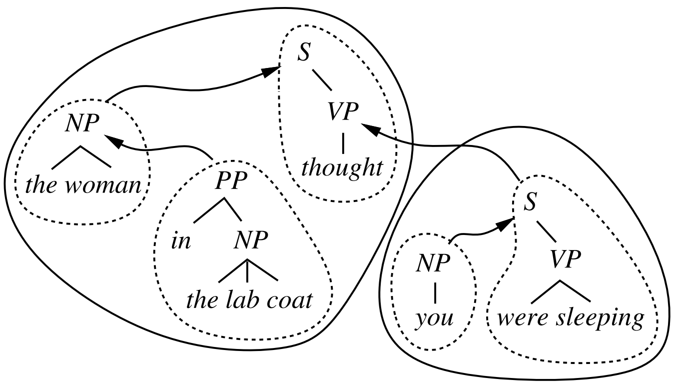
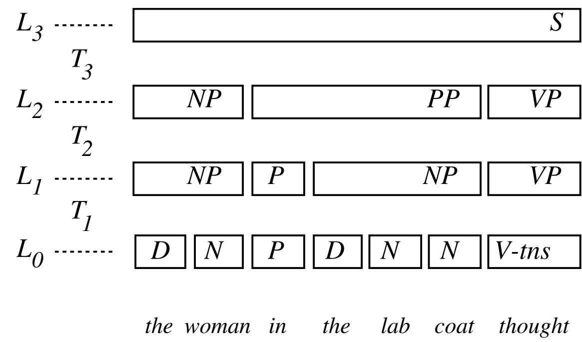
Cascaded Finite-State Parsing

Steven Abney
University of Tübingen

Cascaded Finite-State Parsing

- Grammar divided in strata/levels (chunks & clauses)
- Pipeline of finite-state recognizers/transducers



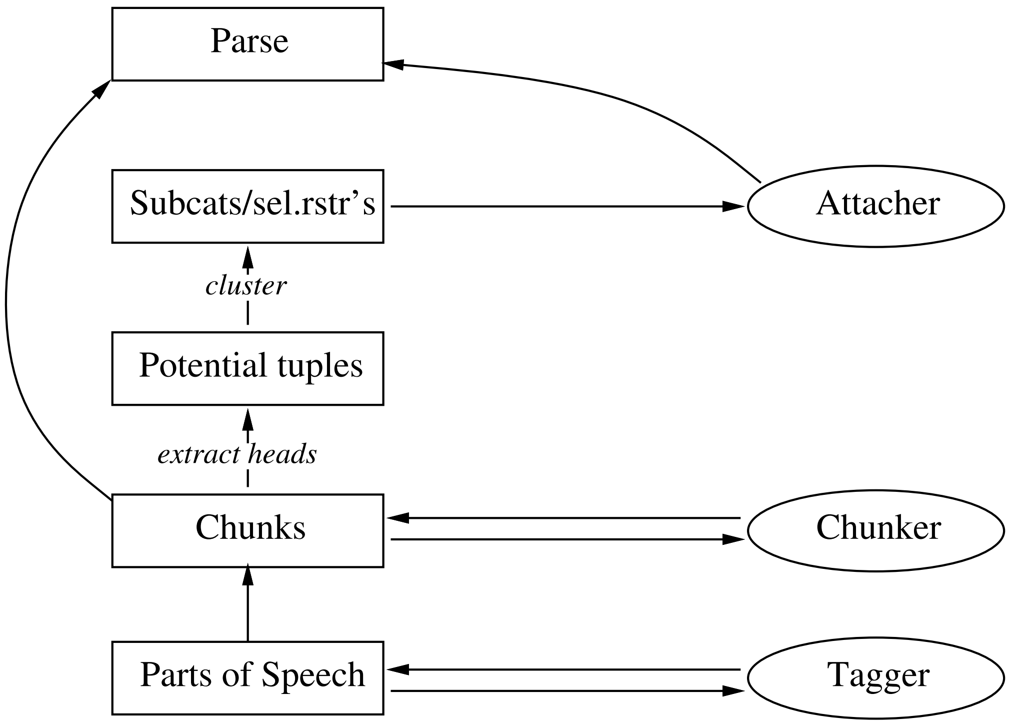


Themes

- Robustness
  - Local decisions, not global optimization
    - No closed-world assumption with consequent fragility
  - Easy-first parsing
    - High-precision (low-entropy) decisions
    - Acceptable error rates even without search
- Speed
  - Deterministic, not exhaustive search
- Reorganizing the decision space
  - Islands of certainty
  - Containment of ambiguity
  - Divorcing control structure from parse structure

Context: The B7 Project

- Bootstrapping



Why Chunking Helps with Valencies

- Fewer input units
- Smaller domain

  $[_S [_{NX}$ *the woman*$] [_{PX}$ *in the lab coat*$] [_{VX}$ *thought*$]]$
  $[_S [_{NX}$ *you*$] [_{VX}$ *were sleeping*$]]$

