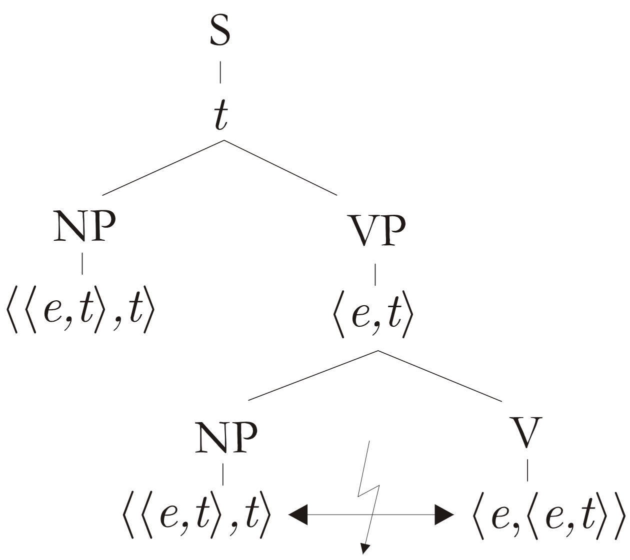
4.3 NPn als Objekte oder als Prädikative

4.3.1 Quantifizierende NPn in Objektposition

Steht eine quantifizierende NP nicht in Subjekt-, sondern in Objektposition, ist unter den bisher angenommenen Voraussetzungen keine semantische Komposition von NP und verbalem Ausdruck möglich.

Da weder der verbale Ausdruck auf die Objekt-NP noch die NP auf den verbalen Ausdruck funktional angewendet werden kann, besteht zwischen beiden ein Typenkonflikt, d.h. ein Konflikt mit Bezug auf ihren semantischen Typ.

Speziell im Falle eines transitiven Verbs ergibt sich das folgende Problem:



Für die Auflösung des Typenkonflikts gibt es zwei Möglichkeiten:

- Semantische Typanpassung des verbalen Ausdrucks;
- Syntaktische Anhebung der quantifizierenden NP.

Beide Verfahren bilden auch eine Basis dafür, um Skopusambiguitäten, die bei Sätzen mit zwei oder mehr quantifizierenden NPn auftreten, kompositional ableiten zu können.

Solche Sätze besitzen für jede ihrer möglichen Lesarten eine entsprechende semantische Repräsentation.

Beispiel:

Jeder Linguist liest ein Buch.

(a) **SR$_1$:** $\forall x[Linguist'(x) \rightarrow \exists y[Buch'(y) \wedge lesen'(y)(x)]]$

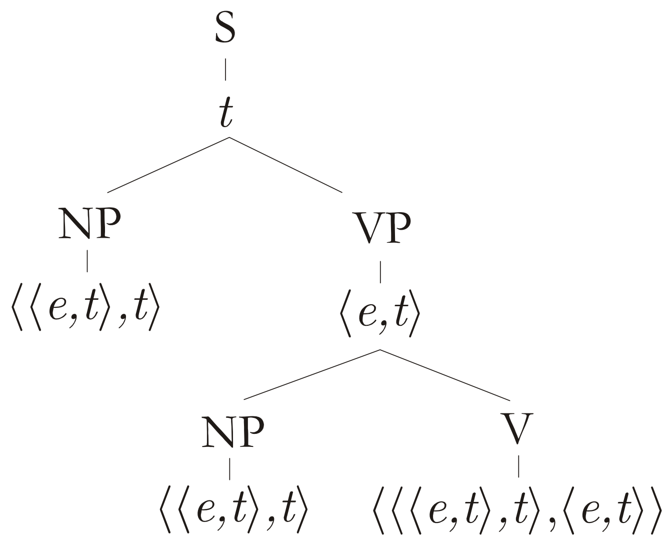
(b) **SR$_2$:** $\exists y[Buch'(y) \wedge \forall x[Linguist'(x) \rightarrow lesen'(y)(x)]]$

4.3.2 Semantische Typanpassung von verbalen Ausdrücken

Beim Verfahren der semantischen Typanpassung wird der Konflikt zwischen verbalem Ausdruck und quantifizierender NP in Objektposition dadurch aufgelöst, dass der semantische Typ des Ersteren an den der Letzteren angeglichen wird.

Im Ergebnis kann der verbale Ausdruck auf die Objekt-NP funktional appliziert werden.

Damit ein transitives Verb eine quantifizierende Objekt-NP als Argument nehmen und daraus eine VP vom Typ $\langle e,t\rangle$ bilden kann, muss das Verb ein Ausdruck vom Typ $\langle\langle\langle e,t\rangle,t\rangle,\langle e,t\rangle\rangle$ sein.



Wie die semantische Repräsentation eines Verbs auszusehen hat, das von diesem Typ ist, wird anhand von *lesen* demonstriert.

Beispiel:

Maria liest ein Buch.

Die VP *liest ein Buch* kann mit der NP *Maria* dann semantisch kombiniert werden, wenn sie die folgende Repräsentation hat:

SR(*liest ein Buch*)

$$= \lambda x \exists y [Buch'(y) \wedge lesen'(y)(x)]$$

Die Repräsentation ist das Ergebnis der funktionalen Anwendung einer Repräsentation T des Verbs auf die Repräsentation der Objekt-NP *ein Buch*.

