

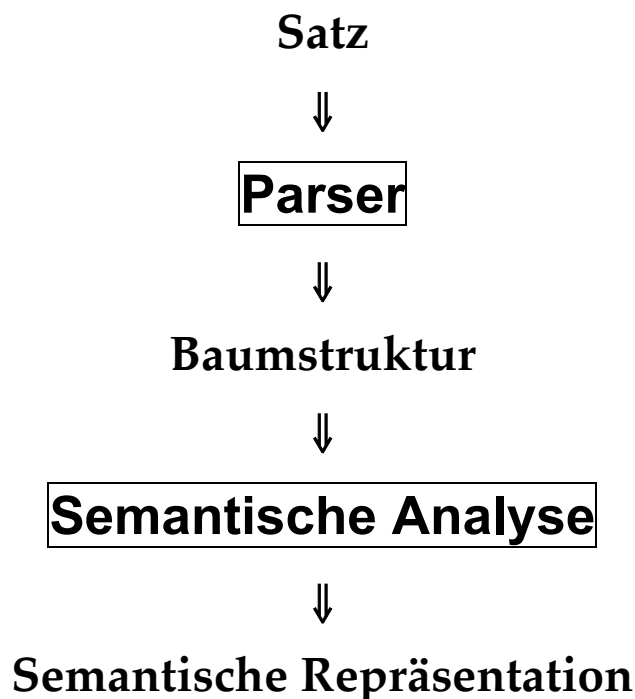
Semantische Analyse

Hierunter wird der Prozess verstanden, der für natürlichsprachliche Eingaben (Sätze) eine logische Bedeutungsrepräsentation erzeugt.

Eine syntaxgesteuerte semantische Analyse setzt Wissen über lexikalische Einheiten und syntaktische Strukturen ein.

Als Leitlinie wird hier das Kompositionalitätsprinzip (Frege-Prinzip) angesehen:

Die Bedeutung eines Satzes ergibt sich aus den Lexemen und der Satzstruktur (syntaktische Konstituenten).



Beispiel: **AyCaramba serves meat.**

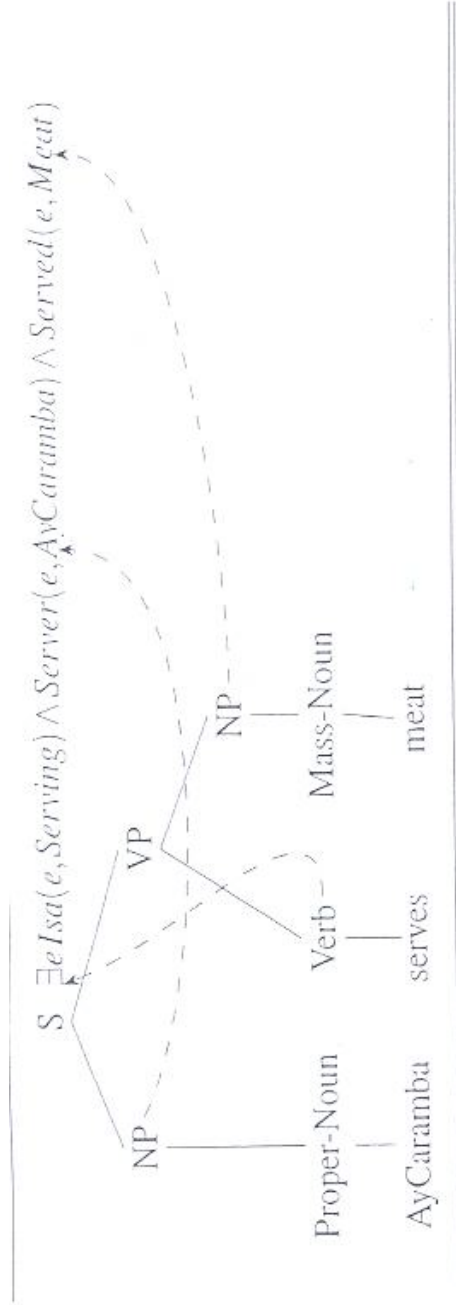


Figure 15.2 Parse tree for the sentence *AyCaramba serves meat*.

Semantische Anreicherung von Grammatikregeln

$$A \rightarrow \alpha_1 \dots \alpha_N \{ f(\alpha_J.\text{sem}, \dots, \alpha_K.\text{sem}) \}$$

Kontextfreie Grammatikregeln werden mit einem ‚semantischen Anhang‘ $\{ \dots \}$ versehen. Die Bedeutung der Konstruktion A ($A.\text{sem}$) ist dann eine Funktion f über einer Teilmenge der semantischen Anhänge der Konstituenten.

