

# **Grammars and Parsing**

**CSCI 1460: Computational Linguistics**

**Lecture 17**

**Ellie Pavlick  
Fall 2022**

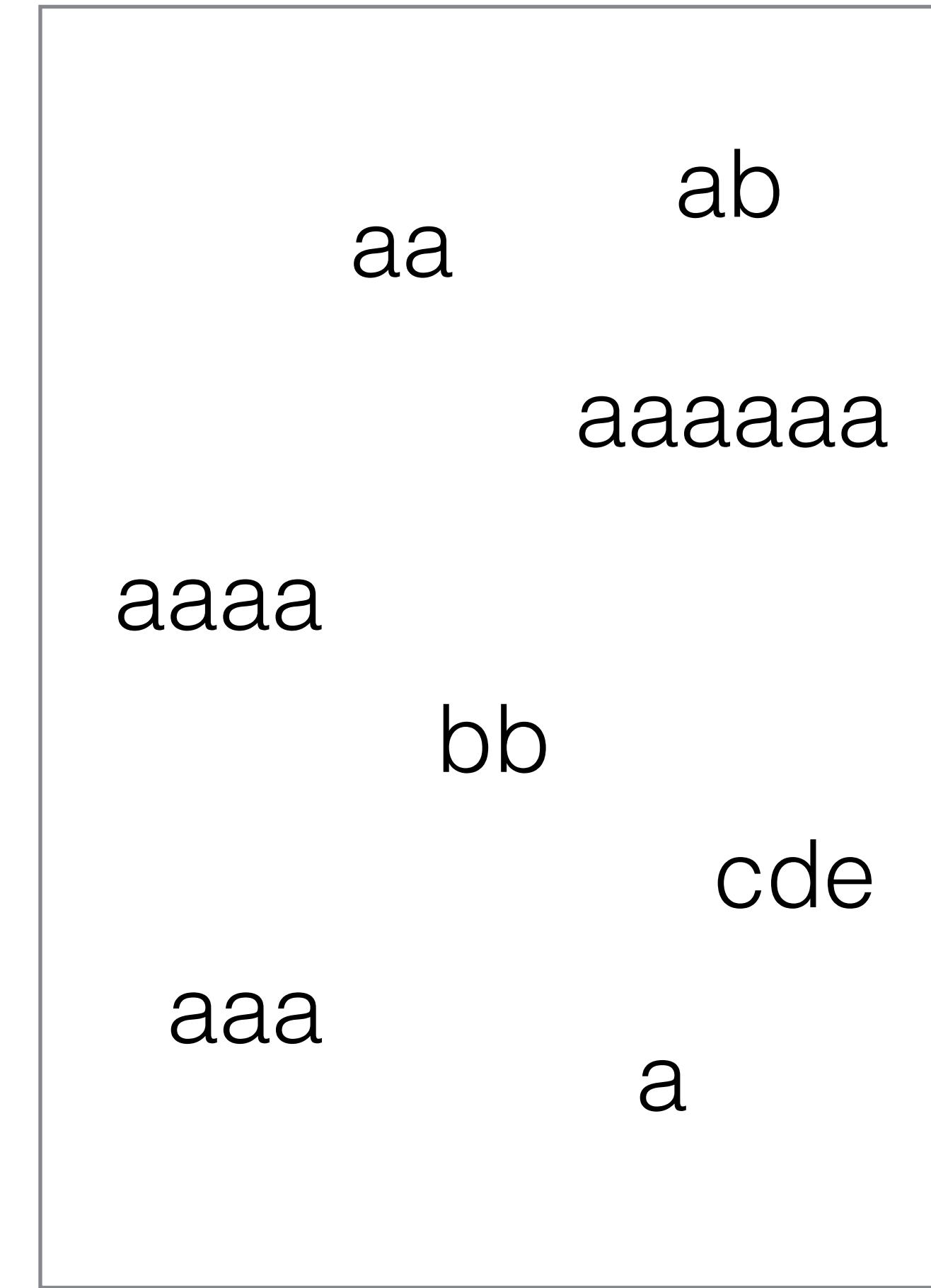
# Topics

- **Formal Grammars**
- Constituency Parsing with the CKY Algorithm
- Dependency Parsing with the Shift-Reduce Algorithm

