Computational Linguistics II: Parsing

Bottom-up Parsing

Frank Richter & Jan-Philipp Söhn

fr@sfs.uni-tuebingen.de, jp.soehn@uni-tuebingen.de

Example: Bottom-Up Parsing

Grammar:

S	$\rightarrow$	NP VP | NP VP kon VP
NP	$\rightarrow$	n
VP	$\rightarrow$	v | v PP | adv v | adv v PP
PP	$\rightarrow$	p NP
n	$\rightarrow$	Beethoven | 1827 | this
v	$\rightarrow$	expired | died
p	$\rightarrow$	in | for
kon	$\rightarrow$	and
adv	$\rightarrow$	later

Example: Bottom-Up Parsing

Grammar:

S	$\rightarrow$	NP VP | NP VP kon VP
NP	$\rightarrow$	n
VP	$\rightarrow$	v | v PP | adv v | adv v PP
PP	$\rightarrow$	p NP
n	$\rightarrow$	Beethoven | 1827 | this
v	$\rightarrow$	expired | died
p	$\rightarrow$	in | for
kon	$\rightarrow$	and
adv	$\rightarrow$	later

Sentence: Beethoven expired in 1827 and later died for this.

Bottom-Up Parsing

a bottom-up parser has 2 different methods:

Bottom-Up Parsing

a bottom-up parser has 2 different methods:

- **shift**: “recognize” the next word in the input

Bottom-Up Parsing

a bottom-up parser has 2 different methods:

- **shift**: “recognize” the next word in the input
- **reduce**: match already recognized symbols with a righthand side of a rule and replace them by the lefthand side

