

Lange Abhängigkeiten in Brosziewski-Derivationen

Christina Unger
(UiL-OTS, Universiteit Utrecht)
C.Unger@uu.nl

GGs, Leipzig, 22.–24. Mai 2009



W-Bewegung

- in einer derivationalen Syntax
- mit strikt lokalen Operationen,
- so wenig syntaktischer Struktur wie möglich
- und Greed-basiert (d.h. Bewegung wird vom bewegenden Ausdruck ausgelöst)

↔ Version syntaktischer Derivationen à la Brosziewski (2003)
(ziemlich ähnlich auch zu Stroik (2009))



Überblick

- 1 Zutaten
- 2 Ausdrücke
- 3 Operationen
- 4 Mehrfache W-Fragen
- 5 W-Inseln



Zutaten: Zeichenketten

Wir beginnen mit Zeichenketten (als grobe Darstellung phonologischen Gehalts) und Funktionen über Zeichenketten.

- John
- $\lambda x \lambda y. y \text{ chased } x$
- zombie
- $\lambda x. a \ x$

Verkettung ist einfach funktionale Applikation (@).

- $(\lambda x \lambda y. y \text{ chased } x) @ \text{John} = \lambda y. y \text{ chased John}$
- $(\lambda x. a \ x) @ \text{zombie} = a \ \text{zombie}$
- $(\lambda y. y \text{ chased John}) @ a \ \text{zombie} = a \ \text{zombie chased John}$



Zutaten: Syntaktische Typen

Jeder Zeichenkette wird ein syntaktischer Typ zugeordnet.

$$\begin{aligned}(\lambda x \lambda y. y \text{ chased } x) &:: NP \rightarrow NP \rightarrow VP @ \text{John} :: NP \\ &= \lambda y. y \text{ chased John} :: NP \rightarrow VP\end{aligned}$$

$$\begin{aligned}(\lambda x. a \ x) &:: N \rightarrow NP @ \text{zombie} :: N \\ &= a \ \text{zombie} :: NP\end{aligned}$$

$$\begin{aligned}(\lambda y. y \text{ chased John}) &:: NP \rightarrow VP @ a \ \text{zombie} :: NP \\ &= a \ \text{zombie chased John} :: VP\end{aligned}$$

Syntaktische Typen steuern die Verkettung von Ausdrücken.



Zutaten: Merkmale

Für lange Abhängigkeiten fügen wir Merkmale hinzu.

Merkmale kommen in zwei Ausprägungen: als Sondenmerkmale $\bullet f$ und als Zielmerkmale f .

Sie drücken aus, dass der Ausdruck, der sie trägt, in einer bestimmten Konfiguration mit einem Ausdruck vorkommen muss, der das passende andere Merkmal trägt. (Diese Konfiguration wird strikt lokal sein.)

Merkmale lösen Bewegung aus.



Ausdrücke (einfach)

Einfache Ausdrücke sind Zeichenketten (und Funktionen darüber) zusammen mit einer ungeordneten Liste von Merkmalen.

- $\text{John}^{\emptyset} :: NP$
- $\text{who}^{wh} :: NP$

C-Projektion:

- $(\lambda x.\text{that } x)^{\emptyset} :: VP \rightarrow CP$
- $(\lambda x.x)^{\bullet wh} :: VP \rightarrow CP$

$\lambda x.\text{that } x :: VP \rightarrow CP$ @ a zombie chased John :: VP
= that a zombie chased John :: CP



Verkettung einfacher Ausdrücke

- Einfache Ausdrücke werden zu einfachen Ausdrücken kombiniert.
- Informationen über Teilausdrücke, Struktur, derivationelle Geschichte, etc. werden vergessen.
- Es wird demnach keine Phrasenstruktur gebaut, sondern nur so viel Struktur behalten, wie unbedingt notwendig ist.

Definition

$$(M1) \text{ merge } \alpha^F \beta^\emptyset = (\alpha @ \beta)^F$$

$$\begin{aligned} \text{merge } \lambda x.a \ x :: N &\rightarrow NP \text{ zombie} :: N \\ &= \lambda x.a \ x :: N \rightarrow NP @ \text{ zombie} :: N \\ &= a \text{ zombie} :: NP \end{aligned}$$



Aber...