SR(*ein Buch*)

$$= \lambda P \exists y [Buch'(y) \wedge P(y)]$$

Nach den bisherigen Voraussetzungen hat *lesen* die folgende Repräsentation vom Typ $\langle e,\langle e,t\rangle\rangle$:

SR(*lesen*)

$= \lambda y \lambda x[lesen'(y)(x)]$

Da *ein Buch* ein Ausdruck vom Typ $\langle\langle e,t\rangle,t\rangle$ ist, kann es sich bei T nicht um diese Repräsentation handeln.

Vielmehr muss T ein λ-Term sein, in dem über eine Variable $\mathcal{Q}$ vom Typ $\langle\langle e,t\rangle,t\rangle$ – sie steht für eine quantifizierende Objekt-NP – abstrahiert wird.

Genauer gesagt, muss als T und damit als Repräsentation von *lesen* ein λ-Term der folgenden Art angenommen werden:

SR(*lesen*)

$= \lambda \mathcal{Q}[...\mathcal{Q}...lesen'(y)(x)]$

Um die Variable Q in T auf einen Ausdruck vom Typ $\langle e,t\rangle$ applizieren zu können, muss bei $lesen'(y)(x)$ – einem Ausdruck vom Typ t – über eines der beiden Argumente x und y abstrahiert werden.

Wir wählen das Objektargument y, da das Subjektargument x später benötigt wird.

Es entsteht die folgende, weiterhin nur vorläufige Repräsentation von *lesen*:

SR(*lesen*)

$$= \lambda Q[...Q(\lambda y[lesen'(y)(x)])]$$

Der Ausdruck $Q(\lambda y[lesen'(y)(x)])$ ist vom Typ t und enthält das Subjektargument x frei.

Über diese Variable müssen wir abschließend abstrahieren, um die gewünschte Repräsentation von *lesen* zu erhalten:

SR(*lesen*)

$= \lambda\mathcal{Q}[\lambda x[\mathcal{Q}(\lambda y[lesen'(y)(x)])]]$

$= \lambda\mathcal{Q}\lambda x[\mathcal{Q}(\lambda y[lesen'(y)(x)])]$

$\boxed{?}$ Zeige, dass der angegebene λ-Term tatsächlich vom geforderten Typ $\langle\langle\langle e,t\rangle,t\rangle,\langle e,t\rangle\rangle$ ist?

Mit Bezug auf die Entscheidung darüber, wie ein transitives Verb semantisch zu repräsentieren ist, gibt es nun drei mögliche Vorgehensweisen:

Variante A:

Als semantische Repräsentation des Verbs wird generell ein λ-Term vom Typ $\langle\langle\langle e,t\rangle,t\rangle,\langle e,t\rangle\rangle$ angenommen.

Nachteil dieses Vorgehens ist, dass damit Eigennamen und definite NPn, sobald sie in Objektposition auftreten, stets vom Typ e in den Typ $\langle\langle e,t\rangle,t\rangle$ angehoben werden müssen.

Variante B:

Als semantische Repräsentationen werden sowohl λ-Terme vom Typ $\langle\langle\langle e,t\rangle,t\rangle,\langle e,t\rangle\rangle$ als auch λ-Terme vom Typ $\langle e,\langle e,t\rangle\rangle$ angenommen.

Welche der Repräsentationen ein und desselben Verbs jeweils ausgewählt wird, hängt davon ab, ob die Objekt-NP vom Typ $\langle\langle e,t\rangle,t\rangle$, d.h. eine quantifizierende NP oder vom Typ e, d.h. ein Eigenname oder eine definite NP ist.

Der Nachteil ist, dass damit beliebige transitive Verben als polysem bezüglich ihres semantischen Typs behandelt werden müssen.

Variante C:

Als die grundlegende semantische Repräsentation wird ein λ-Term vom Typ $\langle e,\langle e,t\rangle\rangle$ angesehen.

Bei Bedarf, d.h. im Falle eines Typenkonflikts mit einer quantifizierenden NP, wird diese Repräsentation mit einem Operator der semantischen Typverschiebung in eine Repräsentation vom Typ $\langle\langle\langle e,t\rangle,t\rangle,\langle e,t\rangle\rangle$ überführt.

Um einen passenden Operator angeben zu können, wird die Ableitung einer solchen Repräsentation – etwas vereinfacht – wie folgt rekonstruiert:

$\lambda y \lambda x[Verb'(y)(x)]$	grundlegende semantische Repräsentation
$\lambda y \lambda x[Verb'(y)(x)](y)$ $= \lambda x[Verb'(y)(x)]$	(Applikation auf ein provisorisches Objektargument)
$\lambda x[Verb'(y)(x)](x)$ $= Verb'(y)(x)$	(Applikation auf ein provisorisches Subjektargument)
$\lambda y[Verb'(y)(x)]$	(Abstraktion über die Objektvariable)
$\mathcal{Q}(\lambda y[Verb'(y)(x)])$	(Applikation einer Variablen für quantifizierende NPn)
$\lambda x[\mathcal{Q}(\lambda y[Verb'(y)(x)])]$	(Abstraktion über die Subjektvariable)
$\lambda \mathcal{Q} \lambda x[\mathcal{Q}(\lambda y[Verb'(y)(x)$	(Abstraktion über die Variable für quantifizierende NPn) abgeleitete semantische Repräsentation