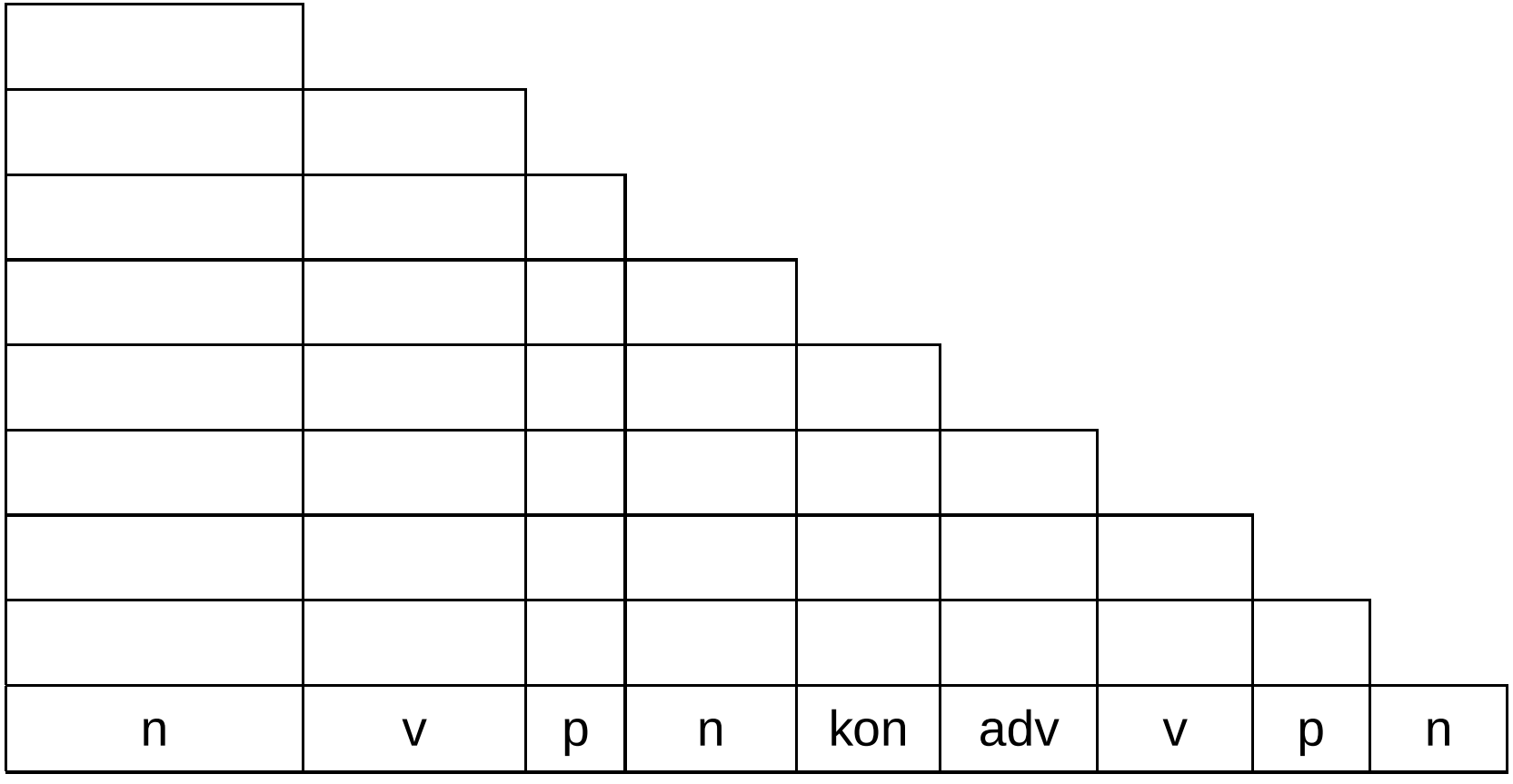
Bottom-Up Parsing

a bottom-up parser has 2 different methods:

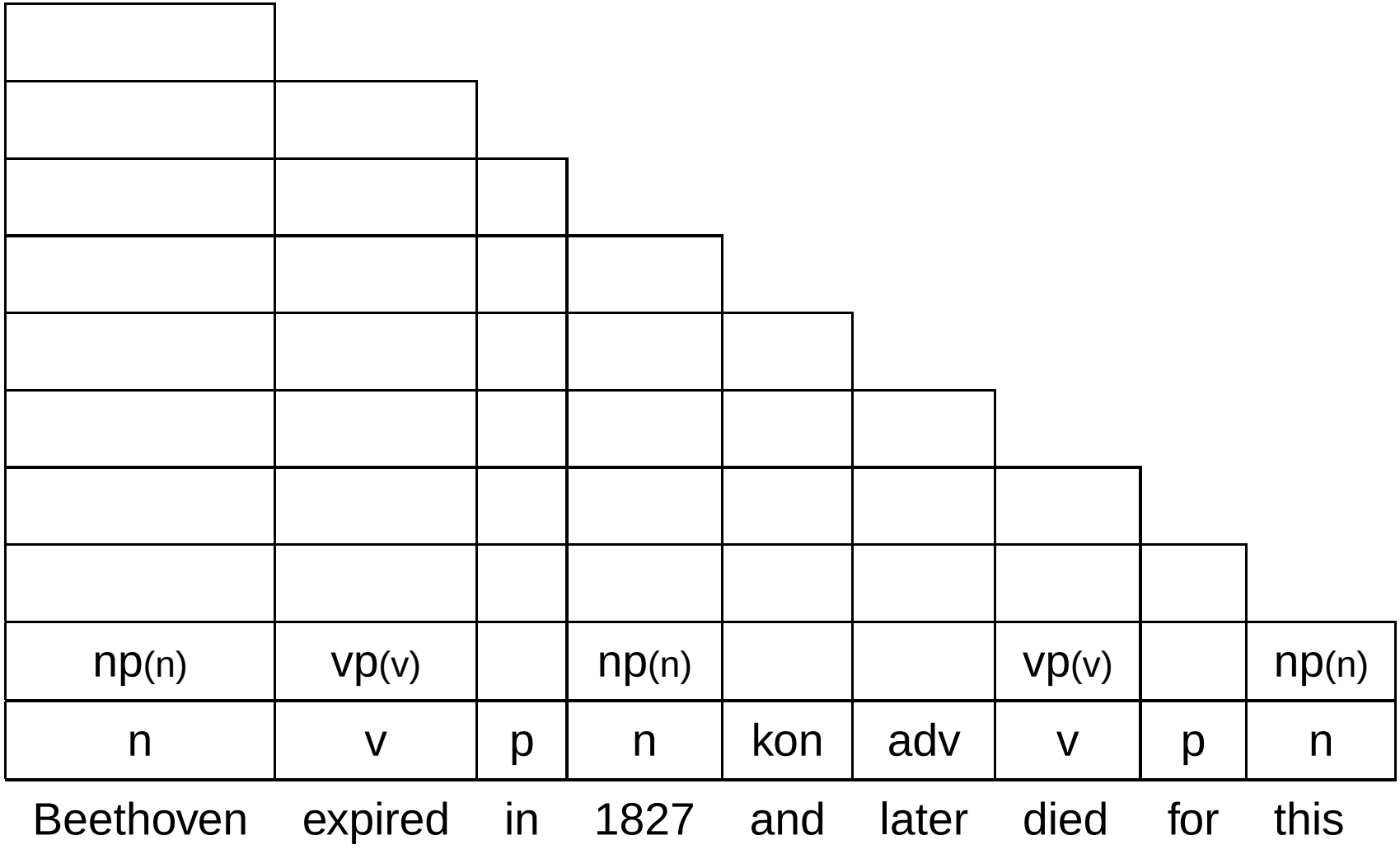
- **shift**: “recognize” the next word in the input
- **reduce**: match already recognized symbols with a righthand side of a rule and replace them by the lefthand side

