin{aligned} \mathrm{M_1} &= \{1, 2, 3, 4, 5, 6, 7, \ldots\} \\ \mathrm{R_1} &\subset \mathrm{M_1} \times \mathrm{M_1} \\ \mathrm{R_1} &= \{(1, 2), (2, 3), (3, 4), (4, 5), (5, 6), \ldots\} \end{aligned}$$

Die Menge $\mathrm{M_1}$ ist die unendliche Menge der natürlichen Zahlen, und die Relation $\mathrm{R_1}$ ist die Nachfolgerrelation, welche die Ordnung der natürlichen Zahlen festlegt. Ohne diese Relation wäre auf der Menge $\mathrm{M_1}$ keine Ordnung definiert. Die Menge $\mathrm{M_1}$ kann jedoch wegen ihrer Unendlichkeit ohne diese Ordnungsrelation gar nicht definiert werden.

Es ist wichtig zu erkennen, daß die gewählte Schreibweise mit den drei Punkten, welche die Unendlichkeit symbolisieren sollen, nur für diejenigen Leser verständlich sein kann, die schon mit der Struktur der natürlichen Zahlen vertraut sind, d.h. denen diese Struktur nicht erst anhand dieser Darstellung beigebracht werden soll.

Es stellt ein schwieriges Problem dar, diese unendliche Struktur durch eine endliche Menge von Aussagen eindeutig zu spezifizieren. Der Mathematiker Giuseppe Peano hat dieses Problem gelöst [Peano–98]. Die fünf sogenannten Peano'schen Axiome und ihre Interpretation sind in Bild 7 dargestellt.

Für die in den prädikatenlogischen Formeln vorkommenden Bezeichnerformen wurde die folgende Norm festgelegt:

Individuenbezeichner:	Kleine griechische Buchstaben,	z.B. α
Prädikatsbezeichner:	Steile lateinische Großbuchstaben,	z.B. N
Funktionsbezeichner:	Steile lateinische Kleinbuchstaben,	z.B. r
Bezeichner von Individuenvariablen:	Kursive lateinische Kleinbuchstaben,	z.B. *x*
Bezeichner von Prädikatsvariablen:	Kursive lateinische Großbuchstaben,	z.B. *P*