...das funktioniert nur solange keine Merkmale involviert sind. Trägt ein Ausdruck nicht überprüfte Merkmale, kann er nicht einfach vergessen werden.

$$\text{merge } (\lambda x \lambda y. y \text{ chased } x) \text{ who}^{wh} \\ \neq \lambda y. y \text{ chased who}$$

Um Informationen zu behalten, die noch benötigt werden, bilden wir komplexe Ausdrücke.

Definition

$$\text{merge } x \ y^F = \text{merge } x \ (\text{split } y^F) \text{ falls } F \neq \emptyset$$



Ausdrücke (komplex)

split spaltet einen einfachen Ausdruck in einen komplexen Ausdruck (ein geordnetes Paar von Ausdrücken) auf. Die Merkmale werden mit dem ersten Element assoziiert, der phonologische Gehalt mit dem ersten oder dem zweiten.

Definition

split $\beta^F :: \tau = \langle \beta^F, \epsilon :: \tau \rangle$ oder $\langle \epsilon^F, \beta :: \tau \rangle$

merge $(\lambda x \lambda y. y \text{ chased } x) \text{ who}^{wh} :: NP$
 $=$ **merge** $(\lambda x \lambda y. y \text{ chased } x) (\text{split } \text{who}^{wh} :: NP)$
 $=$ **merge** $(\lambda x \lambda y. y \text{ chased } x) \langle \text{who}^{wh}, \epsilon :: NP \rangle$



Verkettung komplexer Ausdrücke

Das erste Element eines Paares (der *Rand*) wird mitgeschleppt, das zweite Element (der *Nukleus*) wird wie gewohnt verkettet.

Definition

$$(M2) \quad \mathbf{merge} \langle x, y \rangle \beta = \langle x, \mathbf{merge} y \beta \rangle$$

$$(M3) \quad \mathbf{merge} x \langle y, z \rangle = \langle y, \mathbf{merge} x z \rangle$$

$$= \mathbf{merge} (\lambda x \lambda y. y \text{ chased } x) :: NP \rightarrow NP \rightarrow VP \langle \text{who}^{wh}, \epsilon :: NP \rangle$$

$$= \langle \text{who}^{wh}, \mathbf{merge} (\lambda x \lambda y. y \text{ chased } x) :: NP \rightarrow NP \rightarrow VP, \epsilon :: NP \rangle$$

$$= \langle \text{who}^{wh}, (\lambda x \lambda y. y \text{ chased } x) :: NP \rightarrow NP \rightarrow VP @ \epsilon :: NP \rangle$$

$$= \langle \text{who}^{wh}, \lambda y. y \text{ chased } :: NP \rightarrow VP \rangle$$



Verkettung komplexer Ausdrücke

Der komplexe Ausdruck

$$\langle \alpha_1, \langle \alpha_2, \dots, \langle \alpha_n, \beta \rangle \rangle \rangle$$

kann angesehen werden als der Ausdruck β , dessen Derivation die Ausdrücke $\alpha_1, \dots, \alpha_n$ involvierte, die allesamt noch Merkmale abzugleichen haben. D.h. $\alpha_1, \dots, \alpha_n$ wurden extrahiert – sie werden mitgeschleppt bis sie ihre Merkmale abgleichen können. Bis dahin spielen sie keine Rolle. Der gesamte Ausdruck verhält sich syntaktisch wie β .



Und so fort

$$\begin{aligned} & \text{merge } \langle \text{who}^{wh}, \lambda y.y \text{ chased} :: NP \rightarrow VP \rangle \text{ a zombie} :: NP \\ &= \langle \text{who}^{wh}, \text{merge } \lambda y.y \text{ chased} :: NP \rightarrow VP \text{ a zombie} :: NP \rangle \\ &= \langle \text{who}^{wh}, \text{a zombie chased} :: VP \rangle \end{aligned}$$
$$\begin{aligned} & \text{merge } (\lambda x.x)^{\bullet wh} :: VP \rightarrow CP \langle \text{who}^{wh}, \text{a zombie chased} :: VP \rangle \\ &= \langle \text{who}^{wh}, \text{merge } (\lambda x.x)^{\bullet wh} :: VP \rightarrow CP \text{ a zombie chased} :: VP \rangle \\ &= \langle \text{who}^{wh}, \text{a zombie chased}^{\bullet wh} :: CP \rangle \end{aligned}$$

Das ist schließlich die Konfiguration, in der das wh-Merkmal überprüft werden kann.



Remergere

In einer Konfiguration $\langle \alpha^f, x^{\bullet f} \rangle$ werden die Merkmale abgeglichen (gelöscht) und α wird mit dem Nukleus von x verkettet (+).

