

Chart-Generierung

Proseminar „Generierung“
Alexander Koller

Carolyn Ladda
14.01.11

Ablauf

- Problemstellung für Chart-Generierung
- Kays Chart-Algorithmus von '96
- Integrieren von Indizes
- Moores Algorithmus von '02

Motivation

- Ambiguität in natürlichen Sprachen
- Parsing: Anzahl der möglichen Satzstrukturen wächst exponentiell mit Satzlänge
- Generierung: Vielfalt möglicher Strukturen von Repräsentationen einer Bedeutung

Motivation

run(r), past(r), fast(r), argl (r,j), name(j, John)

Motivation

run(r), past(r), fast(r), argl (r,j), name(j, John)



<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(x)	name(x,John)
ran	vp(x,y)	run(x), argl (x,y), past(x)
fast	adv(x)	fast(x)
quickly	adv(x)	fast(x)

Motivation

run(r), past(r), fast(r), argl (r,j), name(j, John)

+

$s(x) \rightarrow np(y), vp(x,y)$

$vp(x,y) \rightarrow vp(x,y), adv(x)$

Motivation

run(r), past(r), fast(r), arg1 (r,j), name(j, John)

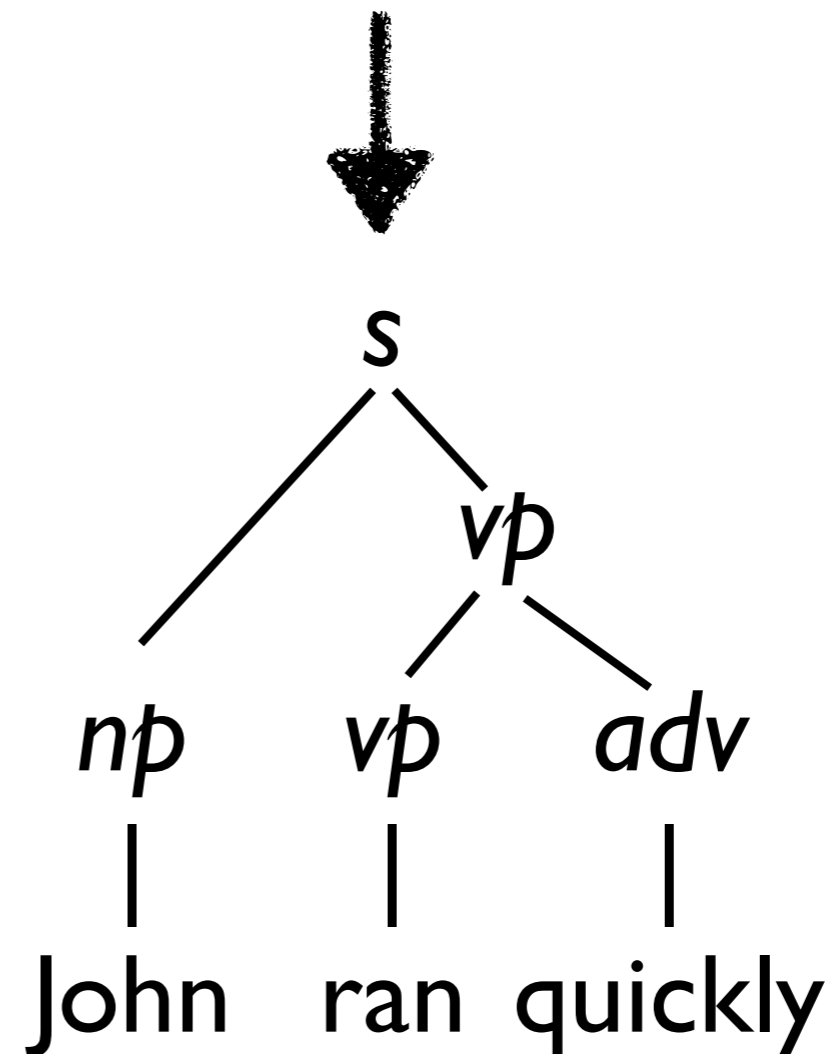
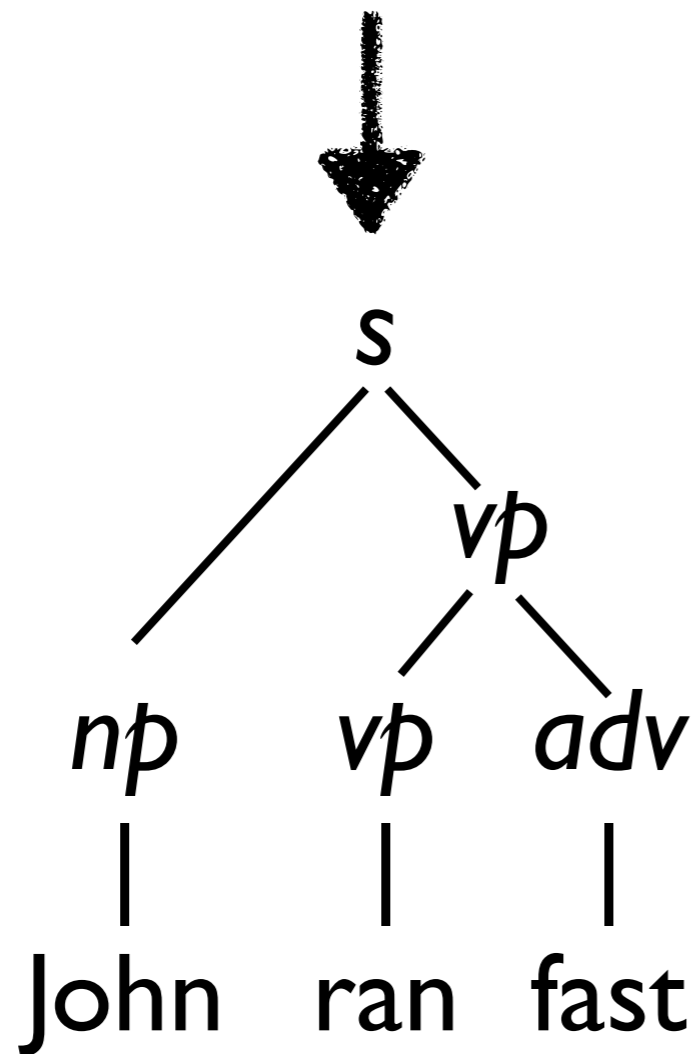


Chart-Algorithmen

- Inaktive Einträge
- Aktive Einträge
- Algorithmus
- Indizes

Chart-Parsing

- Earley-Algorithmus
- 3 Anwendungsregeln:
predict, scan, complete
- Top-Down-Strategie

Earley-Parsing

John ran fast.



