

Fachgruppe Sprachwissenschaft

Universität Konstanz



Arbeitspapier 42

Das PATR II Format in Anwendung auf das Deutsche

Urs Egli & Klaus vonHeusinger

Inhalt

Einleitung

1 Das PATR-II Format

2

2.1 Die Kongruenz im einfachen Verbalsatz

2.2 Nominalphrasen und Kongruenz

2.3 Verbalphrase und Verb

2.4 Lexikalische Regeln

2.5 Subkategorisierung oder Verbvalenz

2.6 Der Lückenmechanismus

2.7 Satzstellung

Das PATR II Format in Anwendung auf das Deutsche¹

Urs Egli & Klaus von Heusinger

Einleitung

Das Ziel dieser Zusammenfassung soll es sein, das PATR-II Format (oder Formalismus) an einfachen syntaktischen Beispielen aus dem Deutschen einzuführen und zu erläutern. Dies kann kein historischer Abriss über die Entstehung eines Grammatiktyps, noch ein Vergleich dieses mit anderen Grammatiktypen sein. Dabei werden wir von einigen grundlegenden und damit als bekannt vorausgesetzten technischen (grammatischen) Begriffen ausgehen. Diese Darstellung des PATR-II Formats ist in dem Seminar "Unifikationsgrammatik" an der Universität Konstanz im Sommersemester 1987 von Herrn Prof. Egli entwickelt worden. Als Grundlage diente *An Introduction to Unification-Based Approaches to Grammar* (Shieber 1986)². Diese nur einführenden Beschreibungen des PATR-II Formats wurden im Seminar auf das Deutsche angewendet und dabei vertieft sowie mit Konzepten anderer Grammatiken angereichert, besonders sei hier die erweiterte Phrasenstrukturgrammatik (vgl. Egli & Egli-Gerber 1991) erwähnt.

Eine einleitende Bemerkung sei noch zu dem Begriff "Unifikationsgrammatik" gemacht. Dieser Begriff kann in zwei Hinsichten verstanden werden. Einmal handelt es sich um eine rein technische Beschreibung des Grammatiktyps. Unifikationsgrammatiken sind Grammatiken, die "unification-based" sind (s.u.). Weitere Merkmale solcher Grammatiken sind Monostratalität (d.h., die Oberfläche wird direkt, ohne Transformationen erzeugt), die Deklarativität und die zentrale Rolle des Lexikons (sie gehören zu den lexikongetriebenen Grammatiken). Solche Unifikationsgrammatiken sind als Instrument entwickelt worden, Sätze syntaktisch zu analysieren (= zu parsen). Sie erheben somit keinerlei Ansprüche auf eine linguistische oder psychologische Realität. Man nennt solche Grammatiken auch "methodische Grammatiken" im Gegensatz zu "substantiellen Grammatiken", in denen linguistische Intuitionen und Lösungen verarbeitet sind. Zu letzteren gehören u.a. GPSG, LFG, GB, etc., zu ersteren FUG, DCG, PATR-II etc.

In diesem Zusammenhang ist auch die zweite Hinsicht von "Unifikation" zu verstehen. Eine solche Unifikationsgrammatik kann den neutralen Formalismus anbieten, in den die linguistisch motivierten Lösungsvorschläge bestimmter Probleme von substantiellen Grammatiken übersetzt, sich vergleichen lassen. So lassen sich Stärken und Schwächen in einem neutralen Formalismus vergleichen und sich möglicherweise zu einer aussagekräftigeren Grammatik "unifizieren".

1. Das PATR-II Format

Bei der Analyse von Sätzen zerlegen wir diese in Satzteile und andere Einheiten (Konstituenten), die auf bestimmte Art miteinander verbunden sind und selbst bestimmte Merkmale tragen. Diese Konstituenten werden in bestimmte (Haupt-) Kategorien eingeteilt, denen je eine bestimmte Merkmalsstruktur (Nebenkategorien) eigen ist. Diese Merkmalsstrukturen werden als <Merkmal, Merkmalswert> angegeben, wobei das Merkmal selbst immer einfach (atomar), der Merkmalswert (zukünftig nur

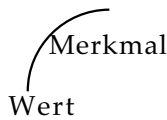
¹ Überarbeitete Version des Protokolls der Veranstaltung "Unifikationsgrammatik" von Prof. Egli im Sommersemester 1987.

² Wedekind 1991 gibt einen formalen Überblick über Systeme, die mit Unifikation arbeiten.

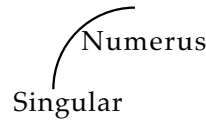
Wert) einfach oder komplex sein kann (1). Es gibt nun verschiedene Arten der Darstellung solcher Merkmalsstrukturen. Dem PATR-II Format ist eine Graphdarstellung eigen, die anschaulich die Merkmalsstrukturen je einer Kategorie zeigt (vergleichbar wie ein Phrasenstrukturbaum die Struktur eines Satzes graphisch deutlich macht). Jeder Graph kann auch als Term geschrieben werden. Doch werden wir hier nicht näher darauf eingehen, da in dieser Zusammenfassung nur der Formalismus des PATR-II Formats eingeführt und erklärt werden soll. In einem Graph entspricht jedem Merkmal je eine "Kante" (zur Unterscheidung von Strukturbäumen sind die "Kanten" hier Bögen), dem Wert entspricht ein an dem Endpunkt eines solchen Bogens angetragener Wert (einfach) oder wiederum Bögen, die dann für komplexe Merkmale stehen.

(1)

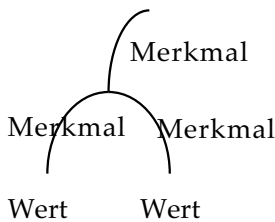
[Merkmal : Wert]



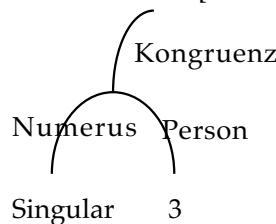
z.B.: [Numerus : Singular]



Merkmal : [Merkmal : Wert]
[Merkmal : Wert]



z.B.: Kongruenz : [Numerus : Singular]
[Person : 3]



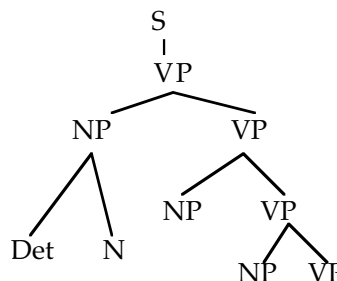
Im weiteren werden wir für Merkmale und ihre Werte entweder übliche (dann meist englische) oder intuitiv verständliche Abkürzungen benutzen. Sie werden beim ersten Gebrauch erklärt und dann in der eingeführten Form weiter gebraucht. Das PATR-II Format stützt sich auf die beiden Hauptoperationen der Verkettung (concatenation) und der Unifikation (unification). Die Verkettung von Konstituenten wird in einer üblichen PS-Regel (Phrasenstrukturregel) angegeben und in einem PS-Baum dargestellt. Die Konstituenten entsprechen im PS-Baum den Knoten.