Definition

$$\text{remerge } \langle \alpha^{f,F}, x^{\bullet f} \rangle = \begin{cases} \alpha + \bar{x} & \text{falls } F = \emptyset \\ \langle \alpha^F, \bar{x} \rangle \text{ oder } \langle \epsilon^F, \alpha + \bar{x} \rangle & \text{sonst} \end{cases}$$

\bar{x} bezeichnet x , in dem alle Vorkommen von f gelöscht wurden.

$$\begin{aligned} & \text{remerge } \langle \text{who}^{wh}, \text{some zombie chased}^{\bullet wh} :: CP \rangle \\ & = \text{who some zombie chased} :: CP \end{aligned}$$



Mehrfache W-Fragen

- alle W-Ausdrücke in situ (z.B. Japanisch)

Taro-ga dare no sushi-o tabeta?
Taro-TOP who.NOM Q sushi-ACC ate
'Whose sushi did Taro eat?'

- alle W-Ausdrücke initial (z.B. Bulgarisch)

Koj₁ kogo₂ [__₁ vižda __₂]?
wer wen sieht
'Wer sieht wen?'

- genau ein W-Ausdruck initial, die anderen in situ (z.B. English)

Who_i does John think ___i will win which race?



Ausgangspunkt

Annahme

Alle W-Ausdrücke tragen ein Merkmal *wh*.

D.h. alle W-Ausdrücke werden gespalten und bewegt.

$$\text{split } w^{wh} = \langle w^{wh}, \epsilon \rangle \text{ oder } \langle \epsilon^{wh}, w \rangle$$

Die Remerge-Konfiguration hat in allen Sprachen die folgende Form:

$$\langle x^{wh}, \langle y^{wh}, z^{\bullet wh} :: CP \rangle \rangle$$

Sprachen unterscheiden sich nur darin, mit welchen Elementen der phonologische Gehalt assoziiert wurde.



Mehrfache W-Bewegung

Bulgarisch

split $\beta^F = \langle \beta^F, \epsilon \rangle$

Das gibt: $\langle \text{koj}^{wh}, \langle \text{kogo}^{wh}, \text{vižda}^{\bullet wh} \rangle \rangle$

Nach **remerge**: $\text{koj} + \langle \text{kogo}, \text{vižda} \rangle$

Annahme

Alle weiteren Ausdrücke, die ein Merkmal abgleichen, können verkettet werden. (Das entspricht der Annahme, das Bulgarisch mehrfache Spezifizierer zulässt.)

Das Resultat ist dann das erwünschte koj kogo vižda .



In situ

Japanisch

split $\beta^F = \langle \epsilon^F, \beta \rangle$

Das gibt: $\langle \epsilon^{wh}, \langle \epsilon^{wh}, \text{darega sushio tabetano}^{\bullet wh} \rangle \rangle$

Nach **remerge**: $\epsilon + \langle \epsilon, \text{darega sushio tabetano} \rangle$

Nun gebrauchen wir entweder wieder die Annahme über mehrfache Spezifizierer oder die folgende.

Annahme

Leere Ausdrücke (d.h. Ausdrücke ohne phonologischen Gehalt und ohne Merkmale) können gelöscht werden, da sie keinerlei Effekt haben (d.h. $\langle \epsilon^\emptyset, x \rangle = x$).



Einfache W-Bewegung

Sprachen mit einfacher W-Bewegung stellen eine Herausforderung dar.
Wir wollen z.B.: $\langle \text{who}^{wh}, \langle \epsilon^{wh}, \text{saw whom}^{\bullet wh} \rangle \rangle$.

Da wir keinerlei Möglichkeiten zum Look-ahead haben, müssen wir annehmen:

Englisch

split $\beta^F = \langle \epsilon^F, \beta \rangle$ oder $\langle \beta^F, \epsilon \rangle$

Das ergibt folgende Möglichkeiten:

- ① $\langle \text{who}^{wh}, \langle \text{whom}^{wh}, \text{saw}^{\bullet wh} \rangle \rangle$
- ② $\langle \epsilon^{wh}, \langle \text{whom}^{wh}, \text{who saw}^{\bullet wh} \rangle \rangle$
- ③ $\langle \text{who}^{wh}, \langle \epsilon^{wh}, \text{saw whom}^{\bullet wh} \rangle \rangle$
- ④ $\langle \epsilon^{wh}, \langle \epsilon^{wh}, \text{who saw whom}^{\bullet wh} \rangle \rangle$



Einfache W-Bewegung

Wir wollen Möglichkeiten 1 und 2 als ungrammatisch ausschließen.