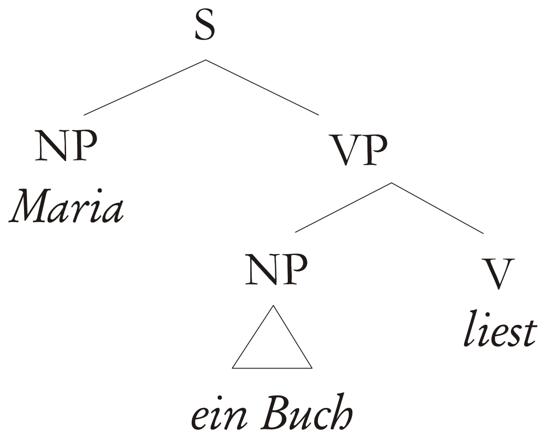
Diese Einzeloperationen können durch Anwendung des folgenden Operators in einem Schritt realisiert werden:

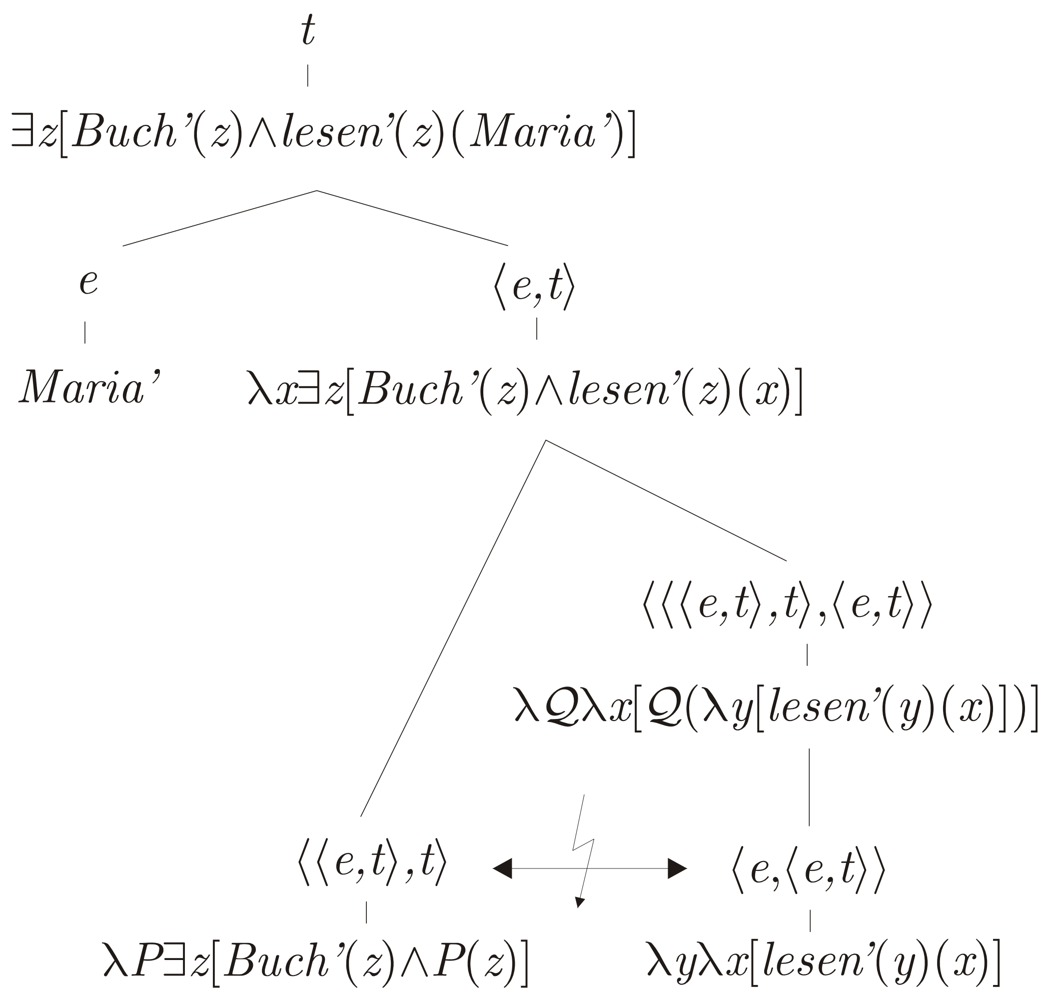
- $ARG_{\langle \underline{e},\langle e,t\rangle\rangle \rightarrow \left\langle \underline{\langle\langle e,t\rangle,t\rangle},\langle e,t\rangle\right\rangle}$: $\lambda R \lambda \mathcal{Q} \lambda x[\mathcal{Q}(\lambda y[R(y)(x)])]$

$\boxed{?}$ Bestimme den Typ des λ-Terms.

Mit dem Operator wird dabei faktisch der Typ e der Objekt-Argumentstelle eines transitiven Verbs zum Typ $\langle e,\langle e,t\rangle\rangle$ angehoben.