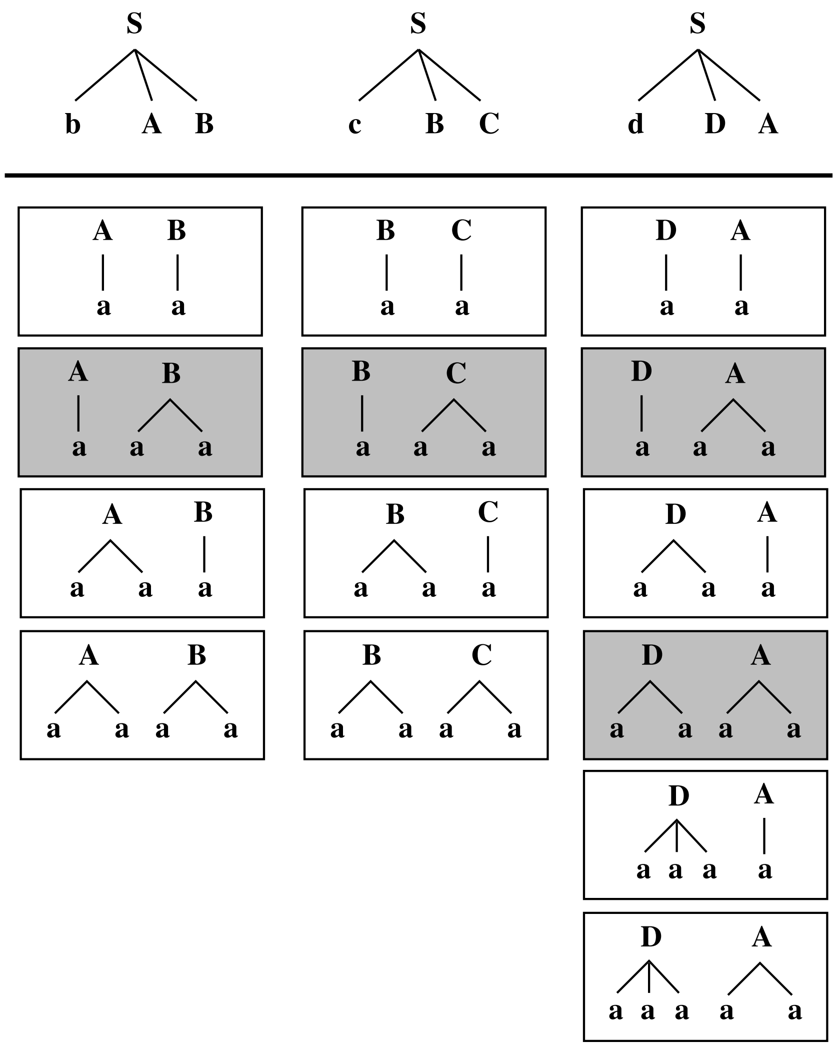
- Statistics filter out noise
  - Adjuncts—*don't appear specially with V*
  - Noun args—*don't appear reliably with V*

Trading Off Speed and Accuracy

- No search → faster, less accurate
- Sometimes faster & more accurate

$$
\begin{aligned}
\text{S} &\rightarrow \text{b A B} &&: p_{S_1}\\
&\mid\ \text{c B C} &&: p_{S_2}\\
&\mid\ \text{d D A} &&: q_S = 1-(p_{S_1}+p_{S_2})\\
\text{A} &\rightarrow \text{a} &&: p_A\\
&\mid\ \text{a a} &&: q_A = 1-p_A\\
\text{B} &\rightarrow \text{a} &&: p_B\\
&\mid\ \text{a a} &&: q_B = 1-p_B\\
\text{C} &\rightarrow \text{a} &&: p_C\\
&\mid\ \text{a a} &&: q_C = 1-p_C\\
\text{D} &\rightarrow \text{a} &&: p_{D_1}\\
&\mid\ \text{a a} &&: p_{D_2}\\
&\mid\ \text{a a a} &&: q_D = 1-(p_{D_1}+p_{D_2})
\end{aligned}
$$

Tree Set



Maximum Likelihood Estimate

- Corpus C = one of each longest-match tree
- Likelihood

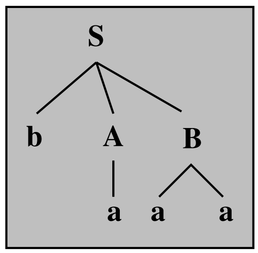
$$L(C;p) = [p_{S_1} p_A p_B \cdot p_{S_1} q_A p_B \cdot p_{S_1} q_A q_B] \cdot$$
$$[p_{S_2} p_B p_C \cdot p_{S_2} q_B p_C \cdot p_{S_2} q_B q_C] \cdot$$
$$[q_S p_{D_1} p_A \cdot q_S p_{D_2} p_A \cdot q_S q_D p_A \cdot q_S q_D q_A]$$

- Maximize $\dfrac{\partial}{\partial p_i} \ln L(C;p)$
  - Result: probabilities are relative counts
  - E.g.: $p_{S_1} = \dfrac{\#(S \to baB)}{\#(S \to)} = \dfrac{3}{10}$
  - E.g.: $p_A = \dfrac{\#(A \to a)}{\#(A \to)} = \dfrac{4}{7}$

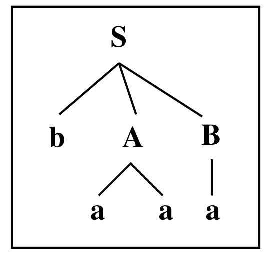
Going Astray

S →	b A B : 3/10	c B C : 3/10	d D A : 4/10
A →	a : 4/7	a a : 3/7	
B →	a : 3/6	a a : 3/6	
C →	a : 2/3	a a : 1/3	
D →	a : 1/4	a a : 1/4	a a a : 1/2

t_1 =



t_2 =



$$p(t_1) = \left(\frac{3}{10}\right)\left(\frac{4}{7}\right)\left(\frac{1}{2}\right)$$

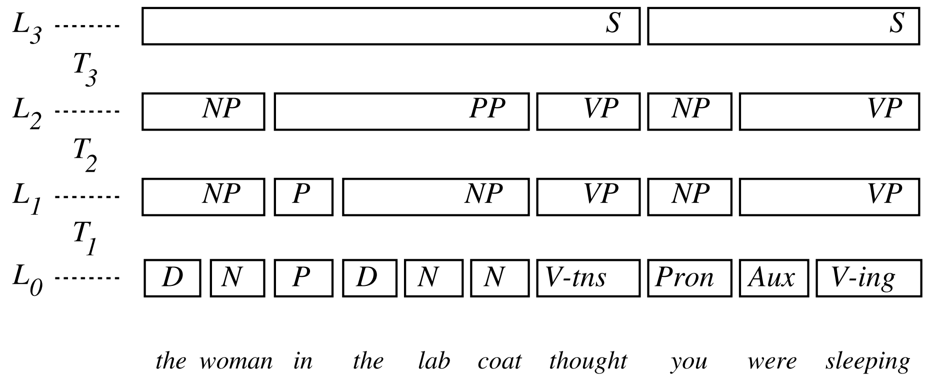
$$p(t_1|baaa) = \frac{4}{7}$$

$$p(t_2) = \left(\frac{3}{10}\right)\left(\frac{3}{7}\right)\left(\frac{1}{2}\right)$$

