Parsing as Search

- Search through possible parse trees
- Want one (or more) that derive input
- Formally, search problems are defined by:
  - Start state S,
  - Goal state G,
  - Successor Function:
    Transitions between states,
  - Path cost function

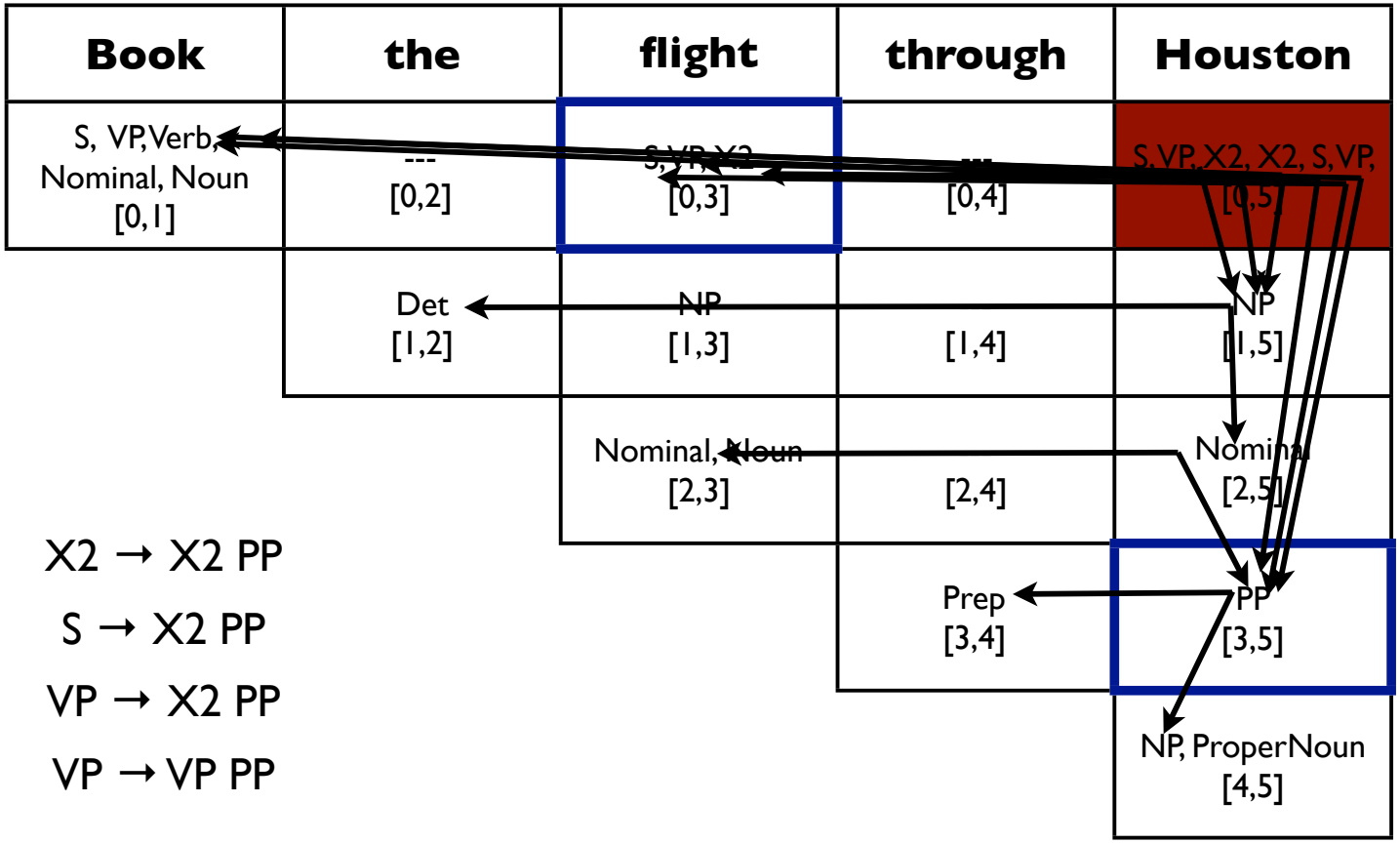
Parsing w/Dynamic Programming

- Makes parsing algorithms (relatively) efficient
  - Polynomial time in input length
  - Typically cubic (n^3) or less
- Several different implementations
  - Cocke-Kasami-Younger (CKY) algorithm
  - Earley algorithm
  - Chart parsing