Beispiel:


S
NP
Maria
VP
NP
ein Buch
V
liest


t
$\exists z[Buch'(z) \wedge lesen'(z)(Maria')]$
e
$Maria'$
$\langle e,t \rangle$
$\lambda x \exists z[Buch'(z) \wedge lesen'(z)(x)]$
$\langle\langle\langle e,t \rangle,t \rangle,\langle e,t \rangle\rangle$
$\lambda \mathcal{Q} \lambda x[\mathcal{Q}(\lambda y[lesen'(y)(x)])]$
$\langle\langle e,t \rangle,t \rangle$
$\lambda P \exists z[Buch'(z) \wedge P(z)]$
$\langle e,\langle e,t \rangle\rangle$
$\lambda y \lambda x[lesen'(y)(x)]$

<u>Beispiel:</u> **$\mathbf{SR_1}$:** $\forall x[\mathit{Linguist}'(x) \rightarrow \exists y[\mathit{Buch}'(y) \wedge \mathit{lesen}'(y)(x)]]$

t

$\forall x[\mathit{Linguist}'(x) \rightarrow \exists y[\mathit{Buch}'(y) \wedge \mathit{lesen}'(y)(x)]]$

$\langle\langle e,t\rangle,t\rangle$ $\qquad$ $\langle e,t\rangle$

$\lambda P \forall x[\mathit{Linguist}'(x) \rightarrow P(x)]$ $\qquad$ $\lambda x \exists y[\mathit{Buch}'(y) \wedge \mathit{lesen}'(y)(x)]$

$\langle\langle\langle e,t\rangle,t\rangle,\langle e,t\rangle\rangle$

$\lambda \mathcal{Q} \lambda x[\mathcal{Q}(\lambda y[\mathit{lesen}'(y)(x)])]$

$\langle\langle e,t\rangle,t\rangle \longleftrightarrow \langle e,\langle e,t\rangle\rangle$

$\lambda P \exists y[\mathit{Buch}'(y) \wedge P(y)]$ $\qquad$ $\lambda y \lambda x[\mathit{lesen}'(y)(x)]$

Die vorgeschlagene Analyse liefert für transitive Verben mit zwei quantifizierenden NPn immer nur die Repräsentation jener Lesart, bei der die NP in Objektposition engen Skopus hat.

Um auch die Repräsentation der anderen Lesart abzuleiten, in der die quantifizierende NP in Objektposition weiten Skopus, d.h. Skopus über die quantifizierende NP in Subjektposition hat, wird eine zusätzliche Regel benötigt.

Die Regel des Hineinquantifizierens, die von Montague in *PTQ* vorgeschlagen worden ist, erlaubt es, für einen beliebigen Satz mit mehr als einer quantifizierenden NP jede seiner möglichen Repräsentationen kompositional herzuleiten

Regel des Hineinquantifizierens (‚Quantifying In')

Wenn ein Satz S durch eine Formel ϕ repräsentiert wird, die einen Ausdruck α vom Typ $\langle\langle e,t\rangle,t\rangle$ enthält, dann kann S auch durch $\alpha(\lambda x[\phi'])$ repräsentiert werden, wobei ϕ' wie ϕ ist, außer dass α durch den Ausdruck $\lambda P[P(x)]$ ersetzt wird und x eine Variable vom Typ e ist, die in ϕ' frei ist.

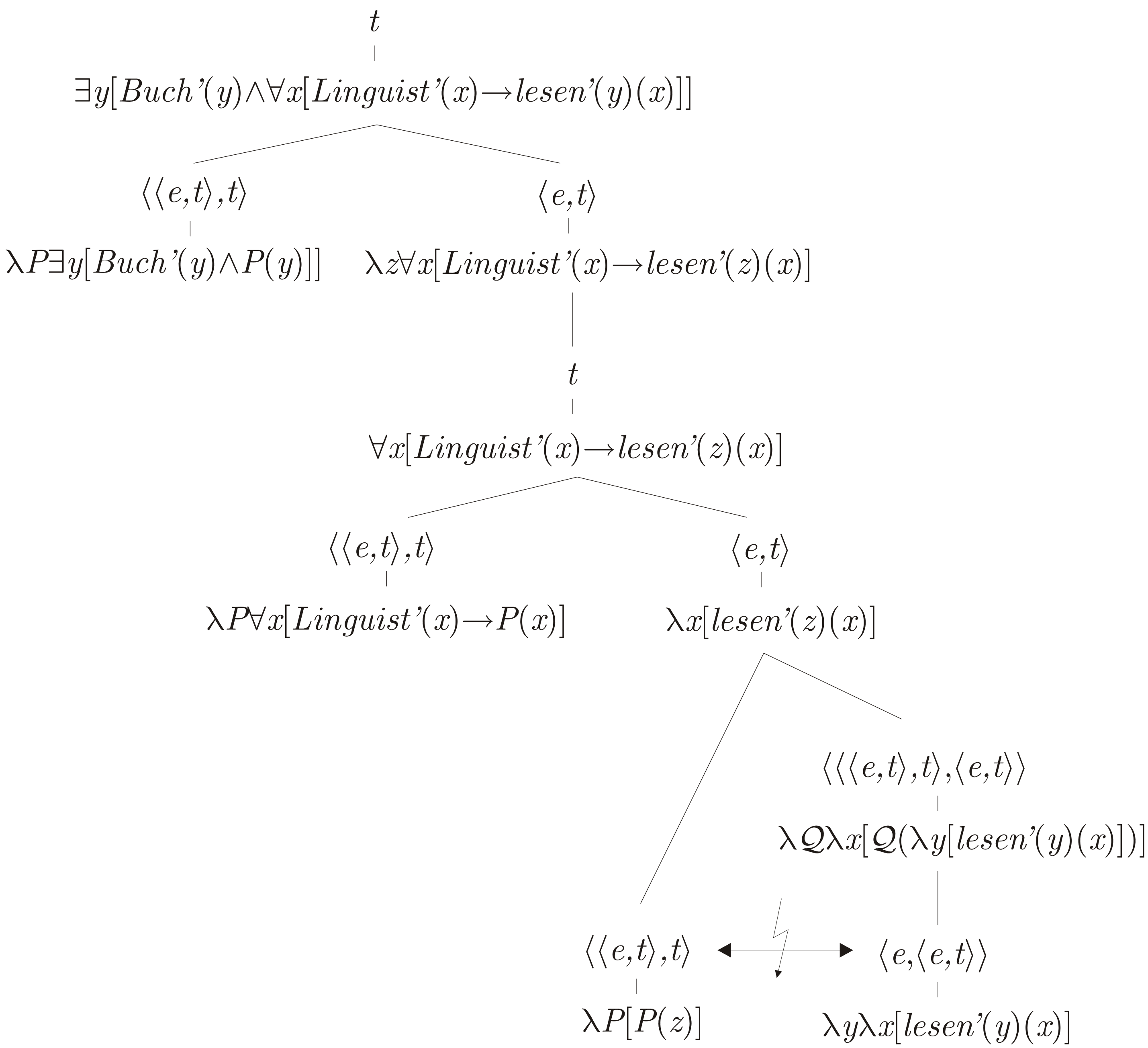
Bei der Ableitung der Repräsentation eines Satzes mit zwei quantifizierenden NPn kann also an die Stelle der Repräsentation α der Objekt-NP der λ-Term $\lambda P[P(x)]$ eingesetzt werden.