$$p(t_2|baaa) = \frac{3}{7}$$

Finite-State Cascade

- Finite-State Cascade



- Regular-Expression Grammar

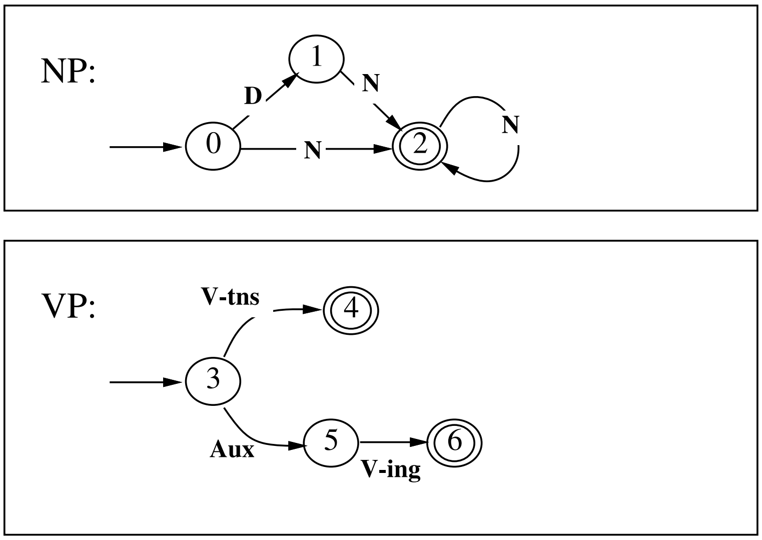
$$L_1 : \left\{ \begin{array}{l} \text{NP} \rightarrow \text{D? N}* \text{ N} \\ \text{VP} \rightarrow \text{V-tns} \mid \text{Aux V-ing} \end{array} \right\}$$

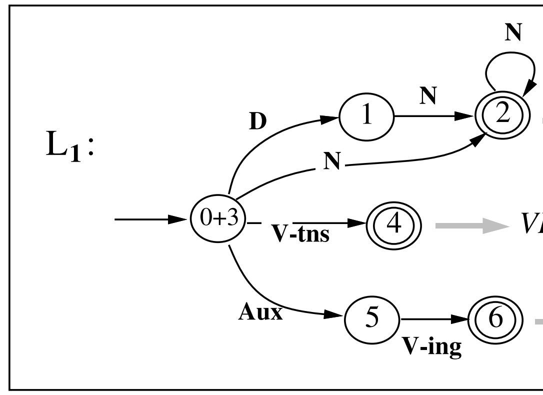
$$L_2 : \{\text{PP} \rightarrow \text{P NP}\}$$