Chomsky Normal Form (CNF)

- CKY parsing requires grammars in CNF
- All productions of the form:
  - A → B C, or
  - A → a
- Most of our grammars are not of this form
  E.g., S -> Wh-NP Aux NP VP
- Need a general conversion procedue

CKY Example



X2 → X2 PP

S → X2 PP

VP → X2 PP

VP → VP PP

Learning Probabilities

- Simplest: Treebank of parsed sentences
  - To compute probability of a rule, count:
    - Times LHS is expanded
    - Times LHS expands to RHS

$$P(\alpha \rightarrow \beta \mid \alpha) = \frac{Count(\alpha \rightarrow \beta)}{\sum_{\gamma} Count(\alpha \rightarrow \gamma)} = \frac{Count(\alpha \rightarrow \beta)}{Count(\alpha)}$$

Example PCFG

Grammar		Lexicon
$S \rightarrow NP\ VP$	[.80]	$Det \rightarrow that\ [.10] \mid a\ [.30] \mid the\ [.60]$
$S \rightarrow Aux\ NP\ VP$	[.15]	$Noun \rightarrow book\ [.10] \mid flight\ [.30]$
$S \rightarrow VP$	[.05]	$\mid meal\ [.15] \mid money\ [.05]$
$NP \rightarrow Pronoun$	[.35]	$\mid flights\ [.40] \mid dinner\ [.10]$
$NP \rightarrow Proper\text{-}Noun$	[.30]	$Verb \rightarrow book\ [.30] \mid include\ [.30]$
$NP \rightarrow Det\ Nominal$	[.20]	$\mid prefer;\ [.40]$
$NP \rightarrow Nominal$	[.15]	$Pronoun \rightarrow I\ [.40] \mid she\ [.05]$
$Nominal \rightarrow Noun$	[.75]	$\mid me\ [.15] \mid you\ [.40]$
$Nominal \rightarrow Nominal\ Noun$	[.20]	$Proper\text{-}Noun \rightarrow Houston\ [.60]$
$Nominal \rightarrow Nominal\ PP$	[.05]	$\mid NWA\ [.40]$
$VP \rightarrow Verb$	[.35]	$Aux \rightarrow does\ [.60] \mid can\ [40]$
$VP \rightarrow Verb\ NP$	[.20]	$Preposition \rightarrow from\ [.30] \mid to\ [.30]$
$VP \rightarrow Verb\ NP\ PP$	[.10]	$\mid on\ [.20] \mid near\ [.15]$
$VP \rightarrow Verb\ PP$	[.15]	$\mid through\ [.05]$
$VP \rightarrow Verb\ NP\ NP$	[.05]	
$VP \rightarrow VP\ PP$	[.15]	
$PP \rightarrow Preposition\ NP$	[1.0]	