<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(x)	name(x,John)
ran	vp(x,y)	run(x), arg1(x,y), past(x)
fast	adv(x)	fast(x)

Earley-Parsing

John ran fast.

+

np → John

vp → ran

adv → fast

Earley-Parsing

John ran fast.



$s \rightarrow np\ vp$

$vp \rightarrow vp\ adv$

Earley-Parsing

John ran fast.

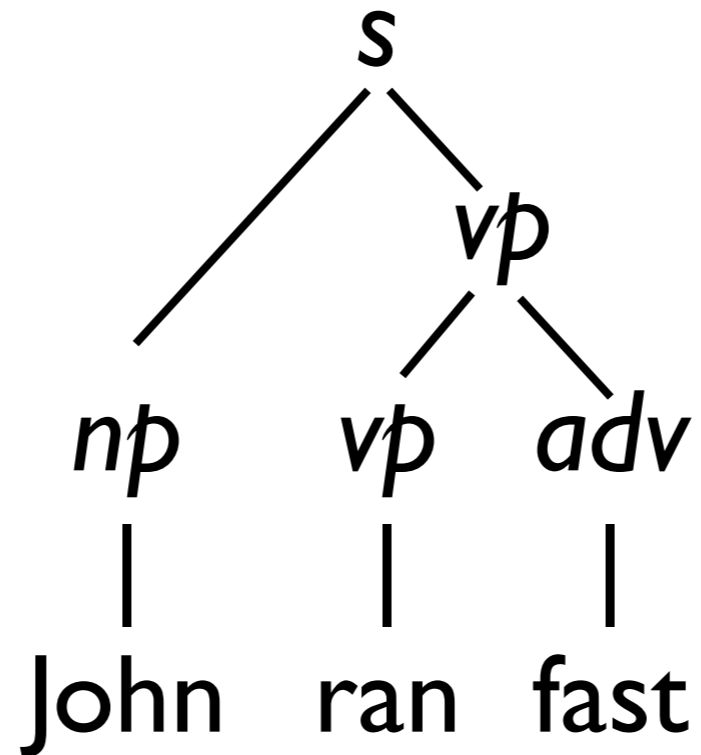


Indices:

0John₁ran₂fast₃

Earley-Parsing

John ran fast.



Earley-Parsing

*0*John₁ran₂fast₃

s → np vp

vp → vp adv

np → John

vp → ran

adv → fast

	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>

Earley-Parsing

*0*John₁ran₂fast₃

$s \rightarrow np\ vp$

$vp \rightarrow vp\ adv$

$np \rightarrow John$

$vp \rightarrow ran$

$adv \rightarrow fast$

	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>
p: $s \rightarrow \cdot np\ vp, 0$				
p: $np \rightarrow \cdot John, 0$				

Earley-Parsing

*0*John₁ran₂fast₃

$s \rightarrow np\ vp$
 $vp \rightarrow vp\ adv$
 $np \rightarrow John$
 $vp \rightarrow ran$
 $adv \rightarrow fast$

	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>
	p: $s \rightarrow \cdot np\ vp, 0$	s: $np \rightarrow John \cdot, 0$		
	p: $np \rightarrow \cdot John, 0$	c: $s \rightarrow np \cdot vp, 0$		
		p: $vp \rightarrow \cdot vp\ adv, 1$		
		p: $vp \rightarrow \cdot ran, 1$		

Earley-Parsing

*0*John₁ran₂fast₃

$s \rightarrow np\ vp$

$vp \rightarrow vp\ adv$

$np \rightarrow John$

$vp \rightarrow ran$

$adv \rightarrow fast$

	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>
	p: $s \rightarrow \cdot np\ vp, 0$	s: $np \rightarrow John \cdot, 0$	s: $vp \rightarrow ran \cdot, 1$	
	p: $np \rightarrow \cdot John, 0$	c: $s \rightarrow np \cdot vp, 0$	c: $s \rightarrow np\ vp \cdot, 0$	
		p: $vp \rightarrow \cdot vp\ adv, 1$	c: $vp \rightarrow vp \cdot adv, 1$	
		p: $vp \rightarrow \cdot ran, 1$	p: $adv \rightarrow \cdot fast, 2$	

Earley-Parsing

John ran fast

$s \rightarrow np\ vp$
 $vp \rightarrow vp\ adv$
 $np \rightarrow John$
 $vp \rightarrow ran$
 $adv \rightarrow fast$

0	1	2	3
p: $s \rightarrow \cdot np\ vp, 0$	s: $np \rightarrow John \cdot, 0$	s: $vp \rightarrow ran \cdot, 1$	s: $adv \rightarrow fast \cdot, 2$
p: $np \rightarrow \cdot John, 0$	c: $s \rightarrow np \cdot vp, 0$	c: $s \rightarrow np\ vp \cdot, 0$	c: $vp \rightarrow vp\ adv \cdot, 1$
	p: $vp \rightarrow \cdot vp\ adv, 1$	c: $vp \rightarrow vp \cdot adv, 1$	c: $s \rightarrow np\ vp \cdot, 0$
	p: $vp \rightarrow \cdot ran, 1$	p: $adv \rightarrow \cdot fast, 2$	