Dies soll im Folgenden an dem Beispiel erklärt werden.

$$\text{ProperNoun} \rightarrow \text{AyCaramba} \{ \text{AyCaramba} \}$$

$$\text{MassNoun} \rightarrow \text{meat} \quad \{ \text{Meat} \}$$

Die lexikalischen Regeln werden als logische Konstanten interpretiert.

$$\text{NP} \rightarrow \text{ProperNoun} \{ \text{ProperNoun.sem} \}$$

$$\text{NP} \rightarrow \text{MassNoun} \quad \{ \text{MassNoun.sem} \}$$

Bei nicht-verzweigenden Grammatikregeln werden die semantischen Anhänge nach oben weitergereicht.

Der semantische Anhang des Verbs könnte so aussehen:

$$\text{Verb} \rightarrow \text{serves}$$

$$\{ \exists e, x, y \text{ ISA}(e, \text{Serving}) \wedge \text{Server}(e, x) \wedge \text{Served}(e, y) \}$$

Die Bedeutung der VP könnte dann sein:

$$\{ \exists e, x, y \text{ ISA}(e, \text{Serving}) \wedge \text{Server}(e, x) \wedge \text{Served}(e, \text{Meat}) \}$$

Problem: woher weiß man, womit die Variablen der Verbbeschreibung instanziiert werden sollen?

Lambda Notation (Kalkül) als Logikergänzung

Durch Lambda-Ausdrücke werden formale Parameter eingeführt, die durch Terme ersetzt werden können:

$$\lambda x P(x) (A)$$

Die λ -Reduktion ersetzt die λ -Variable: $P(A)$

Lambda-Ausdrücke können ineinander eingebettet sein:

$$\lambda x \lambda y \text{ Near } (x, y)$$

Ein Beispiel:

$$\lambda x \lambda y \text{ Near } (x, y) (\text{ICSI}) \rightarrow \lambda y \text{ Near } (\text{ICSI}, y)$$

$$\lambda y \text{ Near } (\text{ICSI}, y) (\text{AyCaramba}) \rightarrow \text{Near } (\text{ICSI}, \text{AyCaramba})$$

Durch dieses Verfahren (Currying) kann man ein Prädikat mit mehreren Argumenten auf eine Folge von jeweils einstelligen Prädikaten abbilden.

Für die Verbrepräsentation heisst dies:

Verb $\rightarrow$ serves

$$\{ \lambda y \exists e, x \text{ ISA } (e, \text{Serving}) \wedge \text{Server } (e, x) \wedge \text{Served } (e, y) \}$$

Hierdurch hat man ein ‚äusseres‘ Element geschaffen, an das man die Objekt-NP binden kann:

$$\text{VP} \rightarrow \text{Verb NP} \quad \{ \text{Verb.sem } (\text{NP.sem}) \}$$

Durch λ -Reduktion erhält man die gewünschte VP-

Bedeutung:

$$\{ \exists e, x \text{ ISA } (e, \text{Serving}) \wedge \text{Server } (e, x) \wedge \text{Served } (e, \text{Meat}) \}$$

Für den Satz muss jetzt noch die Subjektfunktion an das Verb angebunden werden:

$$S \rightarrow NP VP \quad \{ VP.sem (NP.sem) \}$$

Dies wird wiederum durch Lambda-Abstraktion erreicht, indem der semantische Anhang des Verbs revidiert wird:

Verb $\rightarrow$ serves

$$\{ \lambda y \lambda x \exists e \text{ ISA } (e, \text{Serving}) \wedge \text{Server } (e, x) \wedge \text{Served } (e, y) \}$$

Durch die Anordnung der λ -Variablen wird erreicht, dass die Zuordnung der syntaktischen Konstituenten den semantischen Rollen entspricht:

$$\exists e \text{ ISA } (e, \text{Serving}) \wedge \text{Server } (e, \underline{\text{AC}}) \wedge \text{Served } (e, \underline{\text{Meat}})$$

Das bisherige Beispiel war insofern sehr einfach, als die NPs nur aus logischen Konstanten bestanden.

A restaurant serves meat.