Parser Issues

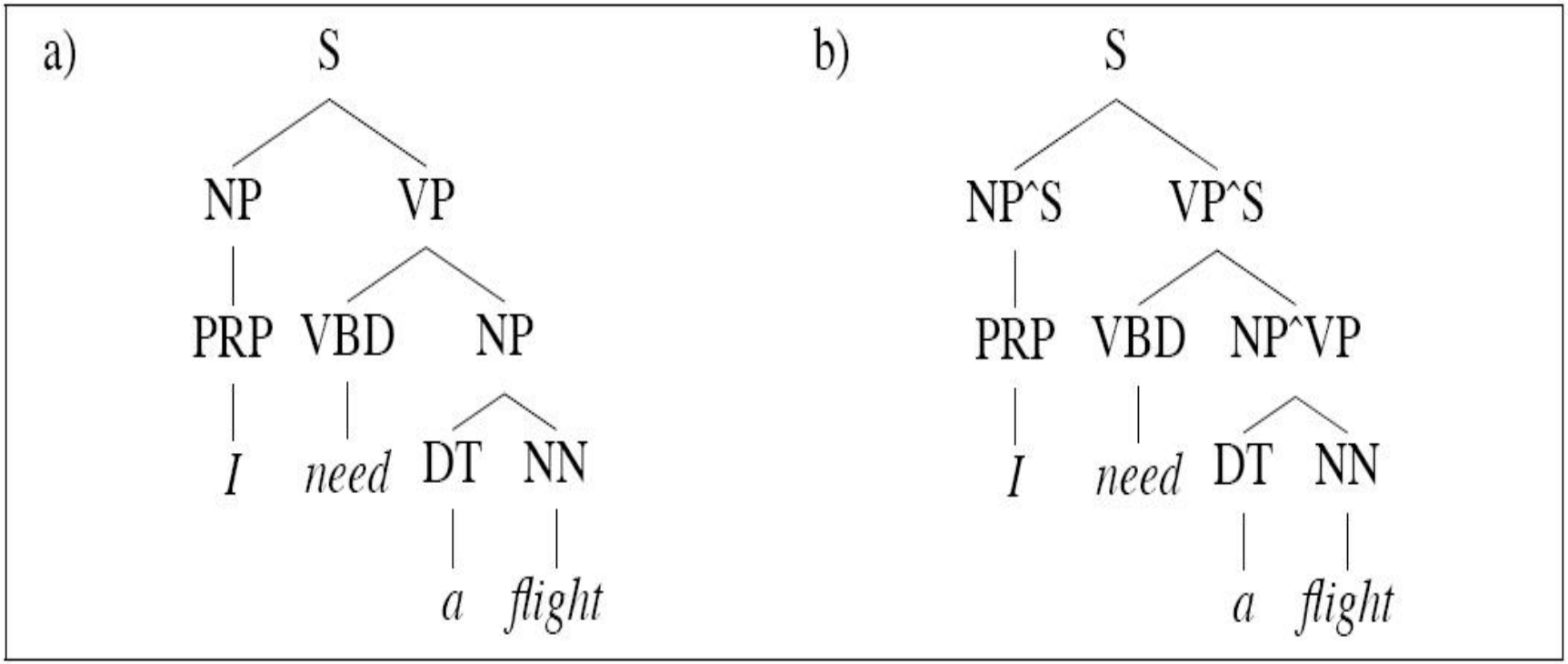
- PCFGs make many (unwarranted) independence assumptions
  - Structural Dependency
    - NP -> Pronoun: much more likely in subject position
  - Lexical Dependency
    - Verb subcategorization
    - Coordination ambiguity

PCFGs & Independence

	Pronoun	Non-Pronoun
Subject	91%	9%
Object	34%	66%

- In Treebank: roughly equi-probable
- How can we handle this?
- Condition on Subj/Obj with parent annotation