Nach Integration der Repräsentation der Subjekt-NP wird über die Objektvariable x abstrahiert, wodurch ein Ausdruck entsteht, auf den α appliziert werden kann.

Im Ergebnis erhält man eine Repräsentation jener Lesart des Satzes, bei der die Subjekt-NP engen Skopus hat.

<u>Beispiel:</u>

SR$_2$: $\exists y[Buch'(y) \wedge \forall x[Linguist'(x) \rightarrow lesen'(y)(x)]]$



4.3.3 Syntaktische Anhebung von quantifizierenden NPn

Eine Alternative stellt das syntaktische Verfahren der **Quantorenanhebung** (des ‚Quantifier Raising', abgekürzt: QR) dar, das auf Überlegungen von Noam Chomsky im Rahmen des GB-Modells der Grammatik (*Government and Binding*, 1981) zurückgeht.

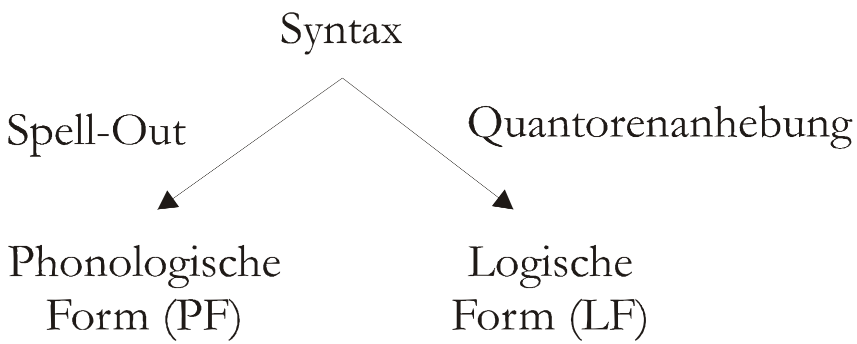
Grundlage für dieses Verfahren ist die Unterscheidung von zwei speziellen syntaktischen Strukturebenen – der Ebene der Phonologischen Form **PF** und der Logischen Form **LF**.

Im GB-Modell gehen sowohl **PF** als auch **LF** aus der Oberflächenstruktur **SS** (‚Surface Structure') hervor, und diese wiederum geht aus der Tiefenstruktur **DS** (‚Deep Structure') hervor.

LF ist die Bezugsebene für die semantische Repräsentation **SR** – bei der Strategie der indirekten Interpretation – bzw. für die modelltheoretische Interpretation – bei der Strategie der direkten Interpretation.

$$
\begin{tikzcd}
 & \textbf{DS} \arrow[d] & \\
\textbf{PF} & \textbf{SS} \arrow[l] \arrow[r] & \textbf{LF}
\end{tikzcd}
$$

In der minimalistischen Syntax sieht die Anordnung von **PF** und **LF** etwa folgendermaßen aus:



Bei der Ausformulierung der **LF** eines Satzes werden dessen quantifizierenden NPn – die Quantoren (‚Quantifiers') – aus ihren ursprünglichen syntaktischen Positionen herausgehoben.

Diese syntaktische Bewegung der Quantorenanhebung spielt nur für die Interpretation des Satzes eine Rolle, wird also phonologisch nicht realisiert.

Für einen Satz mit zwei quantifizierenden NPn gibt es zwei mögliche Reihenfolgen der Quantorenanhebung und damit zwei mögliche **LF-**Strukturen, die jeweils unterschiedliche Skopusverhältnisse der NPn repräsentieren.

Dabei hat eine NP *a* syntaktischen Skopus über eine NP *b* gdw *a* die Konstituente *b* c-kommandiert.

? Wie wird die syntaktische Relation des c-Kommandos definiert?

Zwei Möglichkeiten der Quantorenanhebung bei zwei quantifizierenden NPn:

- 1. Möglichkeit:

  Die quantifizierende NP in Objektposition wird zunächst über die quantifizierende NP in Subjektpostion hinwegbewegt, d.h. an S adjungiert.

  Danach wird die Subjekt-NP ihrerseits über die Objekt-NP hinwegbewegt und damit an den vorher etablierten S-Knoten adjungiert.

  In diesem Fall hat die Subjekt-NP weiten syntaktischen Skopus, d.h. Skopus über die Objekt-NP.