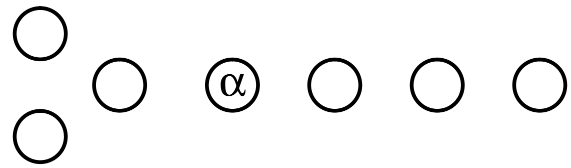
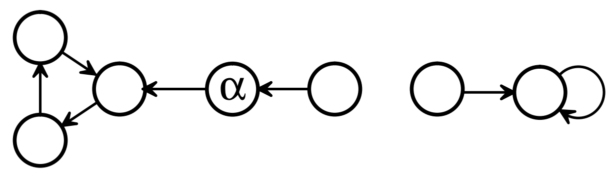
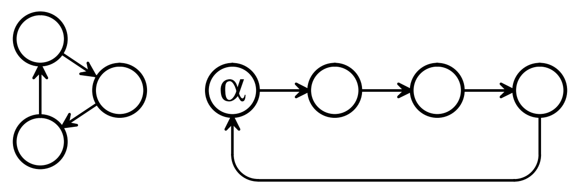
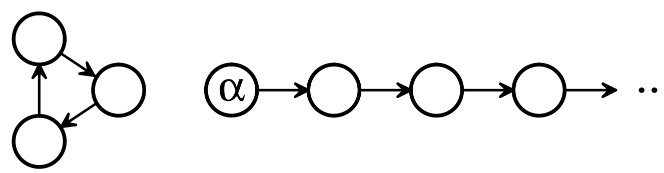
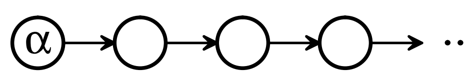
Axiome – als prädikatenlogische Formel – als übersetzte Formel in deutscher Sprache – in einer Formulierung, die auf die gewählte Veranschaulichung zugeschnitten ist	Beispiele von Strukturen, die mit den jeweiligen bisher eingeführten Axiomen verträglich sind
(1) $N(\alpha)$ Es wird die Menge aller Individuen betrachtet, auf die das Prädikat N zutrifft. Eines dieser Individuen heißt α. Es wird eine Menge kreisförmiger Knoten betrachtet. Einer dieser Knoten ist mit ”α” beschriftet.	
(2) $\forall x\colon\ N(x) \rightarrow N(\, r(x)\,)$ Auf der Menge der betrachteten Individuen ist eine Funktion r definiert, die in diese Individuenmenge hinein abbildet. Von jedem der Knoten geht ein Pfeil aus, der zu einem Knoten führt.	
(3) $\forall x, y\colon\ N(x) \cdot N(y) \cdot (\, r(x) = r(y)\,) \rightarrow (\, x = y\,)$ Wenn zwei Ergebnisse von r gleich sind, müssen die beiden Argumente gleich sein, d.h. r ist umkehrbar eindeutig. Es gibt keinen Knoten, auf dem mehr als ein Pfeil endet.	
(4) $\forall x\colon\ N(x) \rightarrow (\, r(x) \neq \alpha\,)$ α kommt nie als Ergebnis von r vor. Auf α endet kein Pfeil.	
(5) $\forall P\colon P(\alpha) \cdot (\, \forall x\colon P(x) \rightarrow P(r(x))\,) \rightarrow (\, \forall x\colon N(x) \rightarrow P(x)\,)$ Jedes beliebige Prädikat *P*, das für α gilt und dessen Gültigkeit stets vom Argument auf das Ergebnis von r übertragen wird, gilt für alle Individuen der betrachteten Menge. Es gibt keine Knoten, die nicht in der mit α beginnenden Kette hängen.	

Bild 7 Axiome von Peano zur Definition der Struktur ”Einseitig begrenzte Kette aus unendlich vielen diskreten Gliedern”

Daß hier das erste Glied in der Kette mit α und nicht mit 1 bezeichnet wurde, soll auf den Sachverhalt hinweisen, daß die Menge der natürlichen Zahlen auf beliebig viele Arten umkehrbar eindeutig auf andere Mengen abgebildet werden kann und daß durch die Peano’schen Axiome nur die gemeinsame Struktur all dieser Mengen beschrieben wird. Einige Beispiele solcher strukturgleichen Mengen sind

die Menge der Quadratzahlen	{ 1, 4, 9, 16, 25, 36, ... }
die Menge der Primzahlen	{ 2, 3, 5, 7, 11, 13, 17, ... }

die Menge der positiven Vielfache von fünf $\{ 5, 10, 15, 20, 25, 30, ... \}$

die Menge der ganzen Zahlen in geeigneter Ordnung $\{ 0, -1, 1, -2, 2, -3, 3, ... \}$

Durch die fünf Peano'schen Axiome werden drei "Sprachelemente" eingeführt, die man in Aussagen, die sich auf die abstrakte Peano–Welt beziehen, verwenden kann:

– das Prädikat N(x), welches besagt: x ist ein Individuum in der Peano–Welt.
– der Name α, der ein ganz bestimmtes Individuum in der Peano–Welt bezeichnet;
– die Funktion r(x), die jedem Individuum in der Peano–Welt ein Individuum in der gleichen Welt zuordnet.