Earley-Parsing

*o*John₁ran₂fast₃

$s \rightarrow np\ vp$
 $vp \rightarrow vp\ adv$
 $np \rightarrow John$
 $vp \rightarrow ran$
 $adv \rightarrow fast$

	0	1	2	3
p: $s \rightarrow \cdot np\ vp$, 0	s: $np \rightarrow John \cdot$, 0	s: $vp \rightarrow ran \cdot$, 1	s: $adv \rightarrow fast \cdot$, 2	
p: $np \rightarrow \cdot John$, 0	c: $s \rightarrow np \cdot vp$, 0	c: $s \rightarrow np\ vp \cdot$, 0	c: $vp \rightarrow vp\ adv \cdot$, 1	
	p: $vp \rightarrow \cdot vp\ adv$, 1	c: $vp \rightarrow vp \cdot adv$, 1	c: $s \rightarrow np\ vp \cdot$, 0	
	p: $vp \rightarrow \cdot ran$, 1	p: $adv \rightarrow \cdot fast$, 2		

Chart-Algorithmen

- Inaktive Einträge
- Aktive Einträge
- Algorithmus
- Indizes

Chart-Algorithmen

- Inaktive Einträge
 - in sich vollständig
 - decken Teil der Eingabe vollständig ab
 - mit neuen Einträgen kombinierbar

Chart-Algorithmen

- Aktive Einträge
 - nicht vollständig
 - erwarten weitere Teile

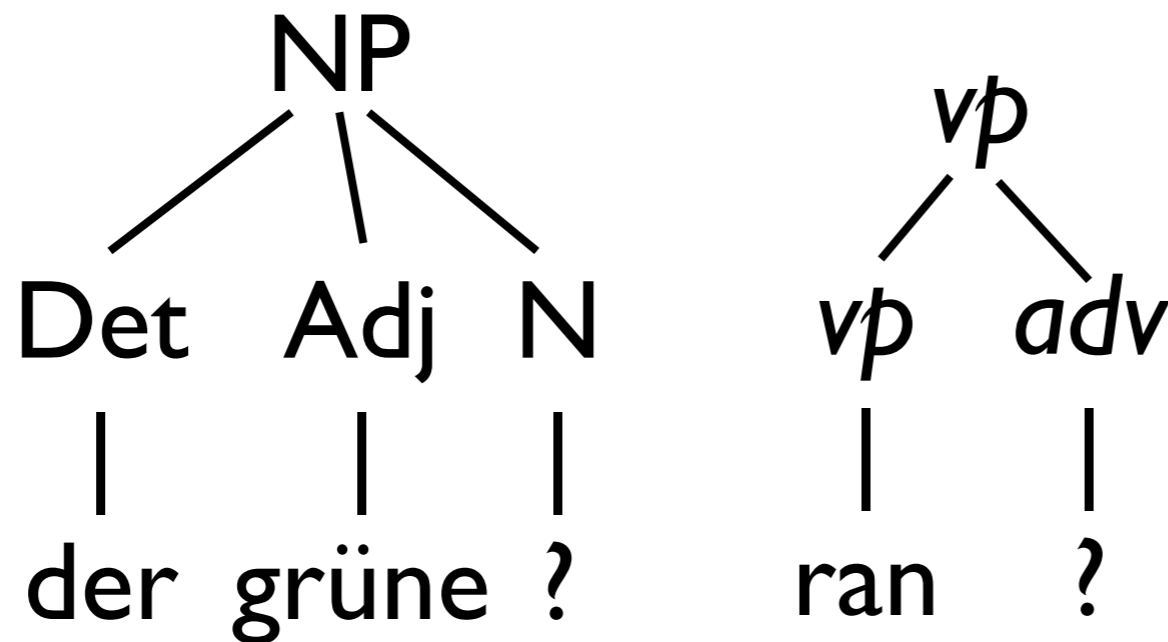


Chart-Algorithmen

- Algorithmus
- Agenda für zu bearbeitende Einträge
- Chart für abgearbeitete Einträge

Chart-Algorithmen

- Indizes
 - ergeben sich für Parsing aus der Wortreihenfolge
 - für Generierung unklar

Chart-Algorithmen

- Generierung
- keine Wortreihenfolge vorgegeben
- jede Teilanalyse kann arbiträre Teilmenge der Informationen beinhalten

Kays Algorithmus

Lexikon:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(x)	name(x,John)
ran	vp(x,y)	run(x), arg1(x,y), past(x)
Mary	np(x)	name(x,Mary)
fast	adv(x)	fast(x)
quickly	adv(x)	fast(x)
walked	vp(x,y)	walk(x), arg1(x,y), past(x)
...

Agenda:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>

Kays Algorithmus

Lexikon:

Wörter	Kategorie	Semantik
John	np(x)	name(x,John)
ran	vp(x,y)	run(x), arg1(x,y), past(x)
Mary	np(x)	name(x,Mary)
fast	adv(x)	fast(x)
quickly	adv(x)	fast(x)
walked	vp(x,y)	walk(x), arg1(x,y), past(x)
...

Agenda:

Wörter	Kategorie	Semantik

Kays Algorithmus

Lexikon:

Wörter	Kategorie	Semantik
John	np(x)	name(x,John)
ran	vp(x,y)	run(x), arg1(x,y), past(x)
Mary	np(x)	name(x,Mary)
fast	adv(x)	fast(x)
quickly	adv(x)	fast(x)
walked	vp(x,y)	walk(x), arg1(x,y), past(x)
...

Agenda:

Wörter	Kategorie	Semantik
John	np(j)	name(j,John)

Kays Algorithmus

Lexikon:

Wörter	Kategorie	Semantik
John	np(x)	name(x,John)
ran	vp(x,y)	run(x), arg1(x,y), past(x)
Mary	np(x)	name(x,Mary)
fast	adv(x)	fast(x)
quickly	adv(x)	fast(x)
walked	vp(x,y)	walk(x), arg1(x,y), past(x)
...