$$L_3 : \{\text{S PP}* \text{ NP PP}* \text{ VP PP}*\}$$

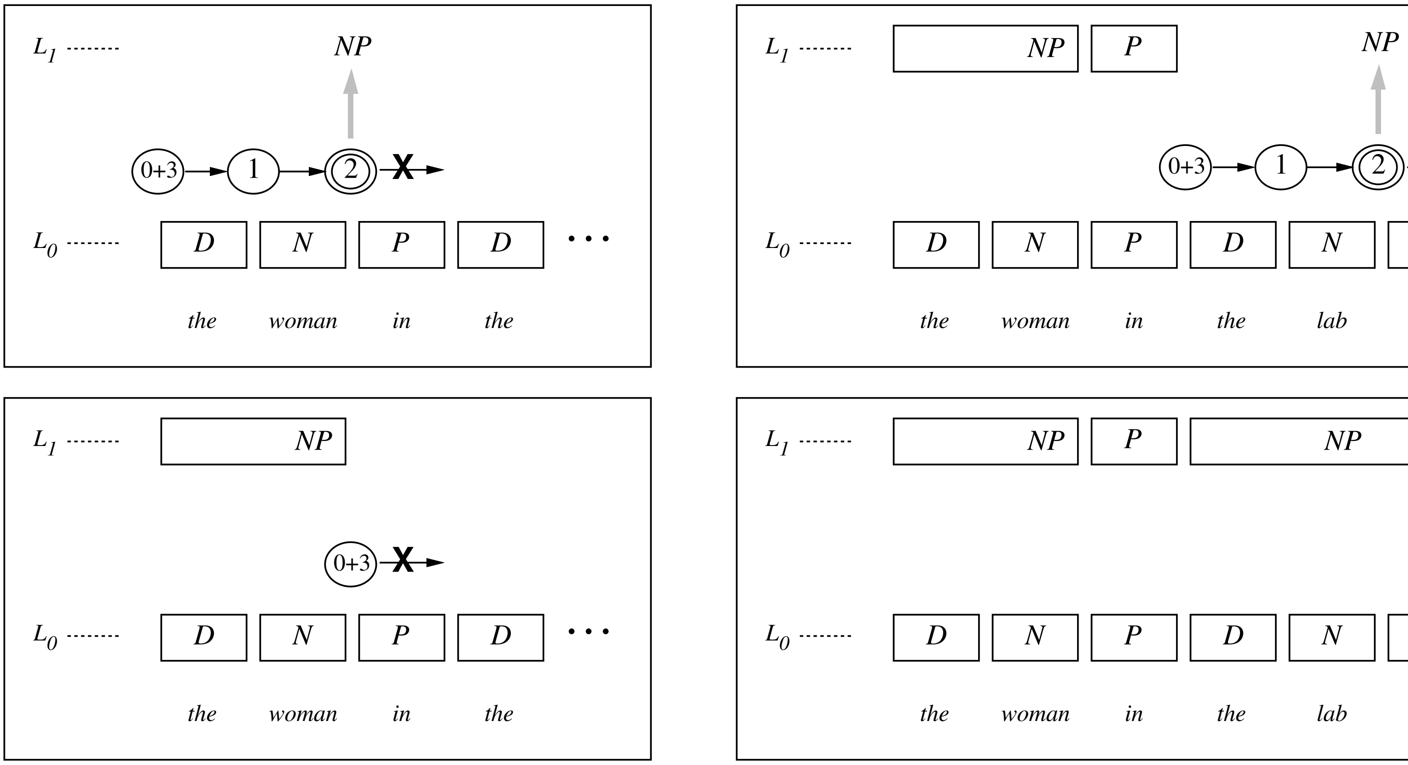
Recognition

- At level L_k : take union, determinize





Longest Match



Grammar

- Tagfixes *(word, tag) → tag*
- Measure Phrase *Dates and times, numerals, predeterminers, dollar amounts*
- Chunks *Noun, adjective, adverb, verb, infinitive chunks*
- N-Mess *Unassembled noun chunk pieces*
- NP: Possessors
- NG: Coordinated NP's
- PP
- RC *Center embedding*
- C0 *Subj+Pred: bleed Clause*
- Clause

```
date    →  month cd (cma cd cma?)?;
cdqlx   →  as much as | more than | about | only | well? over;
cdx     →  (cdql | cdqlx) cd | (cdql | cdqlx)? cd+ cd;
doll    →  cdqlx? dol cdx;
ci-st   →  nnp+ cma place cma?;

mx      →  cdx h=(units | tunits)
        |  (cdql | cdqlx)? dt-a h=(unit | tunit)
        ;

timex   →  dtp h=tunit;

tadvx   →   (  cdx h=(nns | units | tunits)
            |  (cdql | cdqlx)? dt-a h=(nn | unit | tunit)
            )
           ago
        ;

person  →  tt? i∗ fn (i | nnp)∗
        |  tt (i | nnp)∗
        ;

name    →  (nnp | nnps | i)+;
```

Noun Chunk

```
PROPER    =  place | person | name | ci-st | doll;
COMMON    =  nn | nns | month | unit | units | tunit | tunits;
N         =  PROPER | COMMON | date;
ADVHD     =  rb | cdql | then | well;
ADV       =  ADVHD | rbr | more | rbs | ql;
VADVP     =  ADV∗ (ADVHD | only);
JX        =  ADV? (jj | jjr | jjs);
JXC       =  JX (cma JX)∗ (cc | cma) JX;
ADJ       =  JX | JXC | mx;
PTC       =  (ADV | rbr | more | rbs)? (vbn | vbg);
DET       =  dt | dtp | prp$ | (cdql | cdqlx)? (dt-a | dt-q | dtp-q);
NUM       =  cd | cdx;
MX        =  mx | units | tunits;

nx        →  DET? NUM? (ADJ | PTC)∗ (ADJ | N)∗ h=COMMON cd?
          |  DET NUM? (ADJ | PTC)∗ h=PROPER
          |  DET h=(jjr | jjs)
          |  cdql? h=dtp-q
          |  h=(  prp | cd | dtp | cd | dtp | qq | ex
                  | name | person | date | doll | ci-st | rbr | rbs
             )
          ;
```

Verb Chunk

```
VB-TNS    =  vb | vbp | vbz | vbd;
DO-TNS    =  do | doz | dod;
HV-TNS    =  hv | hvz | hvd;
BE-TNS    =  be | bem | bez | bedz | bed | ber | bedr;
MODAL     =  md | doz | do | dod;
VP-PASS   =  VADVP? (vbn | vbd | hvn | ben);
VP-PROG   =  VADVP? (vbg | hvg | beg (VP-PASS | ax)?);
VP-PERF   =  VADVP? (vbn | hvn | ben (VP-PROG | VP-PASS | ax)?);
VP-INF    =  VADVP? (vb | do | hv VP-PERF? | be (VP-PROG | VP-PASS | ax)

vx        →  VADVP?  (  md VP-INF?
                     |  DO-TNS VP-INF?
                     |  VB-TNS
                     |  HV-TNS VP-PERF?
                     |  BE-TNS (VP-PROG | VP-PASS | ax)?
                     )
             ;

inf       →  VADVP? to VADVP? VP-INF;
```

Minor Chunks

```
perx  →   (  (cd | cdx | dt-a) (COMMON | doll)
          |  MX
          )
         (dt-a | per) COMMON;

rx    →  ADV+ ADV
      |  by then
      |  MX ago
      ;

ax    →  (ADV | rx)∗ jj
      |  MX JX
      |  VADVP? (vbn | vbg)
      ;
```

Larger Noun Phrases

```
DET      =  dt | dtp | prp$ cdql? (dt-q | dtp-q);
NOM      =  nx | mx | cdx | place | person | name | ci-st | doll | date;
NP       =  (NOM | np | nmess) (of (NOM | np | nmess))*;
CONNECT  =  cma | cc | cma cc;
```

- nmess:

```
nmess  →  DET ax* h=NOM?;
```

- np:

```
np  →  only? (NOM | nmess) pos h=NOM?;
```

- ng:

```
ng  →  h=NOM (of NOM)+
     |  h=NOM (of NOM)* cc NP
     |  h=NOM (of NOM)* (CONNECT NP)+ cma? cc NP
     ;
```

Prepositional and Larger Verbal Phrases

PREP	=	in | by | to | of | than | as | cdqlx | according to | because of;
NOM	=	ng | nmess | np | nx | mx | cdx | place | person | name | ci-st | doll
PRED-TAIL	=	NOM? (pp | pp-comp | rx)∗;

pp	→	prep h=NOM;
pp-comp	→	p-comp h=NOM;
infp	→	h=inf PRED-TAIL;
vbnp	→	h=(vbn | vnx) (pp | pp-comp | rx)∗;
vbgp	→	h=(vbg | vgx) PRED-TAIL;
cc-vp	→	cc h=vp PRED-TAIL;