Wenn man nun interessante Aussagen über die auf diese Weise eingeführte abstrakte Welt machen will, dann kann man dies nur tun, indem man neue "Sprachelemente" auf der Grundlage der bekannten drei Sprachelemente N, α und r definiert. So können beispielsweise die beiden Operatoren + und $\ast$ rekursiv auf der Grundlage von α und r definiert werden (wie in Abschnitt 3.1.2):

$$x + y = \begin{cases} \mathrm{r}(x) & \text{falls } y = \alpha \\ \mathrm{r}(x) + z & \text{falls } y = \mathrm{r}(z) \end{cases}$$

$$x \ast y = \begin{cases} x & \text{falls } y = \alpha \\ x + x \ast z & \text{falls } y = \mathrm{r}(z) \quad \text{(Ausführungsreihenfolge: } \ast \text{ vor +)} \end{cases}$$

Damit kann man beispielsweise das Prädikat PRIMZAHL definieren:

$$\mathrm{PRIMZAHL}(x) = [\, \exists\, w\!: \mathrm{N}(w) \cdot (\, x = \mathrm{r}(w)\,)\,] \cdot \quad [\, \neg \exists\, y, z\!:\; \mathrm{N}(y) \cdot \mathrm{N}(z) \cdot (\, \mathrm{r}(y) \ast \mathrm{r}(z) = x\,)\,]$$

Man beachte, daß die Definition des Primzahlprädikats keine Aussage über irgendwelche Sachverhalte in der abstrakten Peano–Welt ist, sondern eine Definition eines Sprachelements, das man in Aussagen über Sachverhalte in der Peano–Welt verwenden kann. Als Beispiel für die Verwendung des Sprachelements PRIMZAHL in einer Aussage über einen Sachverhalt in der Peanowelt wird die Aussage "Die Zahl 17 ist eine Primzahl." betrachtet. Diese Aussage hat im betrachteten Kalkül die Form

$$\mathrm{PRIMZAHL}(\; \mathrm{r(r(r(r(r(r(r(r(r(r(r(r(r(r(r(r(r(\alpha)))))))))))))))))}\;)$$

Man kann selbstverständlich das Sprachelement PRIMZAHL auch in falschen Aussagen verwenden, z.B. "Die Zahl 4 ist eine Primzahl", also

$$\mathrm{PRIMZAHL}(\; \mathrm{r(r(r(\alpha)))}\;)$$

Die Spielregeln eines Kalküls sollen dazu dienen, festzustellen, ob eine als Form vorgegebene Aussage über die abstrakte Welt "wahr" ist oder nicht. Deshalb beruhen die Regeln für die zulässigen Spielschritte, durch die man aus der Menge A_k der bisher erreichten Formen (s. Bild 1) eine neue Form F_k erzeugen kann, auf der Logik. Es handelt sich hier um sehr mächtige Spiele mit vielen Regeln, jedoch erfordert es der Zweck dieses Aufsatzes nicht, daß auf diese Regeln weiter eingegangen wird.

Jeder Kalkül grenzt innerhalb einer durch eine Grammatik definierten Aussagenmenge L eine Teilmenge B ab. Diese Teilmenge B ist die Menge der "beweisbaren Aussagen" des Kalküls. Die Abgrenzung der Teilmenge B in L ist dadurch gegeben, daß nicht alle Formen in L in einem Spiel mit endlich vielen Schritten aus den Axiomen abgeleitet werden können. Jede ableitbare Form ist Element von B, und jede nicht ableitbare Form liegt außerhalb von B.

Es wird verlangt, daß die Axiome eines Kalküls widerspruchsfrei sein sollen. Jedoch ist der Beweis, daß eine vorgegebene Menge von Axiomen tatsächlich widerspruchsfrei ist, keineswegs immer leicht zu führen. Man muß ja zeigen, daß die Form FALSE nicht in endlich vielen Spielschritten aus den Axiomen ableitbar ist.

Neben der Widerspruchsfreiheit interessiert auch die Frage nach der Vollständigkeit eines Kalküls. *Vollständigkeit* schließt aus, daß es in L wahre Aussagen über die durch die Axiome spezifizierte abstrakte Welt gibt, die nicht in endlich vielen Spielschritten aus den Axiomen ableitbar sind. Man beachte, daß die Vollständigkeit nicht die Widerspruchsfreiheit impliziert.