Agenda:

Wörter	Kategorie	Semantik
John	np(j)	name(j,John)

Kays Algorithmus

Lexikon:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(x)	name(x,John)
ran	vp(x,y)	run(x), arg1(x,y), past(x)
Mary	np(x)	name(x,Mary)
fast	adv(x)	fast(x)
quickly	adv(x)	fast(x)
walked	vp(x,y)	walk(x), arg1(x,y), past(x)
...

Agenda:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(j)	name(j,John)
ran	vp(r,j)	run(r), arg1(r,j), past (r)

Kays Algorithmus

Lexikon:

Wörter	Kategorie	Semantik
John	np(x)	name(x,John)
ran	vp(x,y)	run(x), arg1(x,y), past(x)
Mary	np(x)	name(x,Mary)
fast	adv(x)	fast(x)
quickly	adv(x)	fast(x)
walked	vp(x,y)	walk(x), arg1(x,y), past(x)
...

Agenda:

Wörter	Kategorie	Semantik
John	np(j)	name(j,John)
ran	vp(r,j)	run(r), arg1(r,j), past (r)

Kays Algorithmus

Lexikon:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(x)	name(x,John)
ran	vp(x,y)	run(x), arg1(x,y), past(x)
Mary	np(x)	name(x,Mary)
fast	adv(x)	fast(x)
quickly	adv(x)	fast(x)
walked	vp(x,y)	walk(x), arg1(x,y), past(x)
...

Agenda:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(j)	name(j,John)
ran	vp(r,j)	run(r), arg1(r,j), past (r)

Kays Algorithmus

Lexikon:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(x)	name(x,John)
ran	vp(x,y)	run(x), arg1(x,y), past(x)
Mary	np(x)	name(x,Mary)
fast	adv(x)	fast(x)
quickly	adv(x)	fast(x)
walked	vp(x,y)	walk(x), arg1(x,y), past(x)
...

Agenda:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(j)	name(j,John)
ran	vp(r,j)	run(r), arg1(r,j), past (r)
fast	adv(r)	fast(r)

Kays Algorithmus

Lexikon:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(x)	name(x,John)
ran	vp(x,y)	run(x), arg1(x,y), past(x)
Mary	np(x)	name(x,Mary)
fast	adv(x)	fast(x)
quickly	adv(x)	fast(x)
walked	vp(x,y)	walk(x), arg1(x,y), past(x)
...

Agenda:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(j)	name(j,John)
ran	vp(r,j)	run(r), arg1(r,j), past (r)
fast	adv(r)	fast(r)

Kays Algorithmus

Lexikon:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(x)	name(x,John)
ran	vp(x,y)	run(x), arg1(x,y), past(x)
Mary	np(x)	name(x,Mary)
fast	adv(x)	fast(x)
quickly	adv(x)	fast(x)
walked	vp(x,y)	walk(x), arg1(x,y), past(x)
...

Agenda:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(j)	name(j,John)
ran	vp(r,j)	run(r), arg1(r,j), past (r)
fast	adv(r)	fast(r)
quickly	adv(r)	fast(r)

Kays Algorithmus

Lexikon:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(x)	name(x,John)
ran	vp(x,y)	run(x), arg1(x,y), past(x)
Mary	np(x)	name(x,Mary)
fast	adv(x)	fast(x)
quickly	adv(x)	fast(x)
walked	vp(x,y)	walk(x), arg1(x,y), past(x)
...

Agenda:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(j)	name(j,John)
ran	vp(r,j)	run(r), arg1(r,j), past (r)
fast	adv(r)	fast(r)
quickly	adv(r)	fast(r)

Kays Algorithmus

Lexikon:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(x)	name(x,John)
ran	vp(x,y)	run(x), argl(x,y), past(x)
Mary	np(x)	name(x,Mary)
fast	adv(x)	fast(x)
quickly	adv(x)	fast(x)
walked	vp(x,y)	walk(x), argl(x,y), past(x)
...

Agenda:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(j)	name(j,John)
ran	vp(r,j)	run(r), argl(r,j), past (r)
fast	adv(r)	fast(r)
quickly	adv(r)	fast(r)

Kays Algorithmus

Agenda:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(j)	name(j,John)
ran	vp(r,j)	run(r), argl(r,j), past(r)
fast	adv(r)	fast(r)
quickly	adv(r)	fast(r)

Chart:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>

Kays Algorithmus

Agenda:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
ran	vp(r,j)	run(r), argl (r,j), past (r)
fast	adv(r)	fast(r)
quickly	adv(r)	fast(r)

Chart:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(j)	name(j,John)

Kays Algorithmus

Agenda:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
fast	adv(r)	fast(r)
quickly	adv(r)	fast(r)

Chart:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(j)	name(j,John)
ran	vp(r,j)	run(r), arg1(r,j), past(r)

Kays Algorithmus

Agenda:

Wörter	Kategorie	Semantik
fast	adv(r)	fast(r)
quickly	adv(r)	fast(r)
John ran	s(r)	run(r), argl(r,j), past(r), name(j,John)

Chart:

Wörter	Kategorie	Semantik
John	np(j)	name(j,John)
ran	vp(r,j)	run(r), argl(r,j), past(r)

Kays Algorithmus

Agenda:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
quickly	adv(r)	fast(r)
John ran	s(r)	run(r), argl(r,j), past(r), name(j,John)

Chart:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(j)	name(j,John)
ran	vp(r,j)	run(r), argl(r,j), past(r)
fast	adv(r)	fast(r)

Kays Algorithmus

Agenda:

Wörter	Kategorie	Semantik
quickly	adv(r)	fast(r)
John ran	s(r)	run(r), argl(r,j), past(r), name(j,John)
ran fast	vp(r,j)	run(r), argl(r,j), past(r), fast(r)