Nach dem bisherigen Vorgehen würde die folgende semantische Repräsentation entstehen:

$$\exists e, x \text{ ISA } (e, \text{Serving})$$

$$\wedge \text{Server } (e, x) \wedge \text{Served } (e, \text{Meat}) \wedge \text{ISA } (x, \text{Restaurant})$$

Durch Lambda-Abstraktion würde die folgende Darstellung entstehen:

$$\exists e \text{ ISA } (e, \text{Serving})$$

$$\wedge \text{Server } (e, \exists x \text{ ISA } (x, \text{Restaurant})) \wedge \text{Served } (e, \text{Meat})$$

Das 2. Argument von **Server** ist jetzt allerdings kein Term!

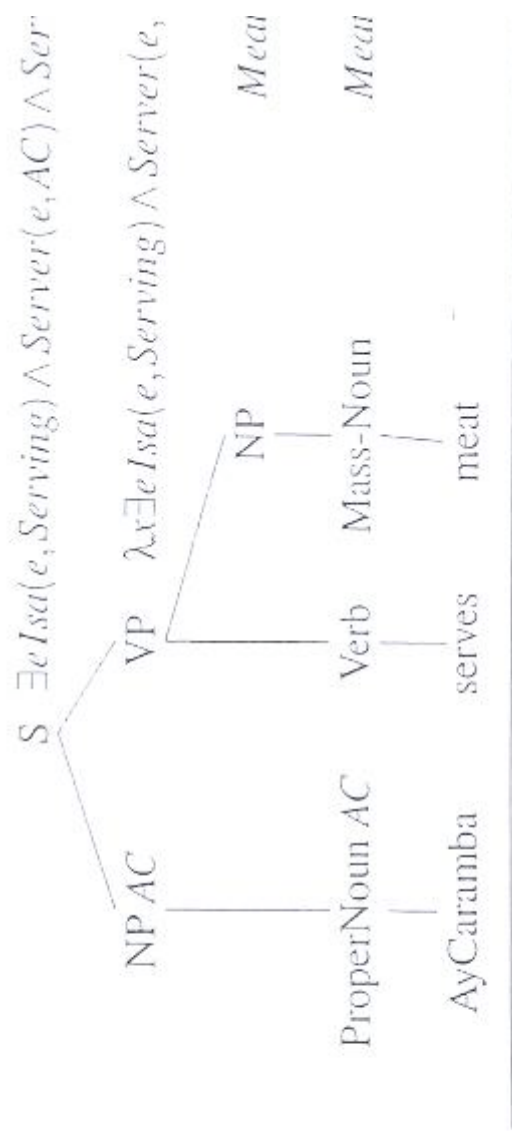


Figure 15.3 Parse tree with semantic attachments f

Hier kann jetzt ein ähnlicher Trick angewendet werden wie bei den λ -Ausdrücken, Komplexe Terme als Tripel:

< Quantifier variable body >

Für das Beispiel heißt dies:

$\exists e$ ISA (e, Serving)

$\wedge$ Server (e, < $\exists x$ ISA(x, Restaurant)>) $\wedge$ Served (e, Meat)

Um dies in Prädikatenlogik überführen zu können, braucht man eine Transformation, wie etwa diese:

P (<Quantifier variable body>)

$\Rightarrow$ Quantifier variable body Connective P(variable)

Im Beispiel (**a restaurant**) sieht das wie folgt aus:

Server (e, < $\exists x$ ISA(x, Restaurant)>)

$\Rightarrow \exists x$ ISA(x, Restaurant) $\wedge$ Server (e, x)

Connective richtet sich nach dem Quantor, $\wedge$ für $\exists$, $\Rightarrow$ für $\forall$.

Der semantische Anhang für die Syntaxregeln des Beispiels:

NP $\rightarrow$ Det Nominal { <Det.sem x Nominal.sem(x)> }

Det $\rightarrow$ a { $\exists$ }

Nominal $\rightarrow$ Noun { λx ISA (x, Noun.sem) }

Noun $\rightarrow$ restaurant { Restaurant }

Die Einführung der λ -Ausdrücke und der komplexen Terme werden oft als quasi-logische Formen oder als Zwischenrepräsentationen bezeichnet.

Quantorenskopus und komplexe Terme

Die schematische Abbildung komplexer Terme in logische Ausdrücke ist nicht eindeutig, wie an folgendem Beispiel gezeigt werden soll:

Every restaurant has a menu.