- 2. Möglichkeit:

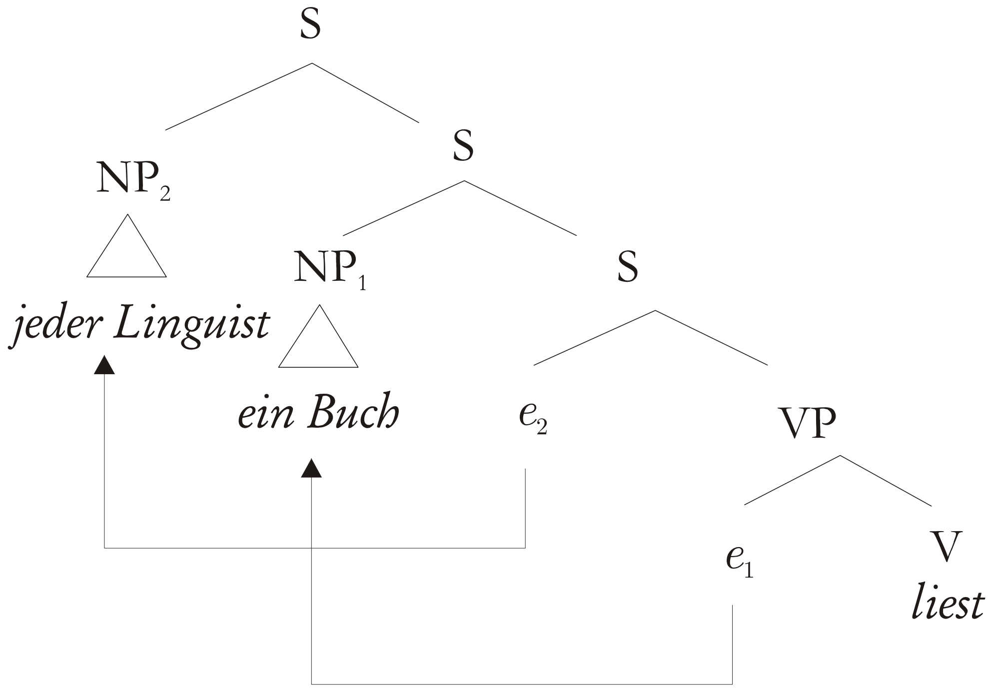
  Es wird in umgekehrter Reihenfolge vorgegangen, d.h. zunächst wird die Subjekt-NP an S adjungiert und danach wird die Objekt-NP über die Subjekt-NP hinwegbewegt.

  In diesem Fall hat also die Objekt-NP syntaktischen Skopus über die Subjekt-NP.

Jede bewegte NP hinterlässt eine Spur e_i, die mit der jeweiligen NP koindiziert ist.

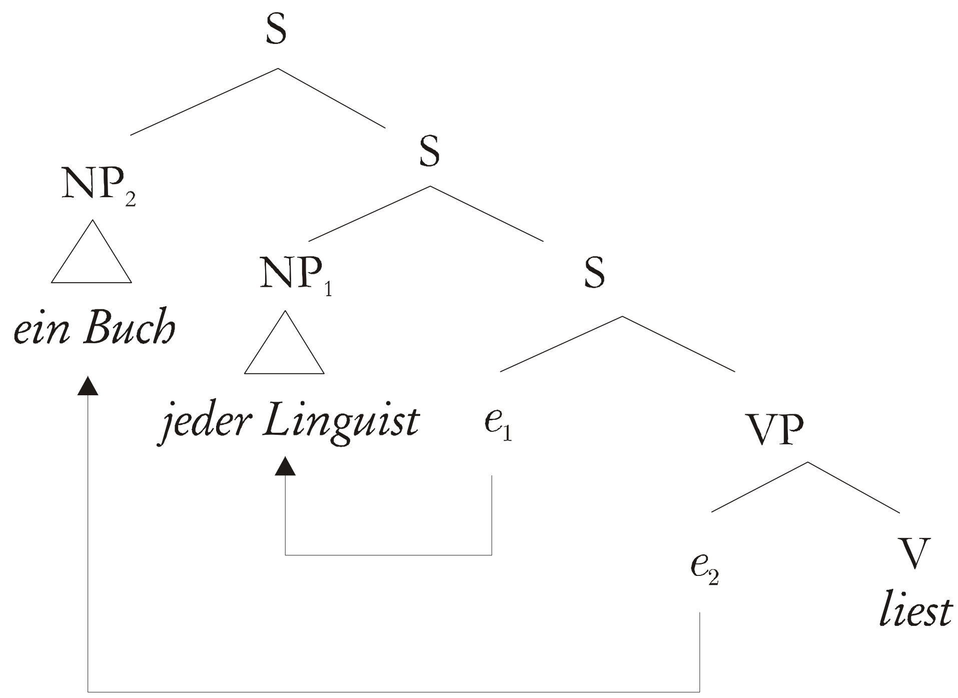
Beispiel: *Jeder Linguist liest ein Buch.*

LF$_1$:



$[_S[\textit{jeder Linguist}]_2\ [_S[\textit{ein Buch}]_1\ [_S\ e_2\ [e_1\ \textit{liest}]]]]$

LF$_2$:



$[_S[$*ein Buch*$]_2\ [_S[$*jeder Linguist*$]_1\ [_S\ e_1\ [e_2$ *liest*$]]]]$

Wenn die Konstituenten der jeweiligen **LF** semantisch repräsentiert werden, erhält man entsprechend die Repräsentationen der beiden möglichen Lesarten.