Clauses

```
NOM          =  ng | nmess | np | nx | mx | cdx | place | person | name | ci-st | doll
WH           =  wdt | wp;
SUBJ-TAIL    =  pp∗  (  cma? (src | orc | infp | vbnp | vbgp) cma?
                     |  cma NOM cma
                     )  ?
               ;
PRED-TAIL    =  NOM? (pp | pp-comp | rx)∗;
```

- rc:

```
src   →  WH h=vp NOM? (pp | rx)∗;
orc   →  WH (ng | np | nx) pp∗ h=vp (pp | rx)∗;
```

- c0:

```
c0     →  NOM SUBJ-TAIL h=vp;
subc0  →  (pp-comp | (comp | because) NOM) SUBJ-TAIL h=vp;
```

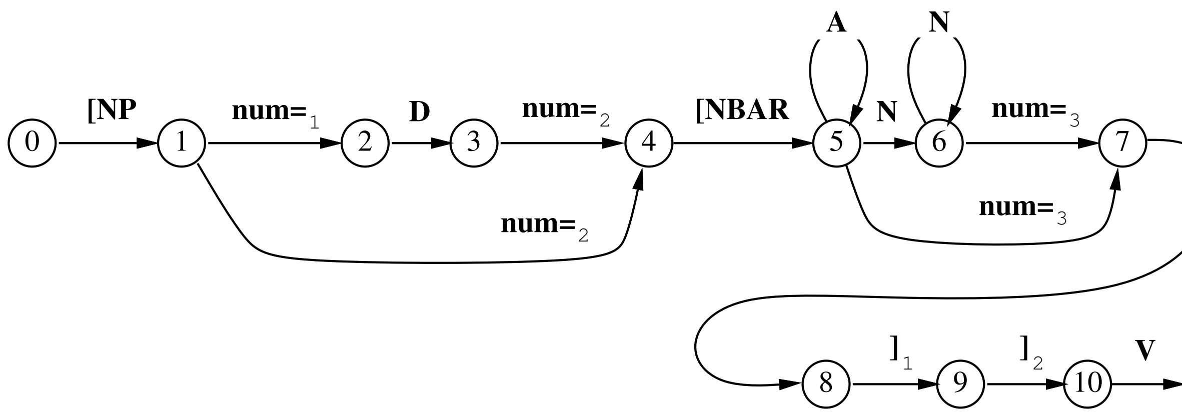
- s:

```
c     →  h=c0 PRED-TAIL;
subc  →  h=subc0 PRED-TAIL;
```

Bells & Whistles

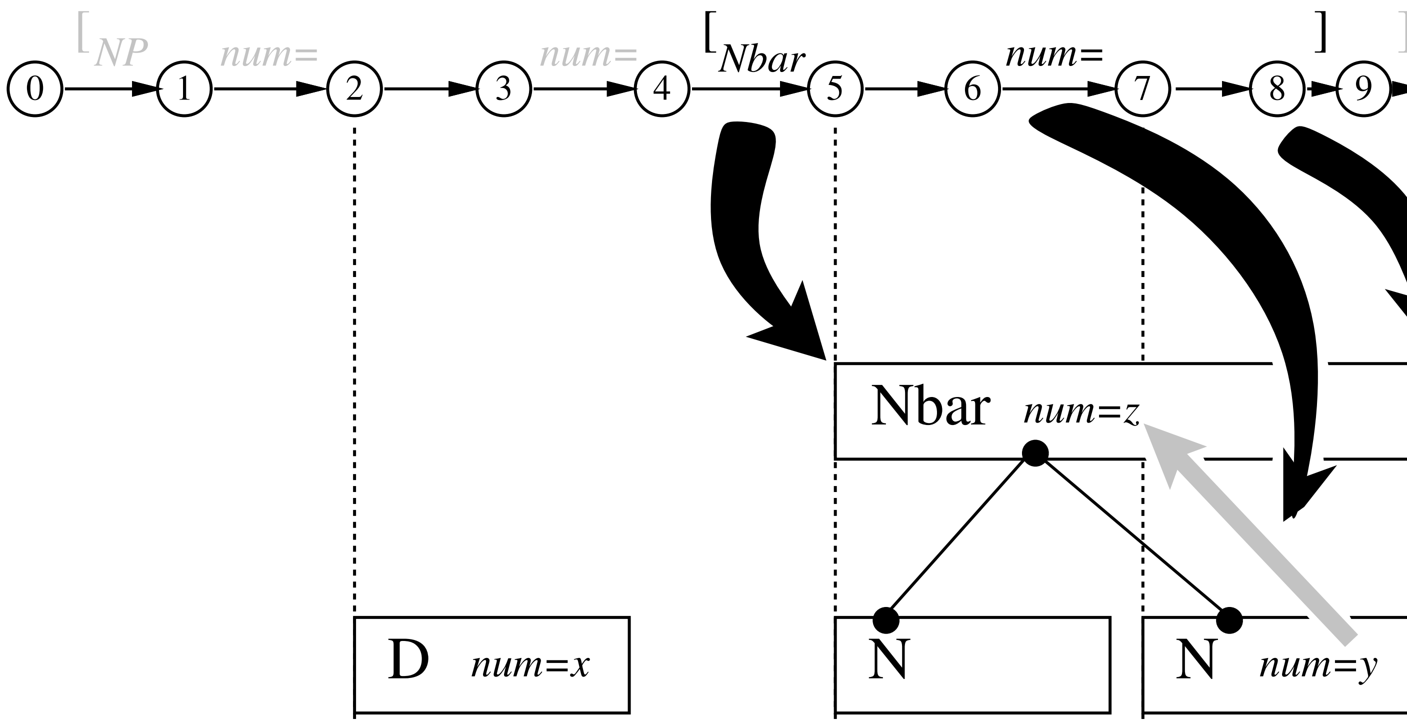
- Features
- Internal phrases

$$\text{Subj} \rightarrow [_{\text{NP}}\ \textit{num}{=}\ \text{D?}\ \textit{num}{=}\ [_{\text{NBAR}}\ \text{A}{*}\ \text{N}{*}\ \textit{num}{=}\ \text{N}\ ]\ ]\ \text{V}$$