(2)

(i) S -> VP

(ii) VP -> NP VP

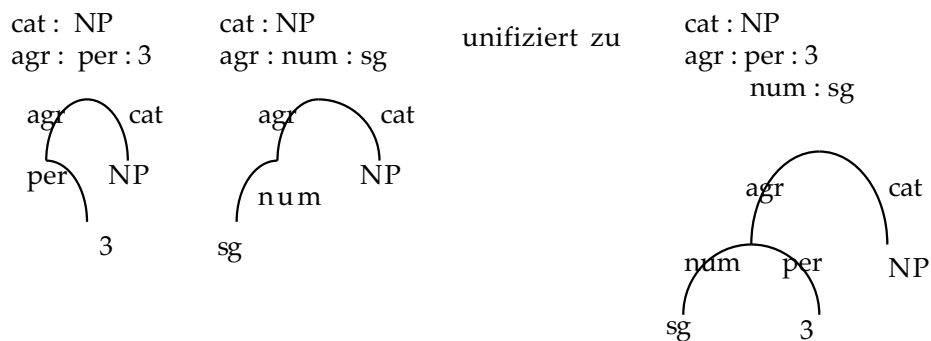
(iii) NP -> Det N



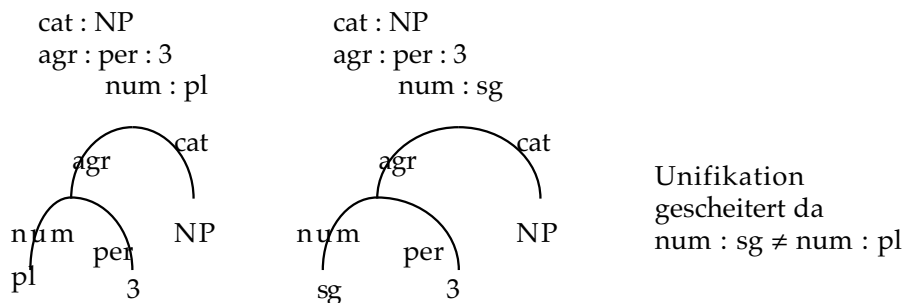
Regel (ii) kann wiederholt angewendet werden. Wir sprechen daher von einer rekursiven Regel. Regel (i) und (iii) sind im Prinzip auch rekursiv, also wiederholt anwendbare Regeln, doch verhindern sie in diesem Beispiel selbst, daß sie nochmals angewendet werden können.

Die zweite Hauptoperation ist die schon erwähnte Unifikation. Zwei Konstituenten lassen sich immer dann unifizieren, wenn sie keine widersprüchlichen Werte für das gleiche Merkmal haben. Die Unifikation läßt sich als Berechnungsanweisung für Terme definieren oder als Graphunifikation durch Übereinanderlegen der Graphen. Betrachten wir zunächst die graphische Darstellung der Unifikation, die typisch für PATR II ist. Sie ist folgendermaßen definiert: Zwei Graphen werden so übereinandergelegt, daß die Kanten, die gleiche Merkmale bezeichnen übereinanderliegen. Wenn es keinen Widerspruch der zugehörigen Werte gibt, ist die Unifikation gelungen (3a), gibt es hingegen einen Konflikt der Werte, so ist sie gescheitert (3b). Ist die Unifikation gelungen, so hat die neue Konstituente die Form des übereinandergelegten Graphen (3a). Die Unifikation eignet sich besonders für "teilspezifizierte" Strukturen, d.h. für Strukturen, die nicht alle Merkmale und Werte spezifiziert haben, wie in (3a). In (3) steht "cat" für Kategorie (< category), "agr" für Kongruenz (< agreement).

(3a)



(3b)



Wenn wir diese graphische Erklärung der Unifikation in Regeln (für Terme) fassen, erhalten wir für eine PATR-II Regel drei Komponenten (Wichtig ist die Unterscheidung in eine Test- und eine Aufbauregel, die in der Literatur nicht unterschieden werden):

1. Eine **Strukturregel** (vgl. (2)). Sie wird als PS-Regel angegeben und in Form eine PS-Baumes graphisch dargestellt.
2. **Testgleichungen**. Diese Gleichungen überprüfen, ob bestimmte Merkmale übereinstimmen. Ist dies der Fall, so ist die Unifikation möglich, andernfalls ist sie gescheitert. In diesem Falle kann die gesamte Regel nicht angewendet werden.
3. **Aufbaugleichungen**. Diese Gleichungen geben an, welche Werte die Merkmale der neu entstehenden Mutterkonstituente bekommen.

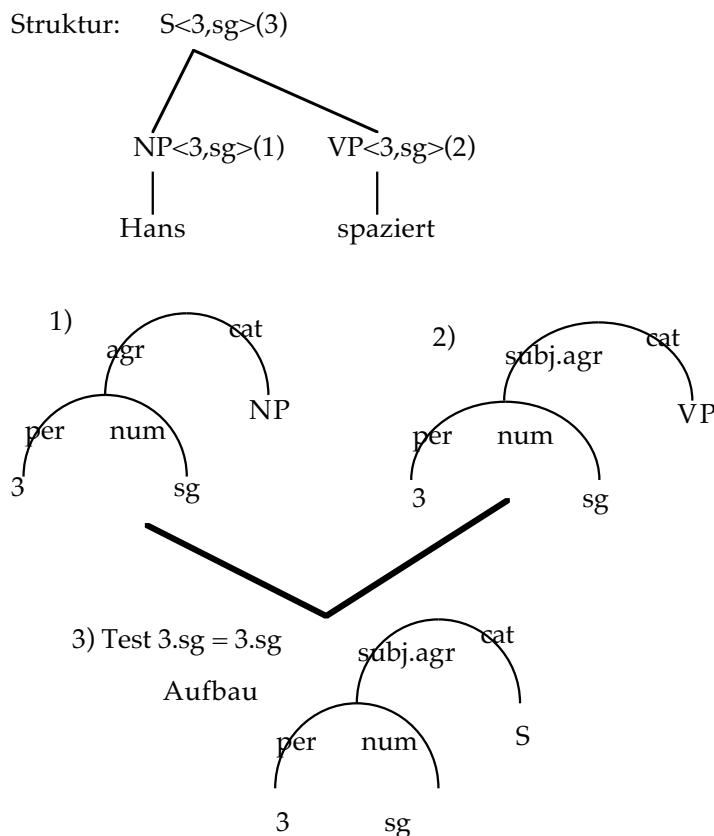
Diese allgemeine Struktur wird in den nächsten Abschnitten für spezielle Regeln benutzt. Jede PATR II- Regel hat die Teile *Struktur*, *Test* und *Aufbau*. Abschließend wollen wir noch bemerken, daß (aus traditionellen Gründen) die Struktur von oben nach unten angegeben wird, jedoch der Aufbau von unten nach oben (bottom up) vor sich geht. Daher eignet sich dies Format besser zur Analyse (Parsen) als zur Generierung von Sätzen. Im folgenden werden verschiedene syntaktische Erscheinungen des Deutschen mit einer PATR-II Regel beschrieben und dies an Beispielen erläutert.