Chart:

Wörter	Kategorie	Semantik
John	np(j)	name(j,John)
ran	vp(r,j)	run(r), argl(r,j), past(r)
fast	adv(r)	fast(r)

Kays Algorithmus

Agenda:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John ran	s(r)	run(r), argl(r,j), past(r), name(j,John)
ran fast	vp(r,j)	run(r), argl(r,j), past(r), fast(r)

Chart:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John	np(j)	name(j,John)
ran	vp(r,j)	run(r), argl(r,j), past(r)
fast	adv(r)	fast(r)
quickly	adv(r)	fast(r)

Kays Algorithmus

Agenda:

Wörter	Kategorie	Semantik
John ran	s(r)	run(r), argl(r,j), past(r), name(j,John)
ran fast	vp(r,j)	run(r), argl(r,j), past(r), fast(r)
ran quickly	vp(r,j)	run(r), argl(r,j), past(r), fast(r)

Chart:

Wörter	Kategorie	Semantik
John	np(j)	name(j,John)
ran	vp(r,j)	run(r), argl(r,j), past(r)
fast	adv(r)	fast(r)
quickly	adv(r)	fast(r)

Kays Algorithmus

Agenda:

Wörter	Kategorie	Semantik
ran fast	vp(r,j)	run(r), argl (r,j), past (r), fast(r)
ran quickly	vp(r,j)	run(r), argl (r,j), past (r), fast(r)

Chart:

Wörter	Kategorie	Semantik
John	np(j)	name(j,John)
...
John ran	s(r)	run(r), argl (r,j), past (r), name(j,John)

Kays Algorithmus

Agenda:

Wörter	Kategorie	Semantik
ran quickly	vp(r,j)	run(r), argl(r,j), past (r), fast(r)

Chart:

Wörter	Kategorie	Semantik
John	np(j)	name(j,John)
...
John ran	s(r)	run(r), argl(r,j), past (r), name(j,John)
ran fast	vp(r,j)	run(r), argl(r,j), past (r), fast(r)

Kays Algorithmus

Agenda:

Wörter	Kategorie	Semantik
ran quickly	vp(r,j)	run(r), argl(r,j), past(r), fast(r)
John ran fast	s(r)	run(r), argl(r,j), past(r), fast(r), name(j,John)

Chart:

Wörter	Kategorie	Semantik
John	np(j)	name(j,John)
...
John ran	s(r)	run(r), argl(r,j), past(r), name(j,John)
ran fast	vp(r,j)	run(r), argl(r,j), past(r), fast(r)

Kays Algorithmus

Agenda:

Wörter	Kategorie	Semantik
John ran fast	s(r)	run(r), argl(r,j), past (r), fast(r), name (j,John)

Chart:

Wörter	Kategorie	Semantik
John	np(j)	name(j,John)
...
John ran	s(r)	run(r), argl(r,j), past (r), name(j,John)
ran fast	vp(r,j)	run(r), argl(r,j), past (r), fast(r)
ran quickly	vp(r,j)	run(r), argl(r,j), past (r), fast(r)

Kays Algorithmus

Agenda:

Wörter	Kategorie	Semantik
John ran fast	s(r)	run(r), argl(r,j), past (r), fast(r), name (j,John)
John ran quickly	s(r)	run(r), argl(r,j), past (r), fast(r), name (j,John)

Chart:

Wörter	Kategorie	Semantik
John	np(j)	name(j,John)
...
John ran	s(r)	run(r), argl(r,j), past (r), name(j,John)
ran fast	vp(r,j)	run(r), argl(r,j), past (r), fast(r)
ran quickly	vp(r,j)	run(r), argl(r,j), past (r), fast(r)

Kays Algorithmus

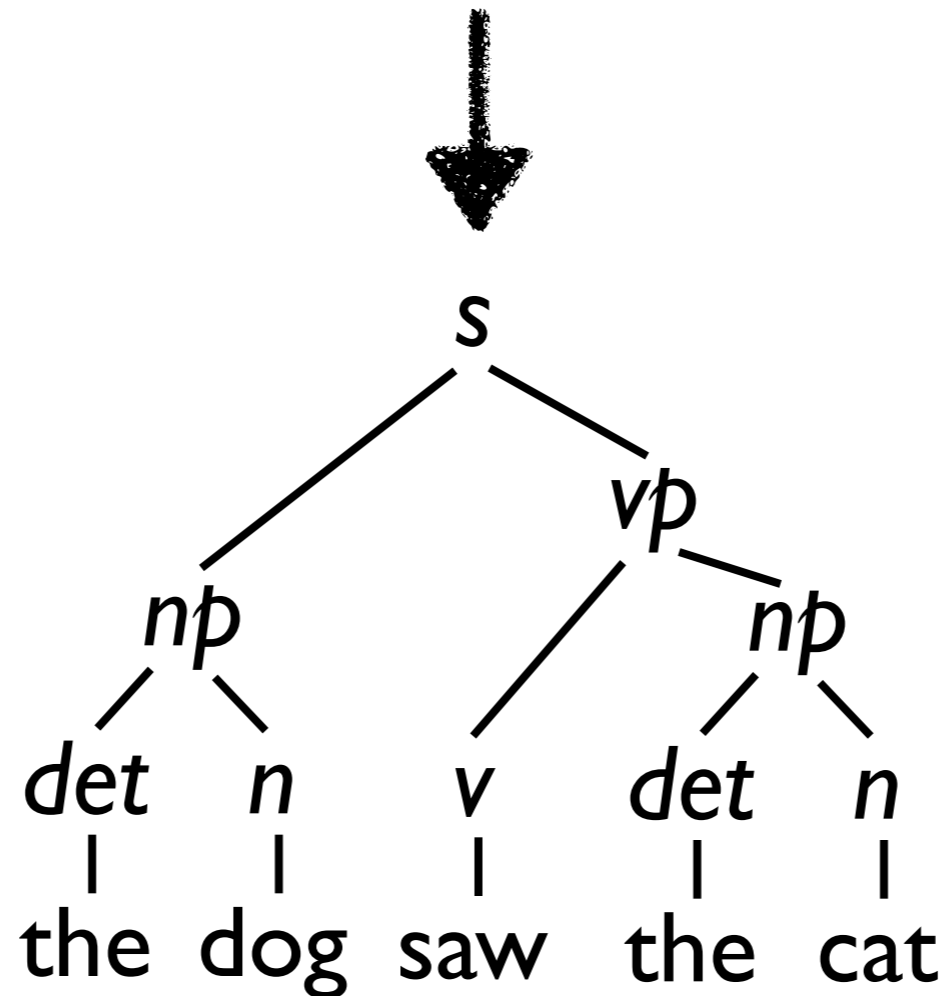
<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
John ran fast	s(r)	run(r), argl(r,j), past (r), fast(r), name (j,John)
John ran quickly	s(r)	run(r), argl(r,j), past (r), fast(r), name (j,John)