Die Definition des Begriffs der Vollständigkeit setzt voraus, daß die Wahrheitswerte der Aussagen in L unabhängig vom betrachteten Kalkül definiert sind, d.h. daß es eine über den betrachteten Kalkül hinausgehende Möglichkeit der Wahrheitsfindung gibt. Diese Möglichkeit könnte in Form eines Metakalküls gegeben sein, aber dann stellt sich sofort die Frage nach der Widerspruchsfreiheit und Vollständigkeit dieses Metakalküls. Dies führt schließlich zu der Frage, ob man nach einem Metameta...metakalkül suchen soll, den man als formale Methode der Wahrheitsfindung akzeptieren könnte, dessen Widerspruchsfreiheit und Vollständigkeit man also nicht mehr anzweifeln dürfte, weil sie mit den Mitteln dieses Kalküls selbst beweisbar wären.

Kurt Gödel [Gödel–31] hat gezeigt, daß man einen solchen Kalkül nicht zu suchen braucht, weil er nicht existiert. Gödel konnte nämlich zeigen, daß es innerhalb einer beliebigen Aussagenmenge L, wenn sie nur genügende Ausdrucksmächtigkeit besitzt, keinen Kalkül geben kann, der sowohl widerspruchsfrei als auch vollständig ist. Dies ist tatsächlich leicht einzusehen: Man nehme an, die Ausdrucksmächtigkeit der betrachteten formalen Sprache erlaube es, eine selbstbezügliche Aussage in L zu formulieren, die besagt: "Diese Aussage ist kein Element von B." Wenn sie zu B gehört, ist sie falsch, und der Kalkül ist widersprüchlich; wenn sie aber nicht zu B gehört, ist sie wahr, und der Kalkül ist unvollständig. Gödel hat gezeigt, daß sich diese Aussage als zahlentheoretische Aussage, also als Aussage über einen Sachverhalt in der abstrakten Peano–Welt formulieren läßt und daß deshalb die Unvereinbarkeit von Widerspruchsfreiheit und Vollständigkeit eine grundsätzliche Begrenzung mathematischer Erkenntnis darstellt.

Da es also einen widerspruchsfreien und vollständigen Kalkül innerhalb der Menge L aller mathematischen Aussagen nicht gibt, muß man sich mit weniger umfassenden Kalkülen zufrieden geben. Im einen Extrem sind dies Kalküle, deren Widerspruchsfreiheit und Vollständigkeit auf ihrer beschränkten Ausdrucksmächtigkeit beruht – hierzu gehört als einfachster Vertreter der sog. Aussagenkalkül; und im anderen Extrem sind dies Kalküle, deren Ausdrucksmächtigkeit bedingt, daß sich hier Widerspruchsfreiheit und Vollständigkeit gegenseitig ausschließen, und deren Widerspruchsfreiheit man bisher nicht beweisen konnte, obwohl man intuitiv davon überzeugt ist – hier handelt es sich um mächtige mathematische Kalküle. Zwischen diesen Extremen liegen diejenigen Kalküle, bei denen sich zwar auch Widerspruchsfreiheit und Vollständigkeit gegenseitig ausschließen, deren Widerspruchsfreiheit man aber immerhin beweisen konnte.

Neben der Frage nach der Widerspruchsfreiheit und der Vollständigkeit ist auch noch die Frage nach der *Entscheidbarkeit* wichtig für die Kennzeichnung eines Kalküls. Ein Kalkül ist entscheidbar, wenn man eine Maschine spezifizieren kann, die zu jedem vorgegebenen Element von L nach endlich vielen elementaren Schritten angibt, ob das Element zu B gehört oder nicht. Der sog. Aussagenkalkül, dessen Formen als Aussagen über Sachverhalte in der abstrakten Boole'schen Welt zu deuten sind, ist entscheidbar, während schon der Prädikatenkalkül erster Stufe nicht mehr entscheidbar ist.

3.3 Formale Systemspezifikation

Im vorliegenden Kontext ist unter einer Systemspezifikation eine endliche Menge von Formen zu verstehen, die als Aussagen über die Menge aller zulässigen Verläufe an den Systemschnittstellen zu interpretieren sind.