Mit quasi-logischen Darstellungen wäre dies:

$\exists e \text{ ISA } (e, \text{Having})$

$\wedge \text{Haver } (e, \langle \forall x \text{ ISA } (x, \text{Restaurant}) \rangle)$

$\wedge \text{Had } (e, \langle \exists y \text{ ISA } (y, \text{Menu}) \rangle)$

Die Umformung der komplexen Terme könnte dies ergeben:

$\forall x \text{ Restaurant } (x) \Rightarrow$

$\exists e, y \text{ Having}(e) \wedge \text{Haver}(e, x) \wedge \text{ISA}(y, \text{Menu}) \wedge \text{Had } (e, y)$

Oder auch dies:

$\exists y \text{ ISA } (y, \text{Menu}) \wedge \forall x \text{ Restaurant } (x) \Rightarrow$

$\exists e, y \text{ Having}(e) \wedge \text{Haver}(e, x) \wedge \text{Had } (e, y)$

Generell heißt dies, dass logische Formeln mit zwei komplexen Termen immer auch zwei mögliche Interpretationen haben.

Praktisch werden solche Mehrdeutigkeiten mit Heuristiken aufgelöst, etwa durch Weltwissen, oder durch die links-rechts Anordnung der Nominalphrasen.

Semantische Anhänge für ein Fragment des Englischen

Unterscheidung zwischen Deklarativsätzen, Imperativen, Ja-Nein Fragen und W-Fragen:

Flight 487 serves lunch.
Serve lunch!
Does Flight 207 serve lunch?
Which flights serve lunch?

Ähnlich wie bei Belief-Ausdrücken werden hier neue Operatoren eingeführt: DCL, IMP, YNQ und WHQ.

- Deklarativsätze:

$S \rightarrow NP VP \quad \{ DCL (VP.sem (NP.sem)) \}$

Eine Interpretation dieser Repräsentation könnte in der Ergänzung der Wissensbasis bestehen.

- Imperative:

$S \rightarrow VP \quad \{ IMP (VP.sem (DummyYou)) \}$

Hier wird für das fehlende Subjekt eine λ -Variable eingeführt, die dann durch eine Konstante belegt wird:

$IMP (\exists e \text{ Serving}(e)$
 $\quad \wedge \text{Server}(e, DummyYou) \wedge \text{Served}(e, Lunch))$

Eine Interpretation könnte eine Aufforderung sein, die durch die VP spezifizierte Handlung auszuführen.

- Ja-Nein Fragen

$S \rightarrow \text{Aux NP VP} \quad \{ \text{YNQ (VP.sem (NP.sem))} \}$

Bis auf den YNQ Operator entspricht das Deklarativsätzen.

$\text{YNQ } (\exists e \text{ Serving}(e))$

$\wedge \text{Server}(e, \text{Flight207}) \wedge \text{Served}(e, \text{Lunch}))$

Eine Interpretation könnte darin bestehen, den Wahrheitswert für die Proposition zu bestimmen.

- W-Fragen

$S \rightarrow \text{WhWord NP VP}$

$\{ \text{WHQ (NP.sem.var, VP.sem (NP.sem))} \}$

(NP.sem.var realisiert den Zugriff auf die Subjektvariable)

$\text{WHQ } (x, \exists e, x \text{ Serving}(e))$

$\wedge \text{Server}(e, x) \wedge \text{Served}(e, \text{Lunch}) \wedge (\text{ISA } (x, \text{Flight}))$

Eine Interpretation wären alle Instanziierungen der Subjektvariablen.

How can I go from Minneapolis to Long Beach?

$S \rightarrow \text{WhWord Aux NP VP}$

$\{ \text{WHQ (WhWord.sem VP.sem (NP.sem))} \}$

$\text{WHQ } (\text{How}, \exists e, \text{ISA } (e, \text{Going}) \wedge \text{Goer } (e, \text{User}))$

$\wedge \text{Origin}(e, \text{Minn.}) \wedge \text{Destination}(e, \text{LongBeach}))$

Eine Interpretation dieser Frage könnte im Anbieten von Reisemöglichkeiten bestehen (Flug, Zug, ...).

Nominalphrasen

- werden entweder als logische Terme oder als komplexe Terme repräsentiert.

Für viele Arten von NPs reicht ihre syntaktische Struktur für die semantische Analyse nicht aus, sondern es kann nur eine ‚vorläufige‘ Repräsentation erzeugt werden.