Eigenschaften des Algorithmus

- Indizes werden nicht benutzt
- keine aktiven Einträge wg. binärer Regeln
- Für die Kombination mit neuen Charteinträgen müssen alle bereits vorhandenen berücksichtigt werden

Eigenschaften des Algorithmus

dog(d), def(d), see(s), past(s), cat(c), def(c), arg1 (s, d), arg2(s, c)

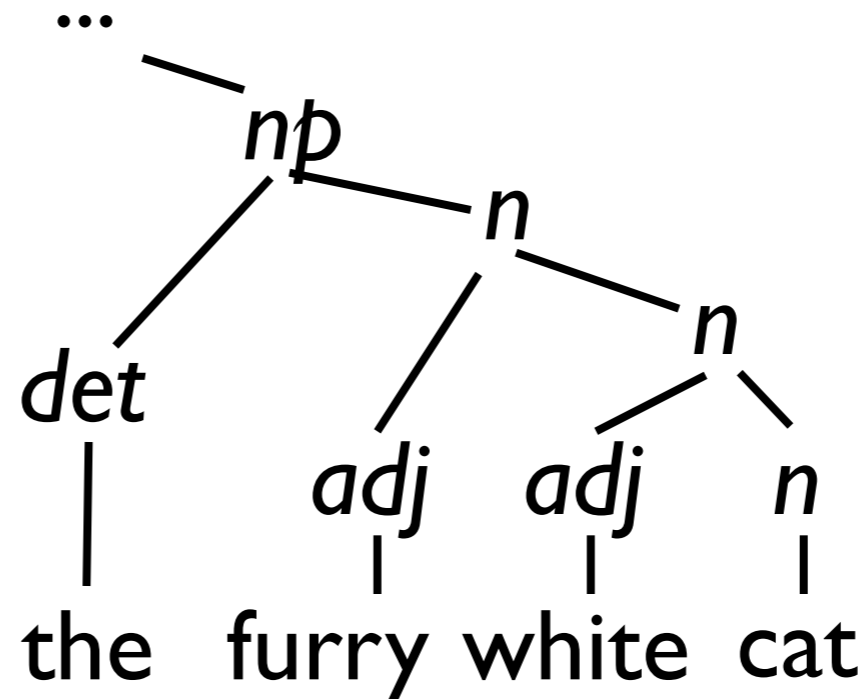


Eigenschaften des Algorithmus

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
dog	n(d)	dog(d)
the	det(d)	def(d)
the	det(c)	def(c)
saw	v(s,d,c)	see(s), past(s), arg1 (s,d), arg2(s,c)
cat	n(c)	cat(c)

Eigenschaften des Algorithmus

... cat(c), def(c), furry(c), white(c)...



Optimierung durch Indizes

- Eintrag: $B(b \dots)$, b ist Index
- Für jeden Eintrag $B(b \dots)$
und jede Regel $A(\dots) \rightarrow B(b \dots), C(c \dots)$:
- Eintrag: $A(\dots)/C(c \dots)$ mit c als Index

Modifizierter Algorithmus

dog(d), def(d), see(s), past(s), cat(c), def(c), arg1 (s, d), arg2(s, c)



<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
cat	n(x)	x: cat(x)
saw	v(x,y,z)	x: see(x), past(x), arg1 (x,y), arg2(x,z)
dog	n(x)	x: dog(x)
the	det(x)	x: def(x)
...

Modifizierter Algorithmus

dog(d), def(d), see(s), past(s), cat(c), def(c), arg1 (s, d), arg2(s, c)