2.1. Die Kongruenz im einfachen Verbalsatz

Im Deutschen (wie in vielen anderen Sprachen auch) stimmt das Subjekt eines Satzes in Person und Numerus mit dem Prädikat überein. Dies nennt man Kongruenz. Eine PATR-II Regel könnte so aussehen:

- (4) Struktur: S -> NP VP
 Test: NP_{agr} = VP_{subj.agr}
 Aufbau: S_{subj.agr} = VP_{subj.agr}

(4a)



"subj.agr" steht für Subjektskongruenz (des Prädikats). Im Strukturbaum geben wir in spitzen Klammern "< >" von Fall zu Fall bestimmte Merkmalswerte an den jeweiligen Knoten an, damit die Weitergabe von diesen Werten anschaulicher verfolgt werden kann. Denn vor allem bei größeren Bäumen verliert man leicht den Überblick. Die Zahlen in den runden Klammern "()" markieren die Knoten, für die der dann extra aufgeführte Merkmalsgraph steht. Hier sei noch einmal darauf hingewiesen, daß diese (Merkmalsstruktur-) Graphen immer nur lokal sind, d.h. sie beschreiben immer nur

einen Knoten des Strukturbaumes. Des weiteren vereinbaren wir, daß wir zwar auch "Kategorie" als Merkmal ("cat") mit den Werten NP, VP, etc. angeben; doch ist diese Merkmal insofern ausgezeichnet, als daß es nur von der Strukturregel nicht jedoch von Gleichungen betroffen ist. Daher spielt es auch bei der Unifikation keine Rolle. Damit ist sichergestellt, daß auch verschiedene Kategorien mit einander unifiziert werden können (vgl. 4a).

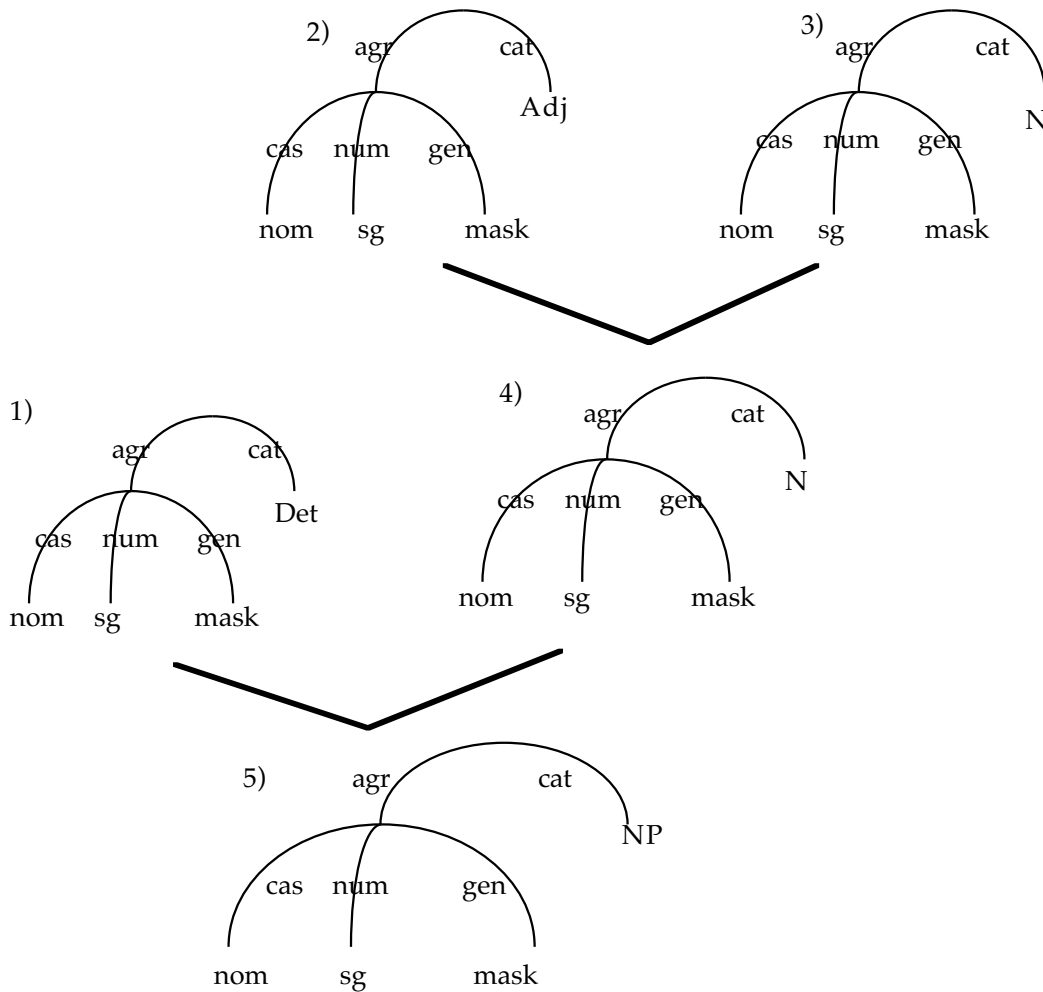
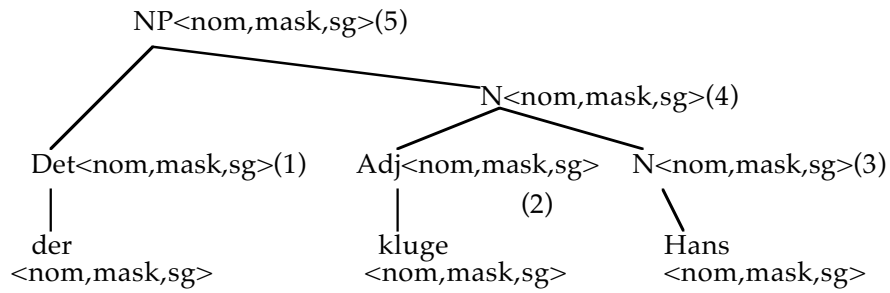
2.2. Nominalphrasen und Kongruenz

Nominalphrasen können im Deutschen in Artikel ("Det" < determiner) Adjektive und einfache Substantive ("N" < noun) zerlegt werden, die in Numerus ("num"), Kasus ("cas") und Genus ("gen") übereinstimmen müssen. So lassen sich u.a. folgende drei Regeln für eine NP aufstellen. Die erste besagt, daß eine NP in einen Determinator (z.B. Artikel) und ein Nomen übergehen kann (Struktur), dabei müssen die Kongruenzmerkmale des Determinators mit denen des Nomens übereinstimmen (Test). Die NP entsteht, indem sie die Kongruenzmerkmale des Nomens übernimmt (Aufbau). Die zweite Gleichung erfaßt den Fall, daß eine NP direkt in ein Nomen (ohne Determinator) übergehen kann. Dabei entfällt natürlich die Testgleichung. Schließlich brauchen wir noch die rekursive Adjektivregel, nach der eine Konstituente der Kategorie N aus einem Adjektiv(attribut) und einen Nomen bestehen kann.