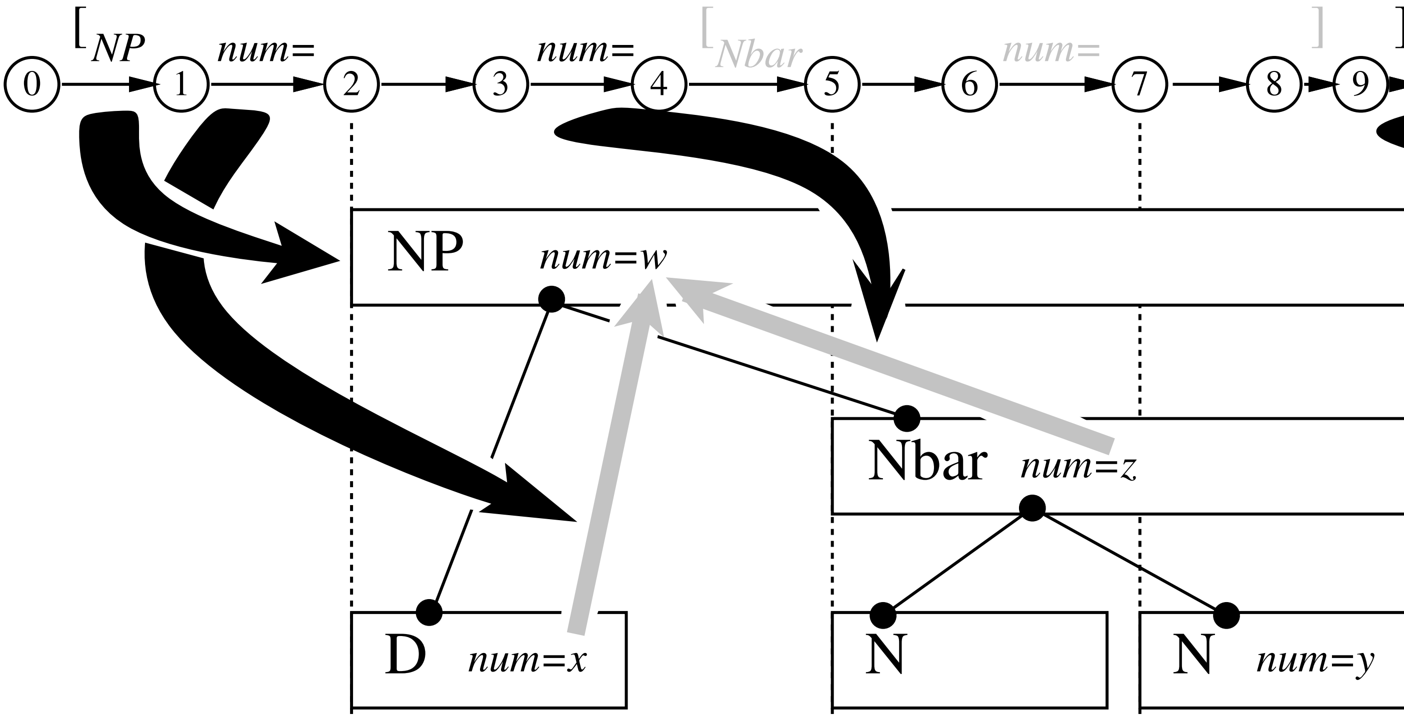


- Take ϵ-closure of left automaton → same recognizer as before

After Recognition: Parse



Parse Cont'd



[NP
num=
num=
[Nbar
num=
]
]
0
1
2
3
4
5
6
7
8
9
NP num=w
Nbar num=z
D num=x
N
N num=y

At Compile Time

- Transducer output = Action
  -] Create node, set end position
  - [Set start position, attach children
  - *ftr=* Assign feature
- Sort actions
  - Right bracket, left bracket, feature assignments
  - Inside to outside

$$\text{Subj} \rightarrow \underset{5}{[_{\text{NP}}} \;\; \underset{6}{\mathit{num}{=}_1} \; \text{D?} \;\; \underset{7}{\mathit{num}{=}_2} \;\; \underset{2}{[_{\text{Nbar}}} \; \text{A}{*} \; \text{N}{*} \;\; \underset{3}{\mathit{num}{=}_3} \; \text{N} \;\; \underset{1}{]_{\text{Nbar}}}$$