Dabei werden die koindizierten Spuren als Variablen, die Indizes an den bewegten NPn als λ-Präfixe über die jeweilige Variable repräsentiert.

LF$_1$: $[_S[$*jeder Linguist*$]_2\ [_S[$*ein Buch*$]_1\ [_S\ e_2\ [e_1$ *liest*$]]]]$

SR$_1$:

$$\lambda P\forall x[Linguist'(x) \rightarrow P(x)](\lambda x_2[\lambda P\exists y[Buch'(y) \wedge P(y)](\lambda x_1[\ \lambda y\lambda x[lesen'(y)(x)](x_1)(x_2)])])$$

$$= \lambda P\forall x[Linguist'(x) \rightarrow P(x)](\lambda x_2[\lambda P\exists y[Buch'(y) \wedge P(y)](\lambda x_1[\ lesen'(x_1)(x_2)])])$$

$$= \lambda P\forall x[Linguist'(x) \rightarrow P(x)](\lambda x_2[\exists y[Buch'(y) \wedge lesen'(y)(x_2)]])$$

$$= \forall x[Linguist'(x) \rightarrow \exists y[Buch'(y) \wedge lesen'(y)(x)]]$$

LF$_2$: $[_S[$*ein Buch*$]_2\ [_S[$*jeder Linguist*$]_1\ [_S\ e_1\ [e_2$ *liest*$]]]]$

SR$_2$:

$$\lambda P \exists y[Buch'(y) \wedge P(y)](\lambda x_2[\lambda P \forall x[Linguist'(x) \rightarrow P(x)](\lambda x_1[\ \lambda y \lambda x[lesen'(y)(x)](x_2)(x_1)])])$$

$$= \lambda P \exists y[Buch'(y) \wedge P(y)](\lambda x_2[\lambda P \forall x[Linguist'(x) \rightarrow P(x)](\lambda x_1[\ lesen'(x_2)(x_1)])])$$

$$= \lambda P \exists y[Buch'(y) \wedge P(y)](\lambda x_2 \forall x[Linguist'(x) \rightarrow lesen'(x_2)(x)])]$$

$$= \exists y[Buch'(y) \wedge \forall x[Linguist'(x) \rightarrow lesen'(y)(x)]]$$

Die beiden Ansätze – semantische Typanpassung bzw. Hineinquantifizieren und syntaktische Quantorenanhebung – führen zum selben Ergebnis und sind damit strukturell gleich.

Sie unterscheiden sich darin, ob man die zugrundeliegende Idee als semantische Konstruktion realisiert oder aber als syntaktischen Prozess implementiert.

4.3.4 Indefinite und definite NPn in Kopula-Prädikativ-Konstruktionen

Indefinite NPn unterscheiden sich dadurch von anderen quantifizierenden NPn, dass sie die Rolle eines Prädikativs übernehmen können.

Gemäß unseren Voraussetzungen muss die betreffende Kopula-Prädikativ-Konstruktion vom Typ $\langle e,t \rangle$ sein, um zusammen mit der Subjekt-NP einen Ausdruck vom Typ t bilden zu können.

Beispiele:

(1) *Cäsar ist ein Hund.*

$\mathbf{SR}([_{S} [_{NP}\ \mathit{Cäsar}][_{VP} [_{Cop}\ \mathit{ist}][_{NP} [_{Det}\ \mathit{ein}][_{N}\ \text{Hund}]]]])$

$= \mathbf{SR}([_{VP} [_{Cop}\ \mathit{ist}][_{NP} [_{Det}\ \mathit{ein}][_{N}\ \text{Hund}]]])\ (\mathbf{SR}(\mathit{Cäsar}))$

$= \ldots$

$= Hund'(Cäsar')$

(2) *Jeder Dackel ist ein Hund.*

$$\mathbf{SR}([_{S}\,[_{NP}\,[_{Det}\,\textit{Jeder}][_{N}\,\textit{Dackel}]][_{VP}\,[_{Cop}\,\textit{ist}][_{NP}\,[_{Det}\,\textit{ein}][_{N}\,\text{Hund}]]]])$$

$$= \mathbf{SR}([_{NP}\,[_{Det}\,\textit{Jeder}][_{N}\,\textit{Dackel}]])\ (\mathbf{SR}([_{VP}\,[_{Cop}\,\textit{ist}][_{NP}\,[_{Det}\,\textit{ein}][_{N}\,\text{Hund}]]]))$$

$$= \ldots$$

$$= \forall x[Dackel'(x) \rightarrow Hund'(x)]$$

Wenn die Kopula eine identische Funktion denotiert und deshalb vom Typ eines Prädikatsmodifikators $\langle\langle e,t\rangle,\langle e,t\rangle\rangle$ ist, ergibt sich ein Typenkonflikt mit der im Prädikativ vorkommenden indefiniten NP.

Von Barbara Partee (1987) stammt der Vorschlag, diesen Konflikt mit Hilfe eines weiteren Operators der semantischen Typverschiebung aufzulösen.

- *BE*: $\lambda\mathcal{P}\lambda x[\mathcal{P}(\lambda y[x = y])]$

$\boxed{?}$ Bestimme den Typ des λ-Terms.

Durch funktionale Applikation von *BE* auf die prädikativ gebrauchte indefinite NP und unter Nutzung der logischen Äquivalenz $\exists x[P(x) \wedge x = y] \Leftrightarrow P(y)$ wird der Typ der NP von $\langle\langle e,t\rangle,t\rangle$ in $\langle e,t\rangle$ verändert.