Im Falle diskreter Systeme – dies ist der einzige Fall, der hier interessiert – begnügt man sich häufig mit einer sog. *zeitfreien Spezifikation*. Bei einer zeitfreien Spezifikation verzichtet man auf jegliche Forderungen nach einer bestimmte Zeitlage bestimmter Ereignisse oder nach einem bestimmten zeitlichen Abstand zweier bestimmter Ereignisse. Für diesen Fall gilt:

Eine formale Systemspezifikation ist ein Kalkül, der eine abstrakte Welt beschreibt, in der Erscheinungen an den Schnittstellen eines Systems als abstrakte Individuen vorkommen, zwischen denen Beziehungen bestehen, die als zeitliche oder kausale Ordnungsbeziehungen oder als funktionale Abhängigkeitsbeziehungen zu deuten sind.

Eine solche Spezifikation setzt immer eine geeignete Abstraktion von den an den Systemschnittstellen beobachtbaren Erscheinungen voraus. Denn die beobachtbaren Erscheinungen spielen sich im räumlichen und zeitlichen Kontinuum ab, wogegen man in der Spezifikation Aussagen über eine Welt diskreter Individuen machen will. So kann beispielsweise der Vorgang "Einwurf einer Münze" in einer Spezifikation als unstrukturiertes Individuum – wie α in der Peano–Welt – behandelt werden, während doch der tatsächlich beobachtete Vorgang aus unendlich vielen infinitesimalen Schritten besteht.

Durch diese Abstraktion läßt sich jeder Verlauf in der zu spezifizierenden Menge aller zulässigen Verläufe als endliche Menge partiell geordneter Individuen auffassen. Jedes dieser Individuen entspricht einer bestimmten, durch die Abstraktion individualiserten Erscheinung, die im Falle des Vorkommens des betrachteten Verlaufs an den Systemschnittstellen beobachtet werden kann. Jedes geordnete Paar von Individuen ist als zeitlich geordnetes Paar von Erscheinungen zu deuten, und jedes Paar nicht geordneter Individuen ist als Paar gleichzeitiger Erscheinungen zu deuten. Ein Verlauf kann als gerichteter zyklenfreier Graph dargestellt werden.

Bild 8 zeigt ein Beispiel. Im oberen Graphen hat jedes Individuum seinen eigenen Knoten, während im unteren Graphen die Individuen zu Äquivalenzklassen zusammengefaßt sind, die vollständig geordnet sind. In den folgenden Betrachtungen ist mit dem Wort "Graph" immer nur ein Graph der oberen Art gemeint.