(5)		i	ii.	iii
Struktur:		NP -> Det N	NP -> N	N ₁ -> Adj N ₂
Test:		Nagr = Detagr	- - -	Adjagr = N ₂ agr
Aufbau:		NPagr = Nagr	NPagr = Nagr	N ₁ agr = N ₂ agr

Der Index "1" und "2" an den Kategorien in der Strukturregel besagt, daß es sich um zwei verschiedene Ausdrücke der Kategorie N handelt (hier z.B. N₁ = [kluge Hans] und N₂ = [Hans]). Es ist nicht mit einer Bar-Schreibung zu verwechseln. Deshalb taucht dies auch nicht mehr in den Graphen auf. Der Unterschied zwischen starker und schwacher Flexion wurde nicht weiter berücksichtigt; auch ist mit diesen Regeln die Struktur der NP noch lange nicht befriedigend beschrieben worden. Hier soll nur versucht werden, zu zeigen, wie man es prinzipiell machen kann. Dazu analysieren wir die NP *der kluge Hans*.

(5a)



Der PS-Baum gibt die Struktur an. Die Knoten werden je durch einen Graphen dargestellt. Die Graphen (2) und (3) unifizieren nach Regel (iii) zum N-Knoten (4), der wiederum mit (1) nach Regel (i) zum NP-Knoten (5) unifiziert.

2.3. Verbalphrase und Verb

So wie N in NP übergehen kann, kann auch V in VP übergehen. Analog zu (5ii) stellen wir folgende Regel auf, die keine Modifikation des Verbes (z.B. durch Adverbien) ermöglicht. Für unsere einfachen Beispielsätze ist diese Regel jedoch ausreichend:

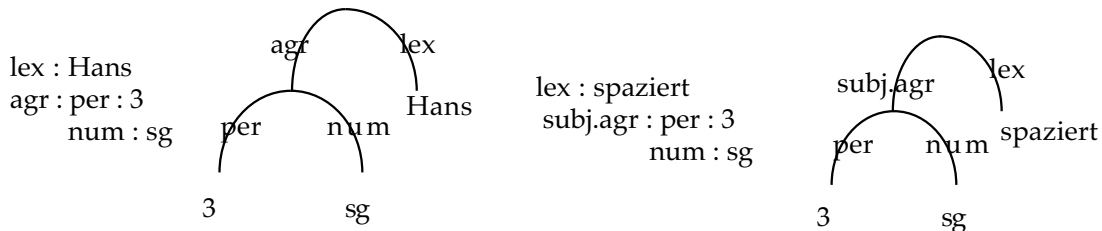
- (6) Struktur: VP -> V
 Test: ---
 Aufbau: VPmerk = Vmerk

"merk" soll für alle Merkmale stehen, die nicht schon genannt sind. Diese Konvention erspart erstens Schreiarbeit, und zweitens müssen die Regeln nicht jedesmal dann verändert werden, wenn ein neues Merkmal eingeführt wird. Da wir nicht alle notwendigen Regeln in jedem Beispiel angeben wollen, vereinbaren wir, daß wir auf die NP -> N und VP -> V Regeln in der Darstellung verzichten.

2.4. Lexikalische Regeln

PATR-II ist eine lexikongetriebene Grammatik d.h. ein Großteil aller Informationen kommt aus dem Lexikon. Ein Lexikoneintrag wird wie in (7) angegeben. Mit dem Merkmal "lex" wird angezeigt, daß es sich nicht mehr um eine Kategorie (nichtterminales Element), sondern um einen lexikalischen Eintrag (terminales Element) handelt. Da wir für jedes Wort (mindestens) einen Lexikoneintrag haben, der sich aber entsprechend der Kategorie in seiner Form gleicht, geben wir hier drei Regeln exemplarisch für andere an:

(7)



(8)

	i	ii	ii
Struktur:	N -> Hans	V -> spaziert	Adj -> kluge
Test:	---	---	---
Aufbau:	Nmerk = Hansmerk	Vmerk = spaziertmerk	Adjmerk = klugemerk

In dieser Schreibweise wird auch deutlich, daß alle Information (hier mit "merk" abgekürzt) aus dem Lexikon "hochgereicht" wird. In Beispiel (10a) wird gezeigt, wie ein solcher Eintrag übertragen werden kann.

In den Regeln wie (5ii), (6) oder (8), d.h. in Regeln, in denen ein Element aus nur einem anderen entsteht oder aufgebaut werden kann, handelt es sich nicht im engeren Sinne um Unifikation, sondern um Unterordnung ("subsumption", Shieber 1986, S. 14). In den Gleichungen fehlt die für die Unifikation charakteristische Testgleichung. Dennoch ist einleuchtend, daß solche Überführungen erlaubt sind. In unseren Fällen ändert sich außer der Kategorie kein weiteres Merkmal, was nicht unbedingt so sein muß. Schon bei den NP Regel hatten wir den Unterschied zwischen NP und N gesehen.

2.5. Subkategorisierung oder Verbvalenz

Wir gehen davon aus, daß die einzelnen NPs von der VP (Kopf des Satzes) abhängen. Dieses Verhältnis, daß nämlich die VP die NPs "regiert" oder bestimmt, nennt man traditionell (im englischen Sprachraum) Subkategorisierung ("subcategorization", Shieber 1986, S. 27ff); doch wollen wir es für unsere Zwecke als Valenz ("val") bezeichnen, da dies den Sachverhalt im Deutschen etwas besser beschreibt. Wir gehen davon aus, daß jedes Verb im Lexikon eine Valenz oder einen Kasusrahmen hat, der nach oben weitergereicht und dabei "abgearbeitet" wird. Ist der Kasusrahmen leer, so sprechen wir von einem Satz. In der anschaulicheren Schreibweise der erweiterte Phrasenstrukturgrammatik (vgl. Egly & Egly-Gerber 1991) sieht das so aus:

- (9) i VP<s> -> NP<a> VP<s,a>
 ii S -> VP< >

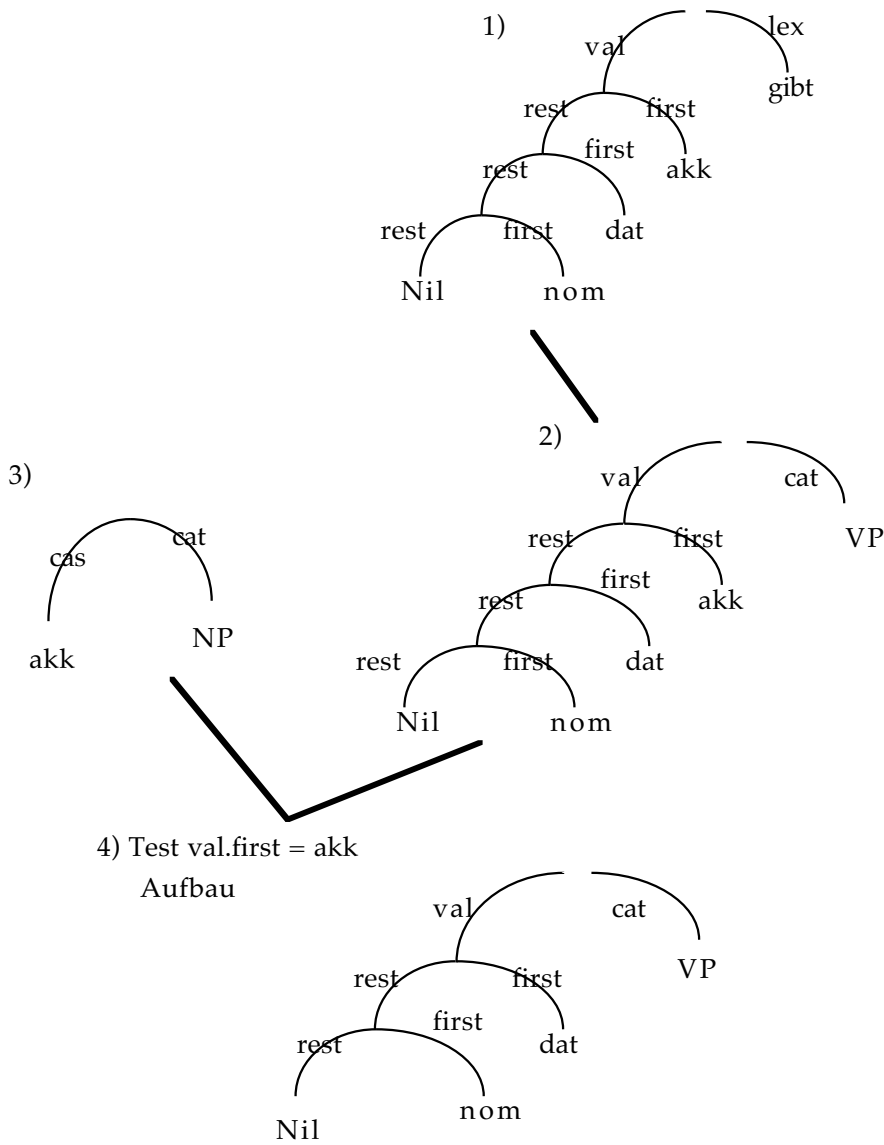
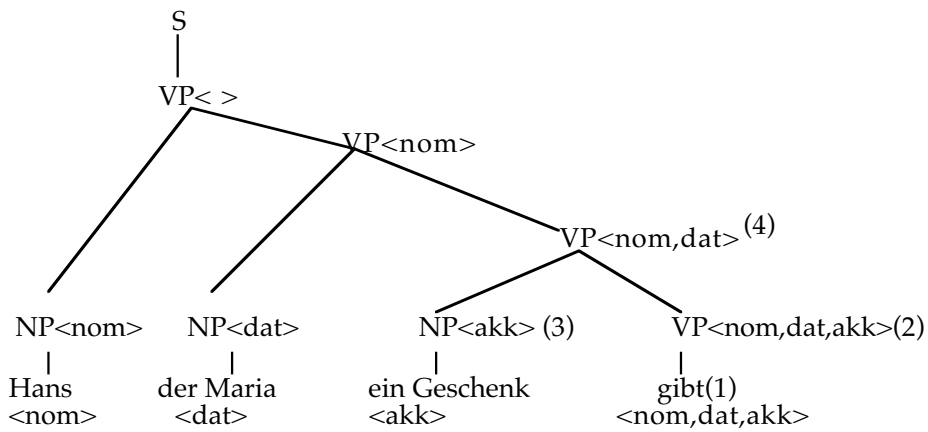
Die erste Regel besagt, daß wenn eine VP mit dem Kasusrahmen <s> zu einer NP mit dem Kasusmerkmal <a> und einer VP expandiert wird, diese den Kasusrahmen der ersten VP <s> plus dem Kasusmerkmal der NP <a> hat, also <s,a>. Dies läßt sich anschaulich an dem Strukturbaum in (10a) von oben nach unten verfolgen. Die zweite Regel S -> VP< > besagt, daß eine VP mit leerem Kasusrahmen ein Satz ist, d.h. der Satz ist immer eine Projektion der VP).

Da nun alle Gleichungen von PATR-II von unten nach oben arbeiten, müssen wir für die Erstellung unserer Gleichungen die Struktur von unten nach oben verfolgen (bottom up). Den Kasusrahmen oder die Valenz geben wir jeweils als geordnete Liste etwa folgender Form an: val = <<<Nil,nom>,dat>,akk>. Das äußerste (oder als erstes zu bearbeitende) Kasusmerkmal bezeichnen wir aufgrund der Stelle als "first", hier: val.first = akk. Den gesamten Rest bezeichnen wir als "rest", hier val.rest = <<Nil,nom>,dat>. "Nil" besagt, daß die Liste leer ist. Dies ist ein rein technisches Hilfsmittel. Nun können wir die beiden Regeln formulieren:

- | | | | | |
|------|-----------|--|--|----------------|
| (10) | | i | | ii |
| | Struktur: | VP ₁ -> NP VP ₂ | | S -> VP |
| | Test: | NPcas = VP ₂ val.first | | VPval = Nil |
| | Aufbau: | VP ₁ val = VP ₂ val.rest | | |
| | | VP ₁ merk = VP ₂ merk | | Smerk = VPmerk |

Die Subkategorisierung zeigen wir nun an dem Satz (*daß*) *Hans der Maria ein Geschenk gibt*. Dabei werden die Merkmalsstrukturen für das Lexem *gibt*, die VP<nom, dat, akk> und die NP<akk> und deren Unifikation VP<nom, dat> angegeben:

(10a)



2.6. Der Lückenmechanismus

Bisher sind wir stillschweigend von einer Verbendstellung im Satz ausgegangen. Doch damit können wir im Deutschen nur die Nebensatzstellung beschreiben. Um eine Hauptsatzstellung (Verbzweitstellung) beschreiben zu können, brauchen wir ein zusätzliches Instrumentarium. Wir gehen auch weiterhin davon aus, daß die Nebensatzstellung die grundlegende ist, entsprechend der Regel $VP_1 \rightarrow NP VP_2$. Da der Kasusrahmen von rechts nach links (im Baum) abgearbeitet wird, muß der Ausdruck mit dem ursprüngliche voll spezifizierte Kasusrahmen immer ganz rechts im Baum stehen. Im Nebensatz macht dies keine Probleme, da genau an dieser Stelle auch das Verb steht, was jedoch für den Haupt- und Fragesatz nicht zutrifft. Deshalb postulieren wir ein Element mit diesem Kasusrahmen am Ende dieser Sätze, das jedoch selbst nicht an der Oberfläche erscheint. Wir nennen dies Lücke. Eine Lücke ist eine Art "virtuelles Verb", das die gleichen Eigenschaften wie eine VP hat und genauso mit anderen Elementen interagiert. Im Gegensatz zu einer "richtigen" VP hat die Lücke ein zusätzliches Merkmal, nämlich das Lückenmerkmal, das anzeigt, für welches Element die Lücke eine Lücke ist. Im Gegensatz zu den anderen Werten der Lücke ist der Wert des Lückenmerkmals unveränderlich. So kann z.B. eine Lücke, die dem Verb *geben* entspricht folgendermaßen aussehen: $VP\langle \text{nom, dat, akk} \rangle / \langle \text{nom, dat, akk} \rangle$. Der Teil nach dem Querstrich "/" ist der Wert des Lückenmerkmals. Für den Umgang mit dieser Lücke (Lückenmechanismus) brauchen wir drei Regeln:

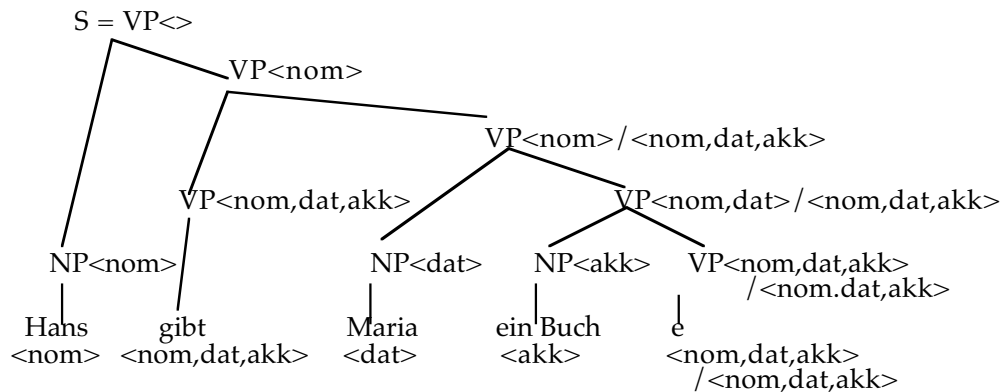
1. Eine Regel zur Schaffung von Lücken.
2. Eine Regel zur Weitergabe der Lücke.
3. Und eine Regel zur Eliminierung der Lücke (auch "linking rule" genannt).

(1) Wir gehen davon aus, daß wir für jede mögliche Verbvalenz eine Lücke mit entsprechendem Kasusrahmen im Lexikon stehen haben. (2) Die Lücke wird im weiteren wie eine normale VP behandelt, besonders die Regel zur Valenz ($VP_1 \rightarrow NP VP_2$) können wir ganz normal anwenden. Wir unterscheiden die Lücke von einer normalen VP durch das zusätzliche Merkmal "lücke", das wie andere Merkmale auch weitergereicht wird. (3) Die Linking-rule, d.h. die Eliminierung der Lücke, wenn man von unten nach oben arbeitet, hat folgende Form:

$$(11) \quad VP\langle s \rangle \rightarrow VP\langle x \rangle VP\langle s \rangle / \langle x \rangle$$

In Worten: Eine VP mit dem Kasusrahmen $\langle s \rangle$ entsteht aus einer VP mit dem Kasusrahmen $\langle x \rangle$ und einer Lücke mit dem Kasusrahmen $\langle s \rangle$ sowie dem Kasusrahmen $\langle x \rangle$ als Lückenmerkmal, der mit dem der anderen VP identisch sein muß. Dies kann man sich am besten wieder an dem Strukturbaum klar machen.

(11a)



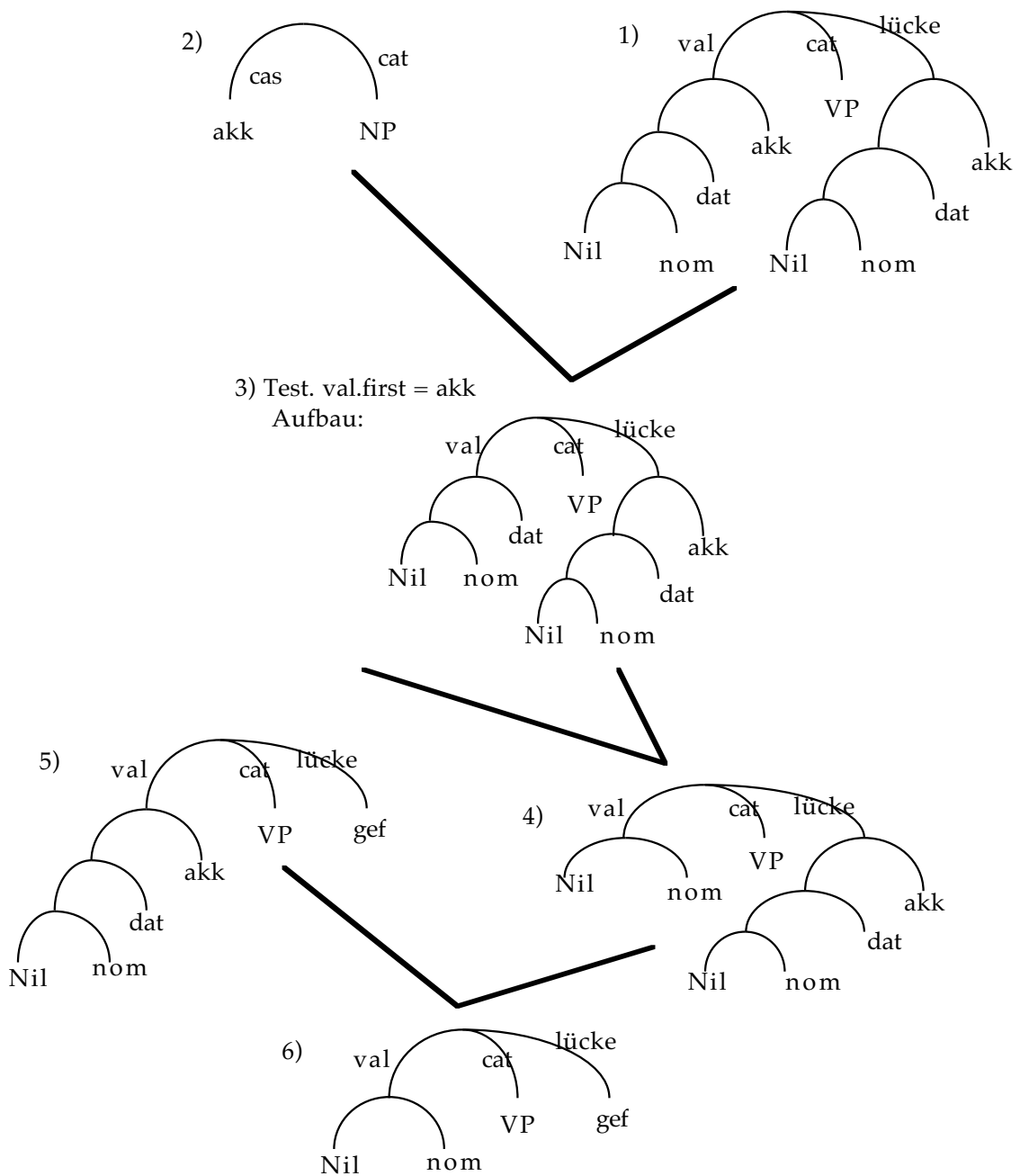
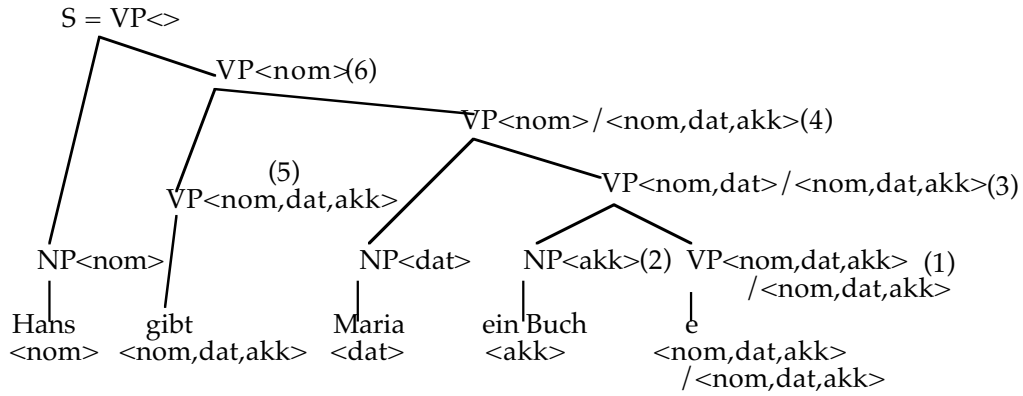
An der Struktur wird deutlich, daß die Lücke (im Baum mit "/" angegeben) ihren Kasusrahmen abarbeitet, ohne jedoch das Lückenmerkmal zu verändern. Die Lücke und ihr Lückenmerkmal wurden im Lexikon gleich dem voll spezifizierten Kasusrahmen gesetzt. Die Lücke wird nun so lange weitergereicht (wobei ihr Kasusrahmen immer weiter abgearbeitet wird), bis sie auf eine VP stößt, die selber keine Lücke ist und die gleichen Merkmale wie das Lückenmerkmal hat. Dann wird das Lückenmerkmal gestrichen, und das nun zur "richtigen" VP gewordene Element kann den Kasusrahmen weiter abarbeiten.