- ① $\langle \text{who}^{wh}, \langle \text{whom}^{wh}, \text{saw}^{\bullet wh} \rangle \rangle \rightsquigarrow \text{who} + \langle \text{whom}, \text{saw} \rangle$
- ② $\langle \epsilon^{wh}, \langle \text{whom}^{wh}, \text{who saw}^{\bullet wh} \rangle \rangle \rightsquigarrow \epsilon + \langle \text{whom}, \text{who saw} \rangle$

Annahme

Weitere Ausdrücke, die Merkmale abgleichen, können nicht verkettet werden. (Das entspricht der Annahme, dass Englisch keine mehrfachen Spezifizierer erlaubt.)

Die Derivation arbeitet mit $\langle \text{whom}, \text{who saw} \rangle$ weiter und kann nie erfolgreich sein.



Einfache W-Bewegung

Möglichkeiten 3 und 4 hingegen sind grammatisch.

$$\textcircled{3} \langle \text{who}^{wh}, \langle \epsilon^{wh}, \text{saw whom}^{\bullet wh} \rangle \rightsquigarrow \text{who} + \langle \epsilon, \text{saw whom} \rangle$$

$$\textcircled{4} \langle \epsilon^{wh}, \langle \epsilon^{wh}, \text{who saw whom}^{\bullet wh} \rangle \rightsquigarrow \epsilon + \langle \epsilon, \text{who saw whom} \rangle$$

Mit der Annahme, dass leere Ausdrücke gelöscht werden können, bekommen wir das gewünschte Resultat `who saw whom`.



Einfache W-Bewegung

Der Unterschied zwischen 3 und 4 zeigt sich in W-Fragen mit nur einem Fragewort.

- 3 $\langle \text{whom}^{wh}, \text{did a zombie chase} \bullet^{wh} \rangle$
- 4 $\langle \epsilon^{wh}, \text{a zombie chased whom} \bullet^{wh} \rangle$

3 entspricht der üblichen direkten Frage, 4 entspricht einer Echo-Frage.

Vorhersage: Sprachen wie Bulgarisch haben keine in situ Echo-Fragen.

- ?* Ivan kupuje **šta**? (Serbo-Kroatisch (Bošković 2002))
- ?* Ivan e kupil **kakvo**? (Bulgarisch (ebd.))
- ?* Ivan kupil **čto**? (Russisch (ebd.))



W-Inseln

* Whom_i did Mary know [which zombie chased ____i]?

$$\begin{aligned} & \text{merge } (\lambda x \lambda y. y \text{ chased } x) (\text{whom})^{wh} \\ & = \langle \text{whom}^{wh}, \lambda y. y \text{ chased} :: NP \rightarrow VP \rangle \end{aligned}$$

$$\begin{aligned} & \text{merge } \text{which}^{wh} :: N \rightarrow NP \text{ zombie} :: N \\ & = \text{which zombie}^{wh} :: NP \end{aligned}$$

$$\begin{aligned} & \text{merge } \langle \text{whom}^{wh}, \lambda y. y \text{ chased} :: NP \rightarrow VP \rangle \text{ which zombie}^{wh} :: NP \\ & = \text{merge } \langle \text{whom}^{wh}, \lambda y. y \text{ chased} \rangle \langle \text{which zombie}^{wh}, \epsilon \rangle \\ & = \langle \text{which zombie}^{wh}, \text{merge } \langle \text{whom}^{wh}, \lambda y. y \text{ chased} \rangle \epsilon \rangle \\ & = \langle \text{which zombie}^{wh}, \langle \text{whom}^{wh}, \text{chased} :: VP \rangle \rangle \end{aligned}$$



W-Inseln

* Whom_i did Mary know [which zombie chased ___i]?

...

remerge $\langle \text{which zombie}^{wh}, \langle \text{whom}^{wh}, \text{chased}^{\bullet wh} :: CP \rangle \rangle$
= which zombie + $\langle \text{whom}, \text{chased} \rangle$
= $\langle \text{whom}, \text{which zombie chased} \rangle$

* Whom_i did Mary wonder [whether a zombie chased ___i]?

...

Allgemein: keine f -Bewegung aus einer f -Domäne