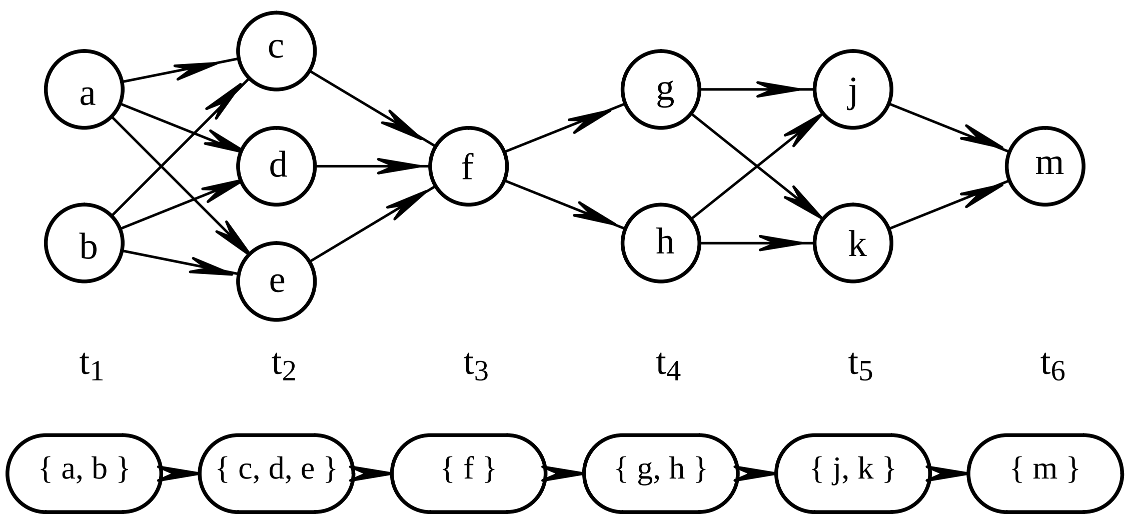


Bild 8 Alternative Graphendarstellungen eines abstrakten diskreten Verlaufs

Eine Spezifikation muß i.a. eine unendliche Menge solcher Graphen umschreiben, da die Menge der Vorgänge, die an den Schnittstellen des zu spezifizierenden Systems vorkommen dürfen, i.a. unendlich ist. Aber die Menge der Formen, die als Beschriftung in den Knoten vorkommen dürfen, muß endlich sein, da man nur endlich viele Typen von Erscheinungen in einer Spezifikation aufzählen kann. Beispiele für solche Typen von Erscheinungen sind

	Eine 1 DM–Münze wird eingeworfen.
oder	*Die rote Anzeigelampe beginnt zu leuchten.*
oder	*Die rote Anzeigelampe erlischt.*

Wenn die Beschriftung der Knoten in einer unendlichen Menge von Graphen aus einem endlichen Repertoire stammt, müssen zwangsläufig mehrere Knoten die gleiche Beschriftung tragen. So könnten im Beispiel in Bild 8 sowohl der Knoten d als auch der Knoten k die Beschriftung tragen *Die rote Anzeigelampe beginnt zu leuchten*. Die bei d eingeschaltete Lampe könnte ja bei f, g oder h wieder ausgeschaltet werden, und dann kann man sie bei k wieder einschalten.

Die Spezifikationen diskreter Systeme lassen sich in drei Komplexitätsklassen einteilen:

Spezifikationen sequentieller Systeme, mit den beiden Unterklassen
(1) Spezifikationen von Zuordnern
(2) Spezifikationen von sequentiellen Systemen mit Gedächtnis
(3) Spezifikationen von nichtsequentiellen, d.h. nebenläufigen Systemen

Bei Spezifikationen sequentieller Systeme geht es immer darum, den Zusammenhang zwischen den zulässigen Eingabefolgen ”X(n)” = (X(1), X(2), X(3), ...) und der jeweils zugehörigen Ausgabefolge ”Y(n)” = (Y(1), Y(2), Y(3), ...) zu beschreiben. Im Falle von *Zuordnern* besteht zwischen den beiden Folgen elementweise ein funktionaler Zusammenhang, also

$$Y(j) = f\,[\,X(j)\,]\,.$$

Die Spezifikation eines Zuordners ist deshalb gleich der Spezifikation der Funktion f. Es muß also ein Formenspiel angegeben werden, welches von der jeweils vorgegebenen Argumentform X(j) ausgehend zur Ergebnisform Y(j) führt. Das ist eine Aufgabe, die bereits im Abschnitt 3.1.2 über formale Semantik behandelt wurde.

Da der Argumentwertebereich endlich ist, kann man in den Fällen, in denen die Mächtigkeit des Argumentwertebereichs dies erlaubt, ein Formenspiel angeben, bei dem jedes Spiel nur aus einem einzigen Schritt besteht. Denn für jedes Element des Argumentwertebereichs kann man in diesem Fall eine eigene Spielregel angeben, die dem Argument unmittelbar das Ergebnis zuordnet.