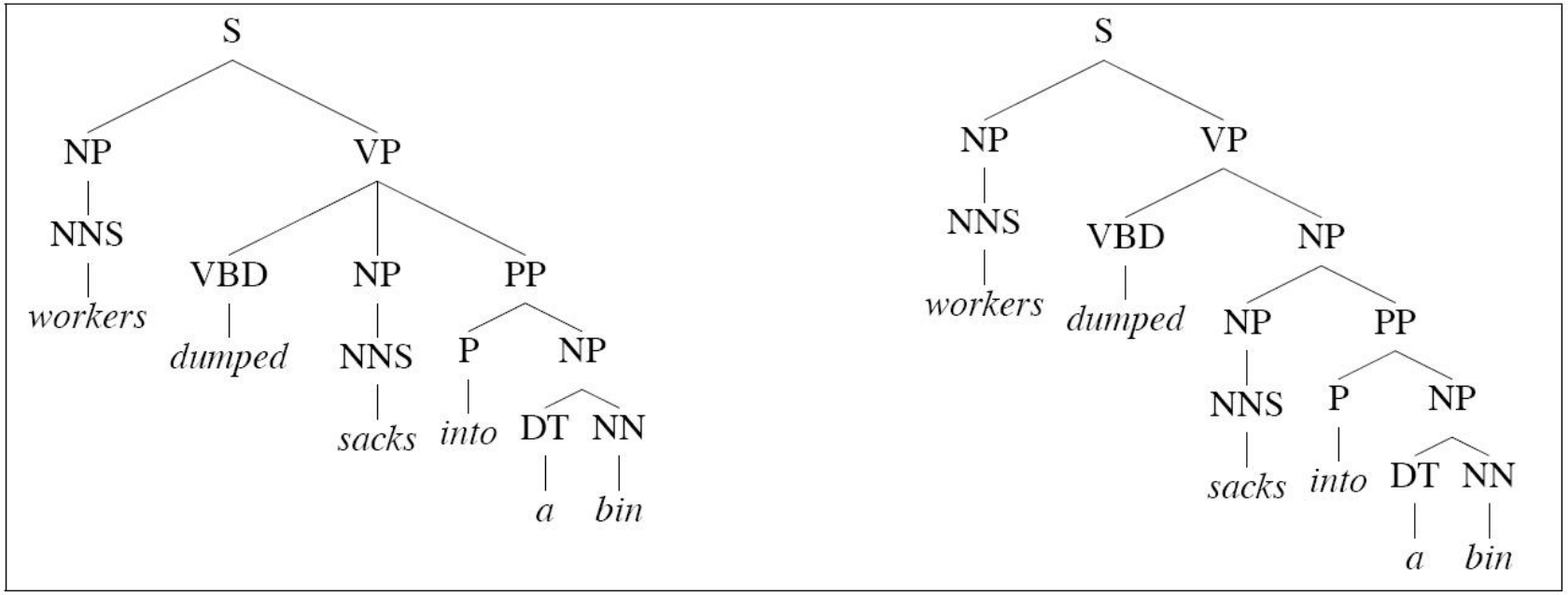
Parent Annotation



Parent Annotation

- Advantages:
  - Captures structural dependency in grammars
- Disadvantages:
  - Increases number of rules in grammar
  - Decreases amount of training per rule
  - Need to search for optimal # of rules

Lexical Conditioning



- Different verbs, prepositions have different attachment preferences

Improving PCFGs: Lexicalized Rules

- Conceptually, add 1 rule per head value

VP(dumped) →
VBD(dumped)NP(sacks)PP(into)