- Komposita

Flight schedule
Summer flight schedule

(Im Deutschen müssten diese Wortbildungen erst segmentiert werden)

Nominal $\rightarrow$ Noun ; Nominal $\rightarrow$ Noun Nominal

Das letzte Element solcher Konstruktionen wird von den davor stehenden modifiziert:

Nominal $\rightarrow$ Noun Nominal

$\{ \lambda x \text{ Nominal.sem}(x) \wedge \text{NN}(\text{Noun.sem}, x) \}$

Für die Beispiele oben ergibt sich folgende Darstellung:

$\lambda x \text{ ISA}(x, \text{Schedule}) \wedge \text{NN}(x, \text{Flight})$

$\lambda x \text{ ISA}(x, \text{Schedule}) \wedge \text{NN}(x, \text{Flight}) \wedge \text{NN}(x, \text{Summer})$

Nur für das letzte Element der Nomenkette wird ein Term gebildet, der durch die anderen durch die Relation NN modifiziert wird.

- Genitiv Nominalphrasen

Atlanta's airport, Maharani's menu

NP → ComplexDet Nominal

$\{ \langle \exists x \text{ Nominal.sem}(x) \wedge \text{GN}(x, \text{ComplexDet.sem}) \rangle \}$

ComplexDet → NP's {NP.sem}

Mit diesen Syntaxregeln und den Anhängen ergibt sich als semantische Repräsentation der komplexe Term:

$\langle \exists x \text{ ISA}(x, \text{Airport}) \wedge \text{GN}(x, \text{Atlanta}) \rangle$

Die abstrakte Relation GN könnte als Ortsangabe gelten.

- Attributive Adjektive

I don't mind a cheap restaurant.

Die syntaktische Darstellung könnte wie folgt aussehen:

Nominal → Adj Nominal

$\{ \lambda x \text{ Nominal.sem}(x) \wedge \text{ISA}(x, \text{Adj.sem}) \}$

Adj → cheap {Cheap}

Das Adjektiv wird als Prädikat repräsentiert und modifiziert die NP-Variable:

$\lambda x \text{ ISA}(x, \text{Restaurant}) \wedge \text{ISA}(x, \text{Cheap})$

Eine solche Repräsentation wird oft ‚Durchschnittssemantik‘ genannt (Intersective Semantics). D.h. letztlich, dass die Bedeutung als Durchschnitt der Menge der billigen Dinge und der Menge von Restaurants dargestellt wird.

Problematischer wird das für die folgenden Beispiele:

small elephant

former friend

fake gun

Das erste Beispiel müsste dann wie folgt aussehen:

$\lambda x \text{ ISA}(x, \text{Elephant}) \wedge \text{ISA}(x, \text{Small})$

Dies entspricht allerdings nicht der Bedeutung von relativen Adjektiven wie **small**, und gleiches gilt für die anderen beiden Beispiele.

Auf Grund der Syntax und den semantischen Anhängen bietet sich eine ähnliche Lösung an wie bei Komposita, d.h. man überlässt die Konkretisierung einer abstrakten Relation weiteren Interpretationsprozessen.

Nominal $\rightarrow$ Adj Nominal

$\{ \lambda x \text{ Nominal.sem}(x) \wedge \text{AM}(x, \text{Adj.sem}) \}$

Das anfängliche Beispiel würde dann so repräsentiert:

$\exists x \text{ ISA}(x, \text{Restaurant}) \wedge \text{AM}(x, \text{Cheap})$

Verbalphrasen

Die Semantik von Verbalphrasen erhält man durch den Lambda-Ausdruck des Verbs, für dessen Variablen die semantischen Anhänge der Argumentterme eingesetzt werden.

Dies geht nicht immer auf einfache Weise, z.B. bei eingebetteten Infinitivkonstruktionen:

I told Harry to go to Maharani.

(Syntaxstruktur auf nächster Seite)

Als semantische Struktur wäre wünschenswert:

$$\begin{aligned} \exists e, f, x \text{ ISA } (e, \text{Telling}) \wedge \text{ISA } (f, \text{Going}) \\ \wedge \text{Teller } (e, \text{Speaker}) \wedge \text{Tellee } (e, \text{Harry}) \\ \wedge \text{Toldthing } (e, f) \wedge \text{Goer } (f, \text{Harry}) \\ \wedge \text{Destination } (f, x) \end{aligned}$$

Harry taucht in den Rollen beider Ereignisse e und f auf, aber in der syntaktischen Struktur nur einmal.