problem: often reduces in wrong situations

Bottom-Up Parsing: Chart



Bottom-Up Parsing: Chart



Bottom-Up Parsing: Chart

		pp(p np)			vp(a v)		pp(p np)	
np(n)	vp(v)		np(n)			vp(v)		np(n)
n	v	p	n	kon	adv	v	p	n
B.	expired	in	1827	and	later	died	for	this

Bottom-Up Parsing: Chart

	vp(v pp)					vp(v pp)		
		pp(p np)			vp(a v)		pp(p np)	
np(n)	vp(v)		np(n)			vp(v)		np(n)
n	v	p	n	kon	adv	v	p	n
B.	expired	in	1827	and	later	died	for	this

Bottom-Up Parsing: Chart

s(np vp)					vp(a v pp)			
	vp(v pp)					vp(v pp)		
		pp(p np)			vp(a v)		pp(p np)	
np(n)	vp(v)		np(n)			vp(v)		np(n)
n	v	p	n	kon	adv	v	p	n
B.	expired	in	1827	and	later	died	for	this

Bottom-Up Parsing: Chart

S(np vp kon vp)								
S(np vp)					vp(a v pp)			
	vp(v pp)					vp(v pp)		
		pp(p np)			vp(a v)		pp(p np)	
np(n)	vp(v)		np(n)			vp(v)		np(n)
n	v	p	n	kon	adv	v	p	n
B.	expired	in	1827	and	later	died	for	this

Bottom-Up Parsing: Chart

unnecessary reductions: 5 / 13 = 38.5%

S(np vp kon vp)								
S(np vp)					vp(a v pp)			
	vp(v pp)					vp(v pp)		
		pp(p np)			vp(a v)		pp(p np)	
np(n)	vp(v)		np(n)			vp(v)		np(n)
n	v	p	n	kon	adv	v	p	n
B.	expired	in	1827	and	later	died	for	this

Combining Top-Down and Bottom-Up

- problem with bottom-up: many reductions cannot be derived from start symbol
  e.g. VP $\rightarrow$ V NP PP
  `the English put tacks in their tea` (which started the Revolutionary Wars)
  no sense in grouping [`tacks in their tea`]

Combining Top-Down and Bottom-Up

- problem with bottom-up: many reductions cannot be derived from start symbol
  e.g. VP $\rightarrow$ V NP PP
  `the English put tacks in their tea` (which started the Revolutionary Wars)
  no sense in grouping [`tacks in their tea`]
- Earley's idea: if a constituent is found, this restricts the possibilities for the next constituents – because of higher rules

Combining Top-Down and Bottom-Up

- problem with bottom-up: many reductions cannot be derived from start symbol
  e.g. VP $\rightarrow$ V NP PP
  `the English put tacks in their tea` (which started the Revolutionary Wars)
  no sense in grouping [`tacks in their tea`]
- Earley's idea: if a constituent is found, this restricts the possibilities for the next constituents – because of higher rules
- ex.: found rule NP (1,1) $\Rightarrow$ find all rules which have NP as first symbol on righthand side:
  S $\rightarrow$ NP VP | NP VP kon VP

Combining Top-Down and Bottom-Up

- problem with bottom-up: many reductions cannot be derived from start symbol
  e.g. VP $\rightarrow$ V NP PP
  `the English put tacks in their tea` (which started the Revolutionary Wars)
  no sense in grouping [`tacks in their tea`]
- Earley's idea: if a constituent is found, this restricts the possibilities for the next constituents – because of higher rules
- ex.: found rule NP (1,1) $\Rightarrow$ find all rules which have NP as first symbol on righthand side:
  S $\rightarrow$ NP VP | NP VP kon VP
- only VPs should be checked next!