Beispiel: *Cäsar ist ein Hund.*

SR([$_S$ [$_{NP}$ *Cäsar*][$_{VP}$ [$_{Cop}$ *ist*][$_{NP}$ [$_{Det}$ *ein*][$_N$ *Hund*]]]])

= **SR**([$_{VP}$ [$_{Cop}$ *ist*][$_{NP}$ [$_{Det}$ *ein*][$_N$ *Hund*]]]) (**SR**(*Cäsar*))

= **SR**(*ist*) (*BE* (**SR**([$_{NP}$ [$_{Det}$ *ein*][$_N$ *Hund*]]))) (**SR**(*Cäsar*))

= **SR**(*ist*) (*BE* (**SR**(*ein*) (**SR**(*Hund*))) (**SR**(*Cäsar*))

$= \lambda P \lambda x\,[P(x)](\lambda \mathcal{P} \lambda x[\mathcal{P}(\lambda y[x = y])](\lambda Q \lambda P \exists x[Q(x) \wedge P(x)](\lambda x[Hund'(x)])))(Cäsar')$

$= \lambda P \lambda x\,[P(x)](\lambda \mathcal{P} \lambda x[\mathcal{P}(\lambda y[x = y])](\lambda P \exists x[Hund'(x) \wedge P(x)]))(Cäsar')$

$= \lambda P \lambda x\,[P(x)](\lambda z[\lambda P \exists x[Hund'(x) \wedge P(x)](\lambda y[z = y])])(Cäsar')$

$= \lambda P \lambda x\,[P(x)](\lambda z[\exists x[Hund'(x) \wedge \lambda y[z = y](x)]])(Cäsar')$

$= \lambda P \lambda x\,[P(x)](\lambda z \exists x[Hund'(x) \wedge z = x])(Cäsar')$

$= \lambda P \lambda x\,[P(x)](\lambda z[Hund'(z)])(Cäsar')$, wegen $\lambda z \exists x[Hund'(x) \wedge z = x] = \lambda z[Hund'(z)]$

$= \lambda z[Hund'(z)](Cäsar')$

$= Hund'(Cäsar')$

$\boxed{?}$ Warum erfolgt in der 7. Zeile der Ableitung eine gebundene Umbenennung von x in z?

Auch definite NPn können als Prädikative auftreten.

Mit Sätzen dieser Art werden **identifizierende** Aussagen gemacht.

Beispiel:

Cäsar ist der Dieb.

SR(*Cäsar ist der Dieb*)

$= (Cäsar' = \iota x[Dieb'(x)])$

Werden definite NPn – wie im Anschluss an Montague – als Ausdrücke vom Typ $\langle\langle e, t\rangle, t\rangle$ behandelt, kann der Typenkonflikt zwischen der prädikativ gebrauchten NP und der Kopula wiederum unter Verwendung des Operators *BE* aufgelöst werden.

Für den Fall, dass definite NPn als Ausdrücke vom Typ e aufgefasst werden, schlägt Partee (1987) die Anwendung des folgenden Operators der semantischen Typverschiebung vor:

- *ident*: $\lambda y \lambda x[x = y]$

Mit *ident* wird ein Ausdruck vom Typ e in einen Ausdruck vom Typ $\langle e, t\rangle$ überführt.

Beispiel:

Cäsar ist der Dieb.

$$\mathbf{SR}([_{\mathrm{S}}\,[_{\mathrm{NP}}\,\mathit{Cäsar}][_{\mathrm{VP}}\,[_{\mathrm{Cop}}\,\mathit{ist}][_{\mathrm{NP}}\,[_{\mathrm{Det}}\,\mathit{der}][_{\mathrm{N}}\,\mathit{Dieb}]]]])$$

$$= \mathbf{SR}([_{\mathrm{VP}}\,[_{\mathrm{Cop}}\,\mathit{ist}][_{\mathrm{NP}}\,[_{\mathrm{Det}}\,\mathit{der}][_{\mathrm{N}}\,\mathit{Dieb}]]])\ (\mathbf{SR}(\mathit{Cäsar}))$$

$$= \mathbf{SR}(\mathit{ist})\ (\mathit{ident}\ (\mathbf{SR}([_{\mathrm{NP}}\,[_{\mathrm{Det}}\,\mathit{der}][_{\mathrm{N}}\,\mathit{Dieb}]])))\ (\mathbf{SR}(\mathit{Cäsar}))$$

$$= \mathbf{SR}(\mathit{ist})\ (\mathit{ident}\ (\mathbf{SR}(\mathit{der})\ (\mathbf{SR}(\mathit{Dieb}))))\ (\mathbf{SR}(\mathit{Cäsar}))$$

$$= \lambda P \lambda x\big[P(x)\big](\lambda y \lambda x[x = y](\lambda P \iota x[P(x)](\lambda x[Dieb'(x)])))(Cäsar')$$

$$= \lambda P \lambda x\big[P(x)\big](\lambda y \lambda x[x = y](\iota x[Dieb'(x)]))(Cäsar')$$

$$= \lambda P \lambda x\big[P(x)\big](\lambda x[x = \iota x[Dieb'(x)]])(Cäsar')$$

$$= (\lambda x[x = \iota x[Dieb'(x)]])(Cäsar')$$

$$= (Cäsar' = \iota x[Dieb'(x)])$$

Übungen

Ü4.5 – Ü4.10

Termin: **nächstes Tutorium**