Somit müssen unsere Testgleichungen für die linking rule zwei Dinge überprüfen. Einmal ob der Kasusrahmen der einfachen VP gleich dem des Lückenmerkmals ist, und zweitens ob die einfache VP nicht etwa eine Lücke ist. Um das sicherzustellen, schreiben wir im Lexikon bei jedem Verb den Wert "gef" (< gefüllt) in das Merkmal "lücke"(vgl. (14a)). Nun können wir folgende Regeln aufstellen. Dabei schreiben wir in das Lückenmerkmal den Kasusrahmen.

(12)	(i) Linking-rule	(ii) Schaffung von Lücken
Struktur:	(1) $VP_1 \rightarrow VP_2 VP_3$	$VP \rightarrow e$
Test:	(2) $VP_2val = VP_3lücke$	---
	(3) $VP_2lücke = gef$	
Aufbau:	(4) $VP_1val = VP_3val$	$VPval = e.val$
	(5) $VP_1lücke = VP_2lücke$	$VPval = e.lücke$

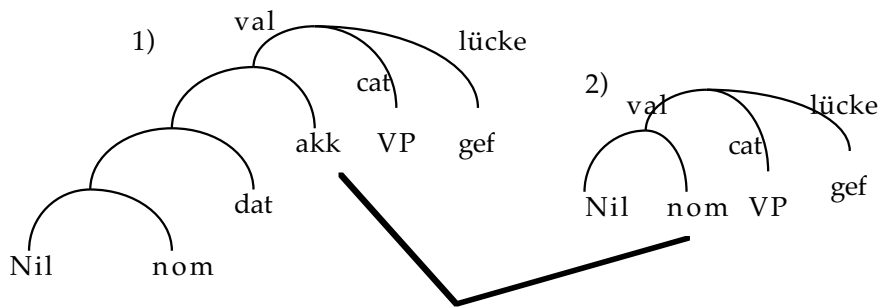
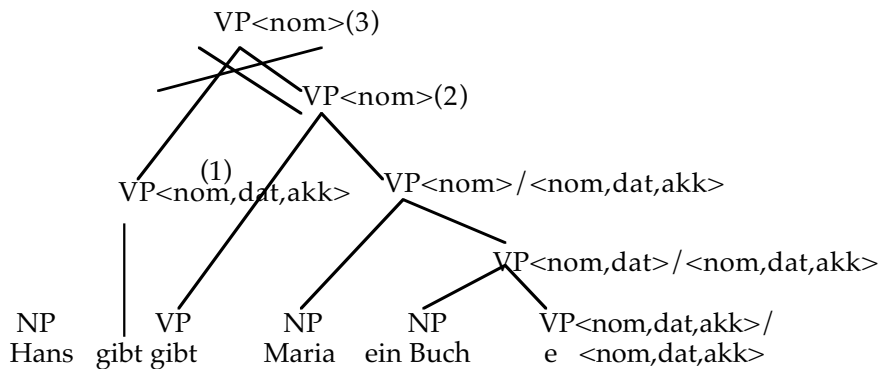
(1) ist die Linking rule aus (11). Der Test (2) prüft, ob das Lückenmerkmal identisch ist mit dem Verb, auf das die Lücke stößt. (3) prüft, ob dieses Verb selbst eine Lücke ist. Ist dies nicht der Fall, kann die neue VP aufgebaut werden. Sie erhält mit (4) den (Rest-) Kasusrahmen von der Lücke und mit (5) wird der Wert des Lückenmerkmals auf *gef* gesetzt. Dies hätte man auch durch eine Regel $VP_1lücke = gef$ erreichen können. Für die Schaffung von Lücken (ii) muß man sowohl den Kasusrahmen (val) als auch das Lückenmerkmal gleich dem Kasusrahmen einer VP setzen (hier: *geben*). Betrachten wir dies nun an dem Satz *Hans gibt Maria ein Buch*, den wir in (12a) analysieren.

(12a)



In (12b) wird gezeigt, wie bei dem ungrammatischen Satz *Hans gibt gibt der Maria ein Buch* die Unifikation scheitert

(12b)



3) Test: gescheitert da:
VP₂val ≠ VP₃lücke

2.7. Satzstellung

Mit dem oben beschriebenen Lückenmechanismus können wir jede beliebige Wortstellung aus einer Verbendstellung ableiten (wobei wir natürlich unterstellen, daß jeder Kasusrahmen in allen möglichen Anordnungen im Lexikon vorhanden ist). Dies wäre eine adäquate Beschreibung für Sprachen mit freier Wortstellung (wie z.B. das Lateinische). Doch das Deutsche unterliegt ganz klaren Beschränkungen: Im Hauptsatz darf das Verb nur an zweiter Stelle, im Fragesatz nur an erster Stelle stehen. Deshalb müssen wir den Lückenmechanismus für das Deutsche beschränken. Dabei stellen wir folgende Überlegung an: Im Fragesatz steht das Verb an erster Stelle, das heißt jedoch nichts anderes, als daß der Kasusrahmen der Lücke völlig abgearbeitet und damit leer sein muß, damit die Linking Rule angewendet werden kann. Im Hauptsatz darf nur noch ein Element in Kasusrahmen stehen. Somit erweitern wir die Testgleichungen um:

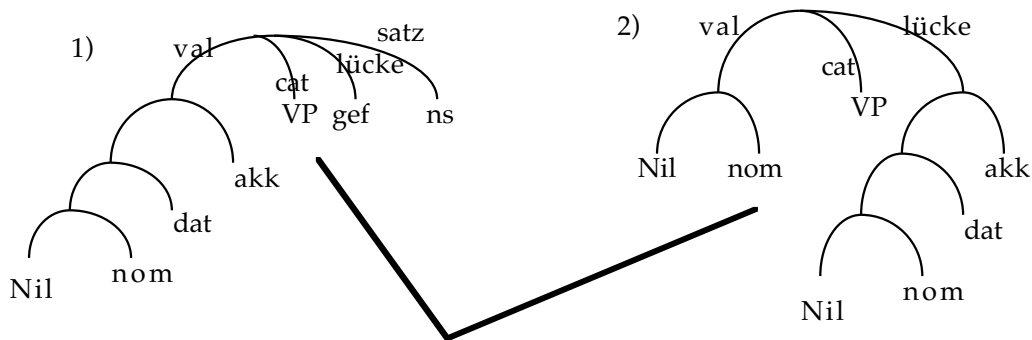
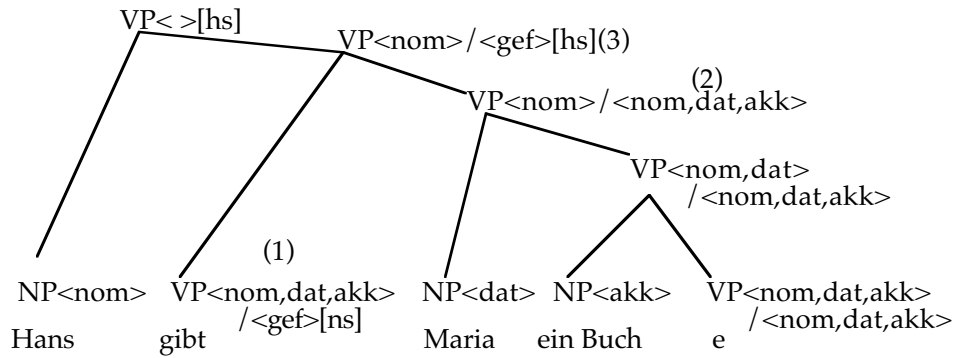
(13) i) Hauptsatz ii) Fragesatz
Test: VP₃val.rest = Nil VP₃val = Nil

Da wir nun gerne auch die Information, um welche Satzart es sich handelt, an den S-Knoten weitergeben wollen, führen wir das Merkmal "satz" (< Satzstellung) ein. Der unmarkierte ("default") Fall ist "ns" (Nebensatz), während "hs" (Hauptsatz) oder "fs" (Fragesatz) durch die Linking-rule eingeführt werden (dabei wird der default *ns* überschrieben). Unsere Regeln für den Haupt- und Fragesatz sehen nun so aus:

(14)	i) Hauptsatz	ii) Nebensatz	iii) Fragesatz
Struktur:	(1) $VP_1 \rightarrow VP_2 VP_3$	$VP_1 \rightarrow VP_2 VP_3$	$VP_1 \rightarrow VP_2 VP_3$
Test:	(2) $VP_2val = VP_3lücke$	$VP_2val = VP_3lücke$	$VP_2val = VP_3lücke$
	(3) $VP_2lücke = gef$	$VP_2lücke = gef$	$VP_2lücke = gef$
	(4) $VP_3val.rest = Nil$	$VP_3val = VP_2val$	$VP_3val = Nil$
Aufbau:	(5) $VP_1satz = hs$	$VP_1satz = ns$	$VP_1satz = fs$
	(6) $VP_1val = VP_3val$	$VP_1val = VP_3val$	$VP_1val = VP_3val$
	(7) $VP_1lücke = VP_2lücke$	$VP_1lücke = VP_2lücke$	$VP_1lücke = VP_2lücke$
	(8) $VP_1merk = VP_3merk$	$VP_1merk = VP_3merk$	$VP_1merk = VP_3merk$

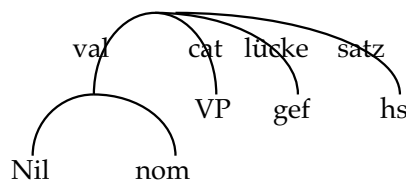
Wir zeigen diese erweiterte Regel an dem bereits bekannten Beispiel von Hans und Maria. In (14a) werden nur die Knoten dargestellt, die für die Linking rule (14i) wichtig sind. Das Merkmal *hs* wird zusätzlich im Strukturbaum eingetragen. In (14b) wird gezeigt, daß der ungrammatische Satz *Hans der Maria gibt ein Buch* nicht analysiert (oder geparst) werden kann, da keine der drei möglichen Linking Rules (14) angewendet werden kann.

(14a)

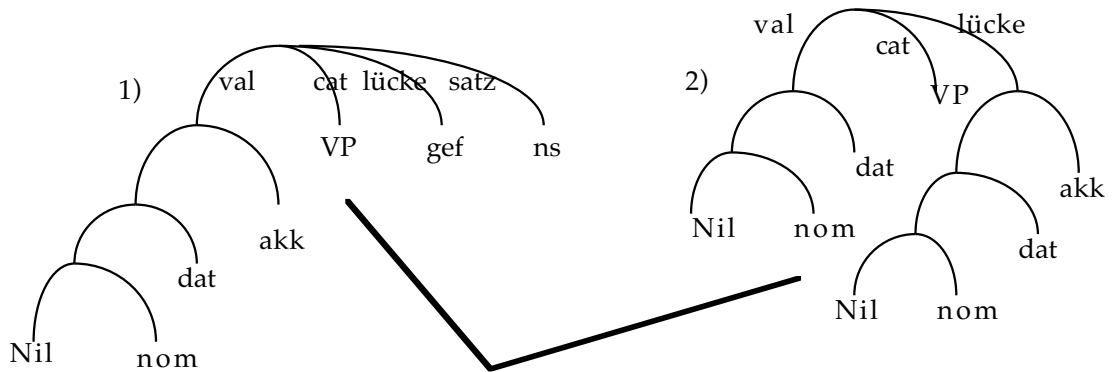
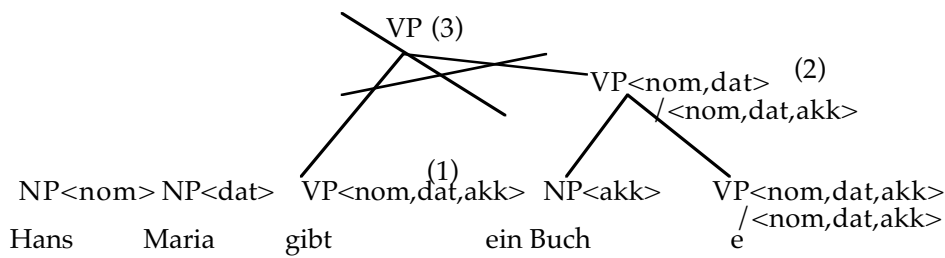


3) Test: 2: ja, 3: ja, 4: ja

Aufbau:



(14b)



3) Test: 2 ja, 3: ja, 4: nein
 - Unifikation gescheitert.

An diesen Beispielen sollte exemplarisch gezeigt werden, wie man in einer Unifikationsgrammatik, hier in PATR-II, den Unifikationsmechanismus für die Beschreibung wichtiger syntaktischer Phänomene einsetzen kann. Dabei haben wir zunächst ein allgemeines Regelformat entwickelt, das wir dann für die jeweiligen syntaktischen Verhältnisse wie Kongruenz, Rektion und Wortstellung speziell ausgearbeitet haben. Es sollte deutlich geworden sein, daß dies ein sehr allgemeiner Mechanismus ist, der sicherlich auch weitere, komplexere syntaktische Verhältnisse beschreiben kann.