Im Falle von *sequentiellen Systemen mit Gedächtnis* besteht zwischen den beiden Folgen ”X(n)” und ”Y(n)” ein funktionaler Zusammenhang derart, daß jedes Element Y(j) eindeutig durch den ”bis dahin” vergangenen Anfangsabschnitt der Folge ”X(n)” festgelegt ist. Je nachdem, wie man definiert, was mit der verbalen Zeitbestimmung ”bis dahin” gemeint sein soll, erhält man einen von zwei unterschiedlichen Systemtypen [Wendt–97.1]:

Abhängigkeit beim Moore–System: $Y(j) = f\,[\,X(1), X(2), X(3), \ldots\, X(j-1)\,]$

Abhängigkeit beim Mealy–System: $Y(j) = f\,[\,X(1), X(2), X(3), \ldots\, X(j-1), X(j)\,]$

Beim Moore–System ist also Y(1) eine von der Eingabe unabhängige Konstante, wogegen beim Mealy–System das erste Ausgabeelement Y(1) erst durch das erste Eingabeelement X(1) festgelegt wird.

Die Spezifikationen von sequentiellen Systemen mit Gedächtnis lassen sich in zwei Klassen einteilen: In *zustandsexpliziten Spezifikationen* wird neben der Eingabefolge ”X(n)” und der Ausgabefolge ”Y(n)” noch eine Zustandsfolge ”Z(n)” betrachtet, deren funktionaler Zusammenhang mit den Folgen ”X(n)” und ”Y(n)” die folgende Form hat:

$$\mathrm{rep}X(j) = a\,[\,Z(j)\,]$$

$$Z(j) = \begin{cases} \text{Konstante} & \text{falls } j = 1 \\ g\,[\,Z(j-1), X(j-1)\,] & \text{falls } j > 1 \end{cases}$$

$$Y(j) = \begin{cases} h\,[\,Z(j)\,] & \text{beim Moore–System} \\ h\,[\,Z(j), X(j)\,] & \text{beim Mealy–System} \end{cases}$$

Mit repX(j) ist das Repertoire bezeichnet, aus dem das Eingabeelement X(j) stammen darf.

In *zustandsimpliziten Spezifikationen* wird die Menge aller systemverträglichen Folgenpaare [”X(n)”, ”Y(n)”] ohne Betrachtung von Zuständen umschrieben. In der Praxis werden bisher fast ausschließlich die zustandsexpliziten Spezifikationen bevorzugt. Es wird hier darauf verzichtet, auf zustandsimplizite Spezifikationen näher einzugehen. Ein Beispiel findet man in [Wendt–97.2].

Größere Systeme sind i.a. nichtsequentiell, d.h. in diesen Systemen spielen sich nebenläufige – also kausal voneinander unabhängige – Vorgänge ab. Zur Spezifikation solcher Systeme benötigt man Konzepte, deren Darstellung den Rahmen dieses Aufsatzes sprengen würde. Die meisten Spezifikationsverfahren für nichtsequentielle Systeme beruhen auf dem Konzept der sog. Petrinetze.

4. Literatur

[Fehr–89] Elfriede Fehr: Semantik von Programmiersprachen.
Springer–Verlag, Berlin; 1989

[Gödel–31] Kurt Gödel: Über formal unentscheidbare Sätze der *Principia Mathematica* und verwandter Systeme I.
Monatshefte für Mathematik und Physik, 38, S. 173–198; 1931.

[Peano–98] Giuseppe Peano: Formulaire de mathemathiques. Band 2, § 2. Bocca, Turin; 1898.
Auszug in:
G. Peano: Arbeiten zur Analysis und zur mathematischen Logik.
Teubner–Archiv zur Mathematik, Band 13. Teubner–Verlag, Leipzig, 1990.

[Wendt–91] Siegfried Wendt: Nichtphysikalische Grundlagen der Informationstechnik.
2. Auflage. Springer–Verlag, Berlin; 1991.

[Wendt–97.1] Siegfried Wendt: Die Modelle von Moore und Mealy – Klärung einer begrifflichen Konfusion.
In: Ein Haufen Mosaiksteine – Gesammelte Aufsätze zur Theorie interpretierter Systeme. Universität Kaiserslautern, 1997.

[Wendt–97.2] Siegfried Wendt: Formalismen und Anschauung.
In: Ein Haufen Mosaiksteine – Gesammelte Aufsätze zur Theorie interpretierter Systeme. Universität Kaiserslautern, 1997.