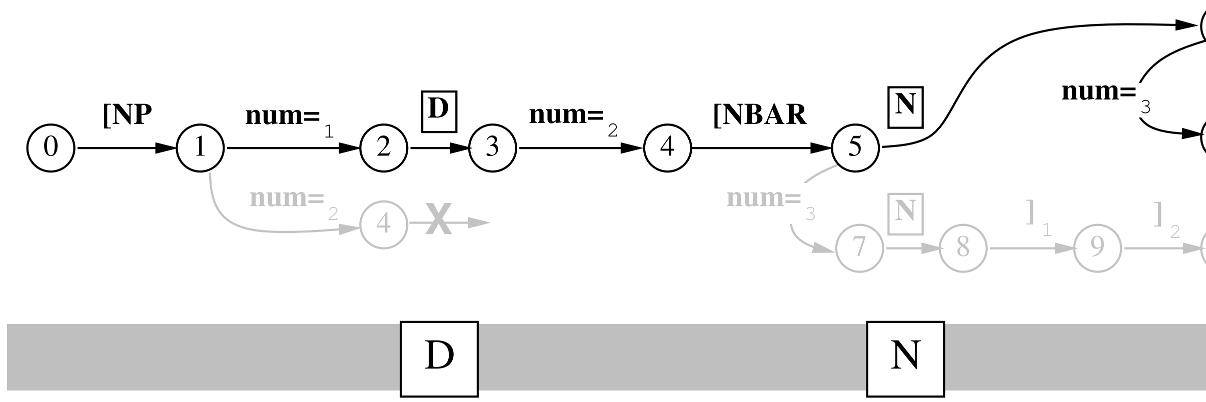
Sorted Actions

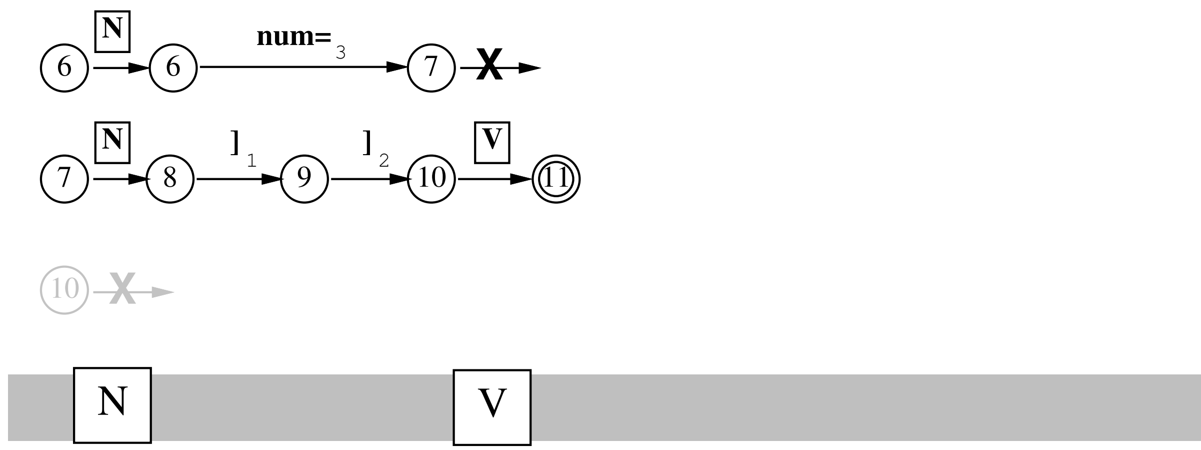
$$\text{Subj} \rightarrow [_{\text{NP}} \quad \mathit{num}{=}_1 \ \text{D?} \quad \mathit{num}{=}_2 \ [_{\text{Nbar}} \ \text{A}{*} \ \text{N}{*} \quad \mathit{num}{=}_3 \ \text{N} \ ]_{\text{Nbar}}$$

5 6 7 2 3 1

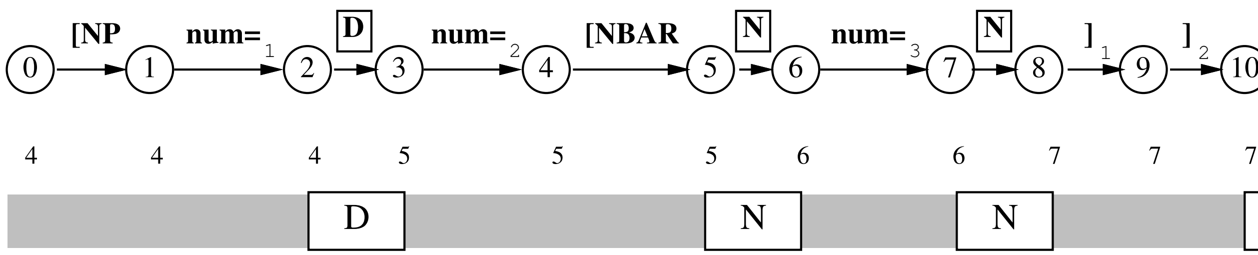
$]_{Nbar}$	☐
$[_{Nbar}$	☐
$num{=}_3$	☐
$]_{NP}$	☐
$[_{NP}$	☐
$num{=}_1$	☐
$num{=}_2$	☐