+

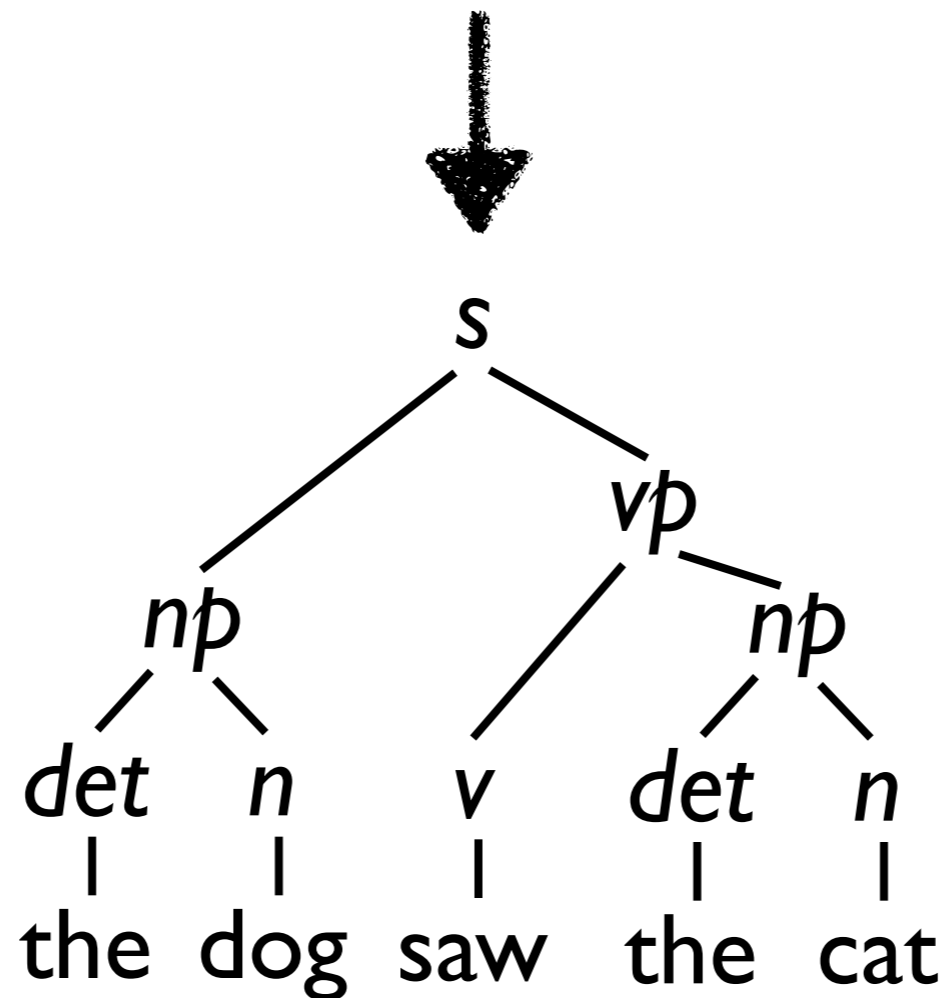
$s(x) \rightarrow np(y), vp(x,y)$

$vp(x,y) \rightarrow v(x,y,z), np(z)$

$np(x) \rightarrow det(x), n(x)$

Modifizierter Algorithmus

dog(d), def(d), see(s), past(s), cat(c), def(c), arg1 (s, d), arg2(s, c)



Modifizierter Algorithmus

dog(d), def(d), see(s), past(s), cat(c), def(c), arg1 (s, d), arg2(s, c)



3 Indizes:

d, s, c

Modifizierter Algorithmus

Lexikon:

Agenda:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
cat	n(x)	x: cat(x)
saw	v(x,y,z)	x: see(x), past(x), arg1(x,y), arg2(x,z)
dog	n(x)	x: dog(x)
the	det(x)	x: def(x)
...

	<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
d			
s			
c			

Modifizierter Algorithmus

Lexikon:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
cat	$n(x)$	$x: \text{cat}(x)$
saw	$v(x,y,z)$	$x: \text{see}(x), \text{past}(x),$ $\text{arg1}(x,y), \text{arg2}(x,z)$
dog	$n(x)$	$x: \text{dog}(x)$
the	$\text{det}(x)$	$x: \text{def}(x)$
...

Agenda:

	<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
d			
s			
c	cat	$n(c)$	$c: \text{cat}(c)$

Modifizierter Algorithmus

Lexikon:

Agenda:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
cat	n(x)	x: cat(x)
saw	v(x,y,z)	x: see(x), past(x), arg1(x,y), arg2(x,z)
dog	n(x)	x: dog(x)
the	det(x)	x: def(x)
...

	<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
d			
s	saw	v(s,d,c)	s: see(s), past(s), arg1(s,d), arg2(s,c)
c	cat	n(c)	c: cat(c)

Modifizierter Algorithmus

Lexikon:

Wörter	Kategorie	Semantik
cat	n(x)	x: cat(x)
saw	v(x,y,z)	x: see(x), past(x), arg1(x,y), arg2(x,z)
dog	n(x)	x: dog(x)
the	det(x)	x: def(x)
...

Agenda:

	Wörter	Kategorie	Semantik
d			
s	saw	v(s,d,c)	s: see(s), past(s), arg1(s,d), arg2(s,c)
c	saw	vp(s,d)/np (c)	c: see(s), past(s), arg1(s,d), arg2(s,c)
	cat	n(c)	c: cat(c)

Modifizierter Algorithmus

Lexikon:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
cat	n(x)	x: cat(x)
saw	v(x,y,z)	x: see(x), past(x), arg1(x,y), arg2(x,z)
dog	n(x)	x: dog(x)
the	det(x)	x: def(x)
...

Agenda:

	<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
d	dog	n(d)	d: dog(d)
s	saw	v(s,d,c)	s: see(s), past(s), arg1(s,d), arg2(s,c)
c	saw	vp(s,d)/np (c)	c: see(s), past(s), arg1(s,d), arg2(s,c)
	cat	n(c)	c: cat(c)

Modifizierter Algorithmus

Lexikon:

Wörter	Kategorie	Semantik
cat	n(x)	x: cat(x)
saw	v(x,y,z)	x: see(x), past(x), arg1(x,y), arg2(x,z)
dog	n(x)	x: dog(x)
the	det(x)	x: def(x)
...

Agenda:

	Wörter	Kategorie	Semantik
d	dog	n(d)	d: dog(d)
	the	det(d)	d: def(d)
s	saw	v(s,d,c)	s: see(s), past(s), arg1(s,d), arg2(s,c)
c	saw	vp(s,d)/np (c)	c: see(s), past(s), arg1(s,d), arg2(s,c)
	cat	n(c)	c: cat(c)

Modifizierter Algorithmus

Lexikon:

Wörter	Kategorie	Semantik
cat	n(x)	x: cat(x)
saw	v(x,y,z)	x: see(x), past(x), arg1(x,y), arg2(x,z)
dog	n(x)	x: dog(x)
the	det(x)	x: def(x)
...