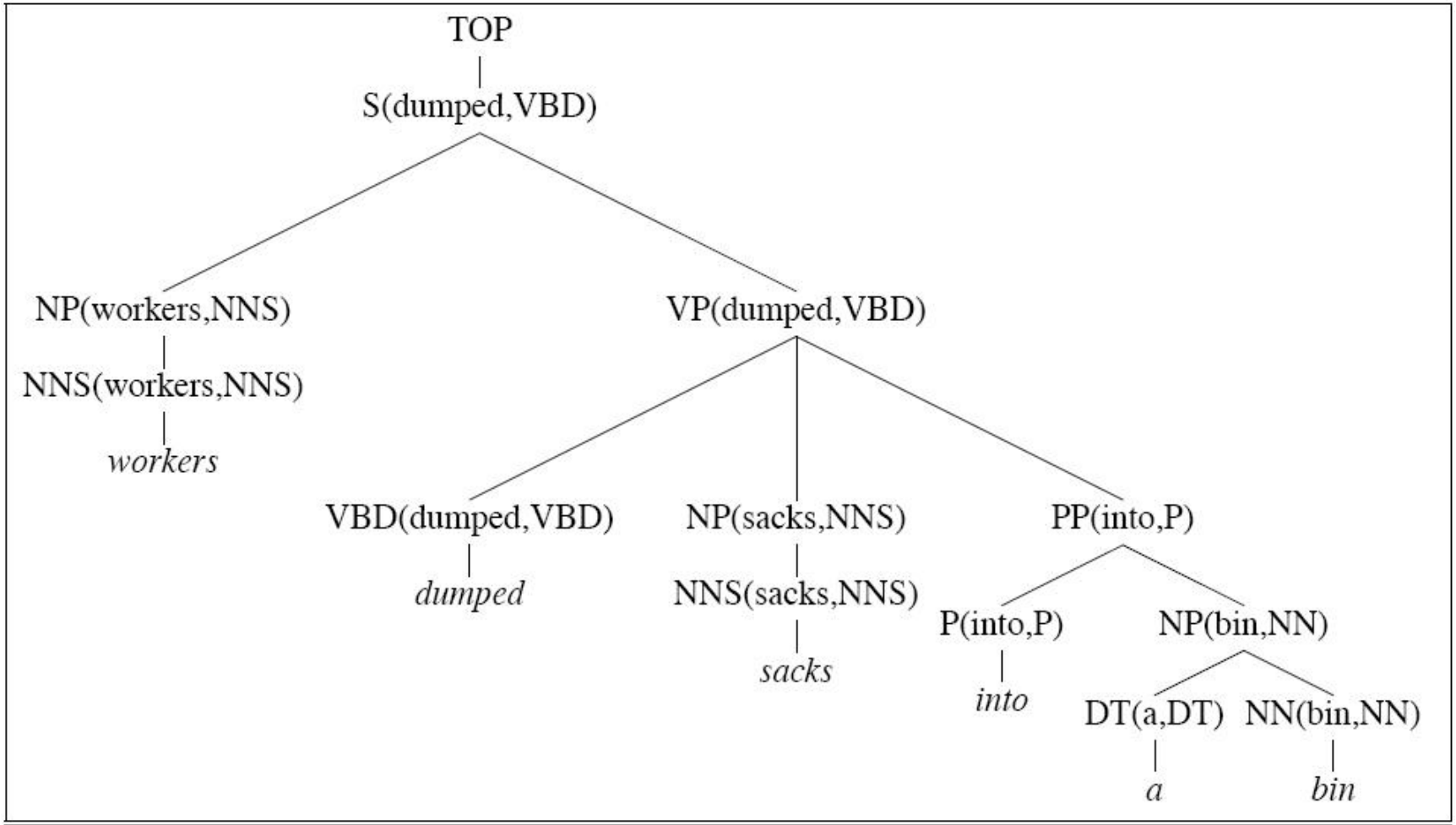
VP(dumped) →
VBD(dumped)NP(cats)PP(into)

Lexicalized PCFGs

- Also add head tag to non-terminals
  - Head tag: Part-of-speech tag of head word

VP(dumped) →
VBD(dumped)NP(sacks)PP(into)

VP(dumped,VBD) →
VBD(dumped,VBD)NP(sacks,NNS)PP(into,IN)



Internal Rules					**Lexical Rules**		
TOP	→	S(dumped,VBD)			NNS(workers,NNS)	→	workers
S(dumped,VBD)	→	NP(workers,NNS)	VP(dumped,VBD)		VBD(dumped,VBD)	→	dumped
NP(workers,NNS)	→	NNS(workers,NNS)			NNS(sacks,NNS)	→	sacks
VP(dumped,VBD)	→	VBD(dumped, VBD)	NP(sacks,NNS)	PP(into,P)	P(into,P)	→	into
PP(into,P)	→	P(into,P)	NP(bin,NN)		DT(a,DT)	→	a
NP(bin,NN)	→	DT(a,DT)	NN(bin,NN)		NN(bin,NN)	→	bin

Improving PCFGs: Tradeoffs

- Tensions:
  - Increase accuracy from increased specificity
    - E.g. Lexicalizing, Parent annotation
  - Increased grammar size
    - Increases processing times
    - Increases training data requirements
- How can we balance?

Why features?

- Need compact, general constraints
  - S -> NP VP *iff* NP and VP agree
- Decompose into elementary features
  - Agreement, subcat: consistency requirements on those features (E.g. number, person, gender)
- Augment CF rules with feature constraints
  - Develop mechanism to enforce consistency
  - Elegant, compact, rich representation

Unification

- Given features for different constituents,
  - When can we combine two feature sets?
  - How can we combine two feature sets?

Reading a Research Paper

- Who is the audience?
- Our questions:
  - What is the main goal?
  - What data did the authors use?
  - How hard was the problem?
  - Were they able to solve it?