Die Semantik der oberen VP mit Tell sollte so aussehen:

$$\begin{aligned} \lambda x, y, \lambda z \exists e \text{ ISA } (e, \text{Telling}) \\ \wedge \text{Teller } (e, z) \wedge \text{Tellee } (e, x) \wedge \text{ToldThing } (e, y) \end{aligned}$$

Dies bedeutet für die VP-Regel:

$$\begin{aligned} \text{VP} &\rightarrow \text{Verb NP VPto} \\ &\{ \text{Verb.sem } (\text{NP.sem}, \text{VPto.sem}) \} \end{aligned}$$

$$\text{VPto} \rightarrow \text{to VP} \quad \{ \text{VP.sem} \}$$

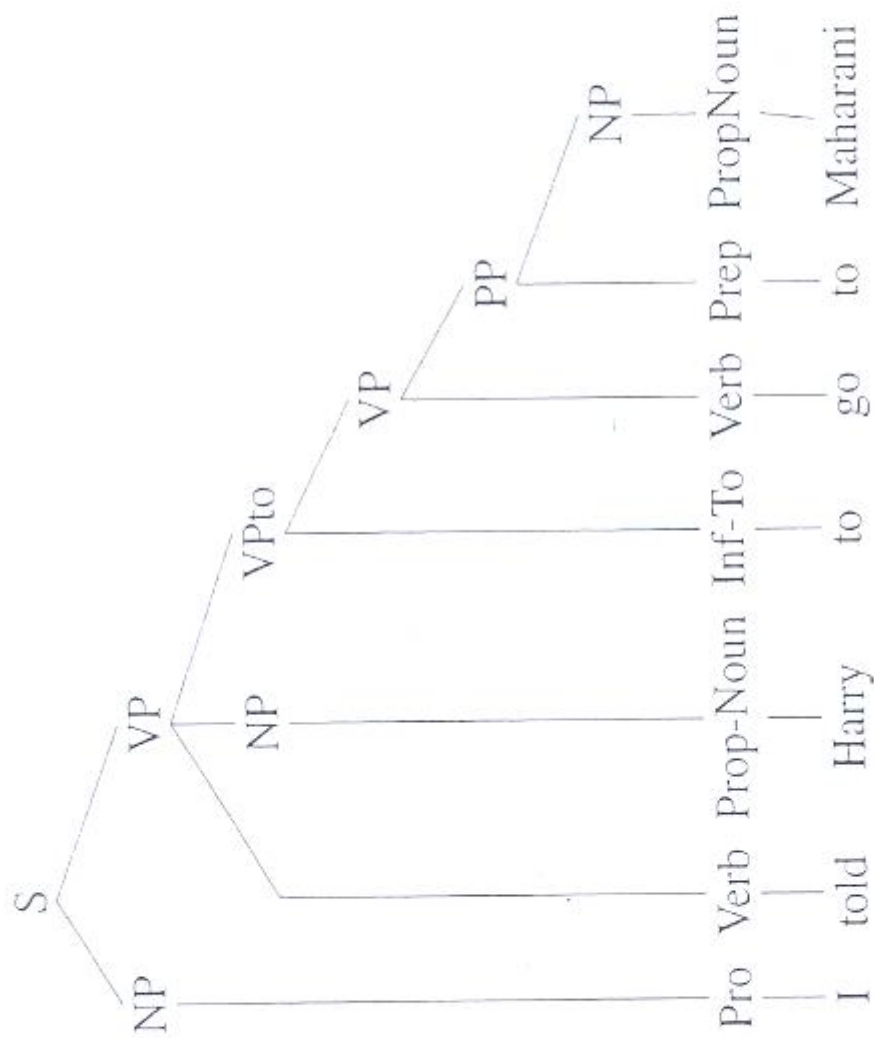


Figure 15.4 Parse tree for *I told Harry to go to Maharani*.

Hierdurch wird einerseits die **NP.sem** in die Subjektrolle der **VPto. sem** eingebunden, und das Ereignis der Infinitivkonstruktion **Going** wird zur **ToldThing**-Rolle des **Telling**.

Verb → tell

$\{ \lambda x, y$

λz

$\exists e, y.variable \text{ ISA } (e, \text{Telling})$

$\wedge \text{Teller } (e, z) \wedge \text{Tellee } (e, x)$

$\wedge \text{ToldThing } (e, y.variable) \wedge y(x))$

Die λ -Variable **x** steht für die **Tellee**-Rolle und gleichzeitig als Argument des Infinitivs.

y (x) führt Harry als **Goer**-Rolle innerhalb der **ToldThing**-Rolle ein.

y.variable führt das **Going**-Ereignis des Infinitivs ein.

Man beachte, dass hier Lambda-Ausdrücke auch als Argument für andere Lambda-Ausdrücke verwendet werden, obwohl es eigentlich nur logische Terme sein können.