Agenda:

	Wörter	Kategorie	Semantik
d	dog	n(d)	d: dog(d)
	the	det(d)	d: def(d)
	the	np(d)/n(d)	d: def(d)
s	saw	v(s,d,c)	s: see(s), past(s), arg1(s,d), arg2(s,c)
c	saw	vp(s,d)/np (c)	c: see(s), past(s), arg1(s,d), arg2(s,c)
	cat	n(c)	c: cat(c)

Modifizierter Algorithmus

Lexikon:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
cat	n(x)	x: cat(x)
saw	v(x,y,z)	x: see(x), past(x), arg1(x,y), arg2(x,z)
dog	n(x)	x: dog(x)
the	det(x)	x: def(x)
...

Agenda:

	<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
	dog	n(d)	d: dog(d)
d	the	det(d)	d: def(d)
	the	np(d)/n(d)	d: def(d)
s	saw
	saw
c	cat	n(c)	c: cat(c)
	the	det(c)	c: def(c)

Modifizierter Algorithmus

Lexikon:

<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
cat	n(x)	x: cat(x)
saw	v(x,y,z)	x: see(x), past(x), arg1(x,y), arg2(x,z)
dog	n(x)	x: dog(x)
the	det(x)	x: def(x)
...

Agenda:

	<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
d	dog	n(d)	d: dog(d)
	the	det(d)	d: def(d)
	the	np(d)/n(d)	d: def(d)
s	saw
c	saw
	cat	n(c)	c: cat(c)
	the	det(c)	c: def(c)
	the	np(c)/n(c)	c: def(c)

Modifizierter Algorithmus

	<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
d	dog	n(d)	d: dog(d)
	the	det(d)	d: def(d)
	the	np(d)/n(d)	d: def(d)
s	saw	v(s,d,c)	s: see(s), past(s), arg1(s,d), arg2(s,c)
c	saw	vp(s,d)/np (c)	c: see(s), past(s), arg1(s,d), arg2(s,c)
	cat	n(c)	c: cat(c)
	the	det(c)	c: def(c)
	the	np(c)/n(c)	c: def(c)

Modifizierter Algorithmus

	<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
d	dog	n(d)	d: dog(d)
	the	det(d)	d: def(d)
	the	np(d)/n(d)	d: def(d)
s	saw	v(s,d,c)	s: see(s), past(s), arg1(s,d), arg2(s,c)
c	saw	vp(s,d)/np (c)	c: see(s), past(s), arg1(s,d), arg2(s,c)
	cat	n(c)	c: cat(c)
	the	det(c)	c: def(c)
	the	np(c)/n(c)	c: def(c)

Modifizierter Algorithmus

	<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
d	dog	n(d)	d: dog(d)
	the	det(d)	d: def(d)
	the	np(d)/n(d)	d: def(d)
s	saw	v(s,d,c)	s: see(s), past(s), arg1(s,d), arg2(s,c)
c	saw	vp(s,d)/np (c)	c: see(s), past(s), arg1(s,d), arg2(s,c)
	cat	n(c)	c: cat(c)
	the	det(c)	c: def(c)
	the	np(c)/n(c)	c: def(c)

Modifizierter Algorithmus

	<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
d	dog	n(d)	d: dog(d)
	the	det(d)	d: def(d)
	the	np(d)/n(d)	d: def(d)
s	saw	v(s,d,c)	s: see(s), past(s), arg1(s,d), arg2(s,c)
c	saw	vp(s,d)/np (c)	c: see(s), past(s), arg1(s,d), arg2(s,c)
	cat	n(c)	c: cat(c)
	the	det(c)	c: def(c)
	the	np(c)/n(c)	c: def(c)

Modifizierter Algorithmus

	<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
d	the dog	np(d)	d: dog(d), def (d)
s	saw the cat	vp(s,d)	s: see(s), past(s), arg1 (s,d), arg2(s,c), cat(c), def(c)
	the dog	s(s)/vp (s,d)	s: dog(d), def (d)
c	the cat	np(c)	c: cat(c), def(c)

Modifizierter Algorithmus

	<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
d	the dog	np(d)	d: dog(d), def (d)
s	saw the cat	vp(s,d)	s: see(s), past(s), arg1(s,d), arg2(s,c), cat(c), def(c)
	the dog	s(s)/vp (s,d)	s: dog(d), def (d)
c	the cat	np(c)	c: cat(c), def(c)

Modifizierter Algorithmus

	<i>Wörter</i>	<i>Kategorie</i>	<i>Semantik</i>
s	the dog saw the cat	s(s)	s: see(s), past(s), arg1(s,d), arg2(s,c), cat(c), def(c), dog (d), def (d)

Moore's Algorithmus

- Funktioniert für flache Darstellungen der logischen Form ähnlich wie Kays
- Kann auch mit baumstrukturartig aufgebauten LFs umgehen

Moore's Algorithmus

Mary sees Jane.

→ $see(s), agt(s, Mary), pat(s, Jane)$

→ $see(Mary, Jane)$

Moore's Algorithmus

John runs fast.

→ $\text{run}(r), \text{fast}(r), \text{agt}(r, \text{John})$

→ $\text{fast}(\text{run}(\text{John}))$

Grammatik

s:[stype=decl]/VP_sem

→ np:[prsn=P, num=N]/NP_sem

vp:[vtype=tensed, prsn=P, num=N,
sub=NP_sem]/VP_sem

Moore's Algorithmus

vp:[vtype=tensed, prsn=3, num=sg, sub=S]/
run(S) → runs

np:[prsn=3,num=sg]/john → John

Moore's Algorithm

- Wenn die Struktur der Syntax der Struktur der logischen Form folgt, ist die Laufzeit polynomiell
- Andere äquivalente Strukturen werden dabei ignoriert

Zusammenfassung

- Chart-Algorithmen im Allgemeinen
- Anwendung für Generierungsprobleme
- Erweiterungen zur Effizienzsteigerung