Run Transducer





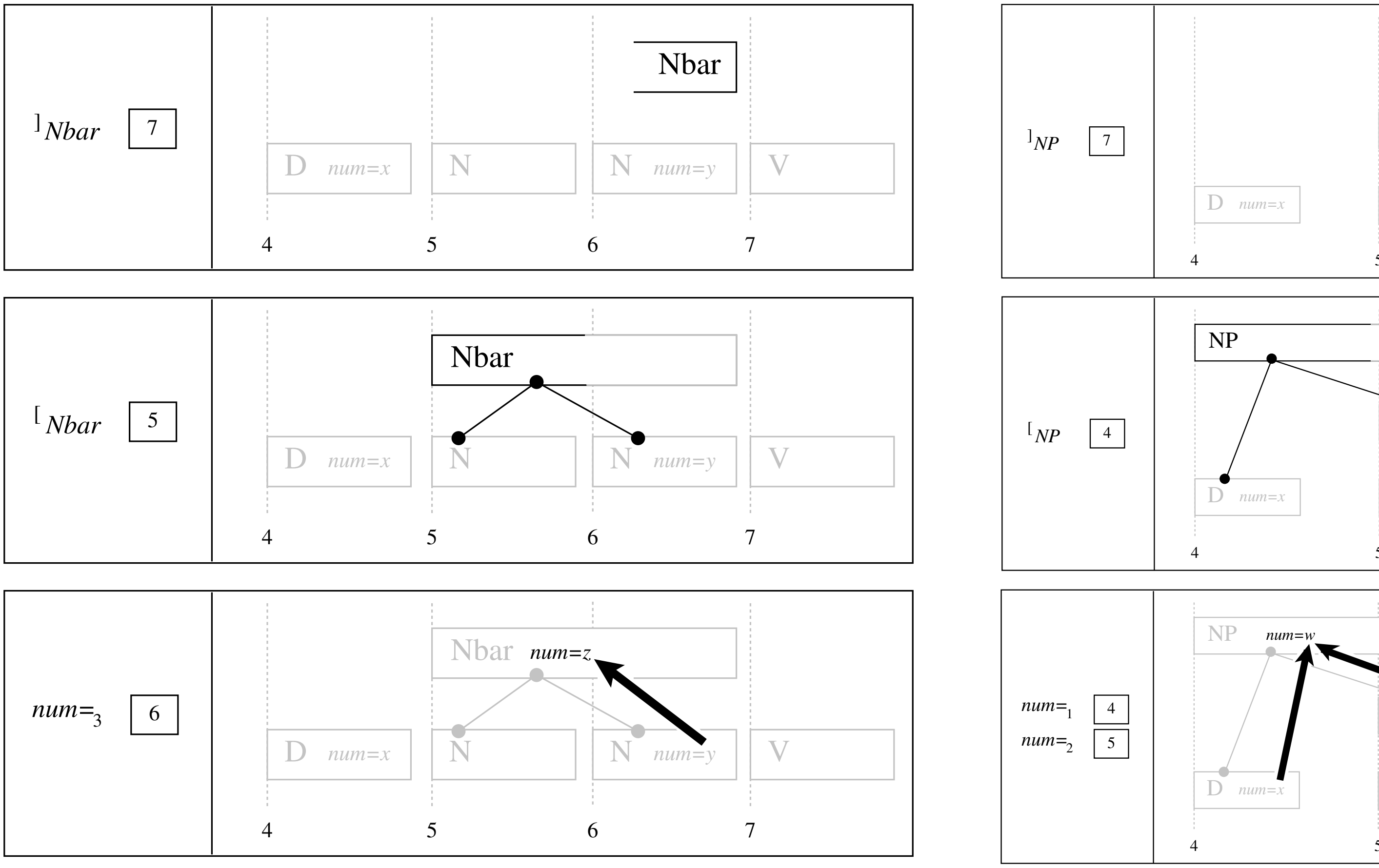
Set Positions for Actions



$]_{Nbar}$	7
$[_{Nbar}$	5
$num{=}_3$	6
$]_{NP}$	7
$[_{NP}$	4
$num{=}_1$	4
$num{=}_2$	5

- If same action appears at multiple positions, take last

Execute Actions



Wrap Up

- Restart at end of NP, *before* V
- No unification failures—bit operations
- Disambiguation—postpone actions
- Divorce control from structure

Evaluation

- Utility

  "Chunks as defined by this grammar are good because they improve the perfor
  IR/MT/... system"

  "Trees as defined by this treebank are good because ..."

  "Selectional restrictions as defined in this stylebook are good because they impro
  mance at resolving attachment ambiguities"

  – Of grammar/treebank/stylebook/specification/theory
  – Custom statistical model vs. Intuitive classes—greater utility for specific a

- Accuracy
  – At meeting specification
  – Upper bound: interjudge agreement

Sample from Corpus Positions

corp.tag:	*smp:*	*smp.conc:*		*smp.*
In in	82900	em directly	comparable , each index i	
an dt	102153	says Krys	Spain , research speciali	
Oct. month	133333	x lower for	the next fiscal year **]nx**	nx 4
19 cd	63092	imated at $	193 , according to House	nx 1
review nn	121520	only about	8,000 are written for NEC	nx 1
of of	60127	it stopped	advertising **]vgx** its name	vgx 1
“ nil	3805	eds will be	male sterile and herbicid	nx 1
The dt	147276	ry in share	prices . // The dollar fi	
Misanthrope nn	66527	o the world	that now is the time to g	
” nil	76930	h defensive	issues as food , tobacco	
at in	113209	lle , Ky.)	– David R. Jackson , for	
Chicago nnp	21391	ireproofing	concern said the transact	
’s pos	121081	lationships	rather than complex finan	
Goodman nnp	137109	n was being	“ conservative ” in his	
Theatre nnp	25825	% less full	fees via Nikko Securities	

Scoring

test file		=	s3.tst		
std file		=	s3.std		
writing diff file		=	s3.diff		
sample size:	N	=	1000		
answers in common:	X	=	921		
nonzeros in test:	t	=	390		
nonzeros in std:	s	=	394		
nonzeros in common:	x	=	343		
per-word accuracy:	X/N	=	921/1000	=	92.1 ± 1.7 %
precision:	x/t	=	343/390	=	87.9 ± 3.2 %
recall:	x/s	=	343/394	=	87.1 ± 3.3 %

Speed

- Factor 1: fast machine
- Factor 2: determinism
- Factor 3: write C

Program	depth	sw	hardware	
Fidditch3	parse	C	SGI	56[illegible]
Cass2	chunk	C	SparcELC	29[illegible]
Copsy	np	Pascal	BS2000	27[illegible]
Cass2	clause	C	SparcELC	19[illegible]
CG	dep		Sparc10	15[illegible]
Fidditch3	parse	C	Sun4	12[illegible]
Pos	tag		Sun4	2[illegible]
Fidditch2	parse	Lisp	Sun4	[illegible]
Cass1	chunk	Lisp	Sun4	[illegible]
Clarit	np	Lisp		[illegible]
Fastus	chunk	Lisp	Sparc2	[illegible]
Cass1	chunk	Lisp	UX400S	[illegible]
Scisor	skim			[illegible]
Fidditch1	parse	Lisp	Sym-36xx	[illegible]
McDonald	parse		MacII	
Chupa	parse	Lisp	UX400S	
Traditional	parse			

Summary

- Finite-State Cascade, recognize then parse
- Speed: determinism
- Robustness
  - Local decisions
  - Easy-first
  - Islands of certainty
  - Containment of ambiguity
  - Divorcing control from structure