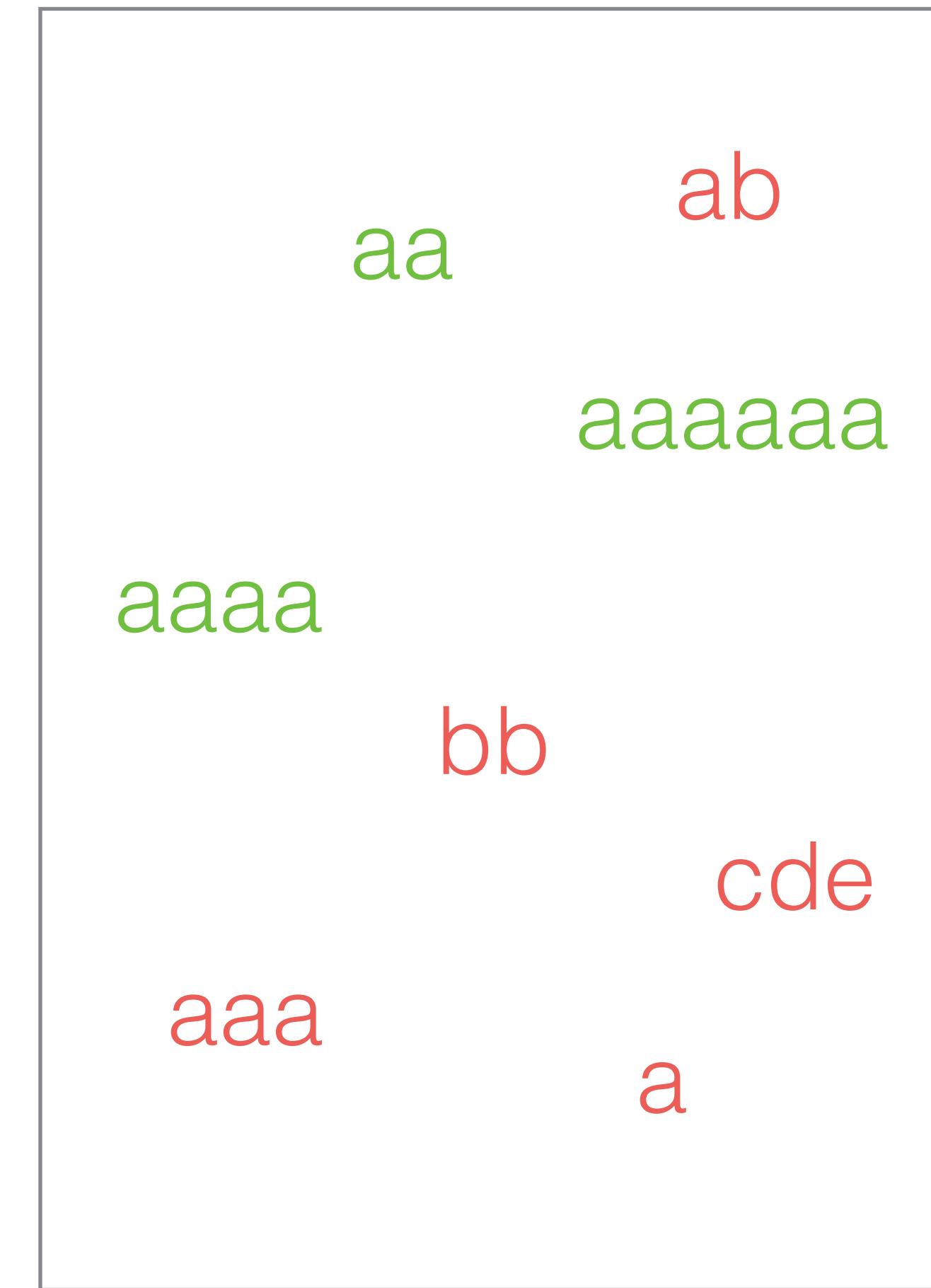
# Formal Grammar

- Language: (Possibly infinite) set of strings
- Grammar: Set of rewrite rules which define a language
  - Following the rules will generate all the strings that are in the language and none of the strings that are not in the language
  - Allows us to precisely define an (infinite) formal language
- Used also in Programming Languages, Theory of Computation...

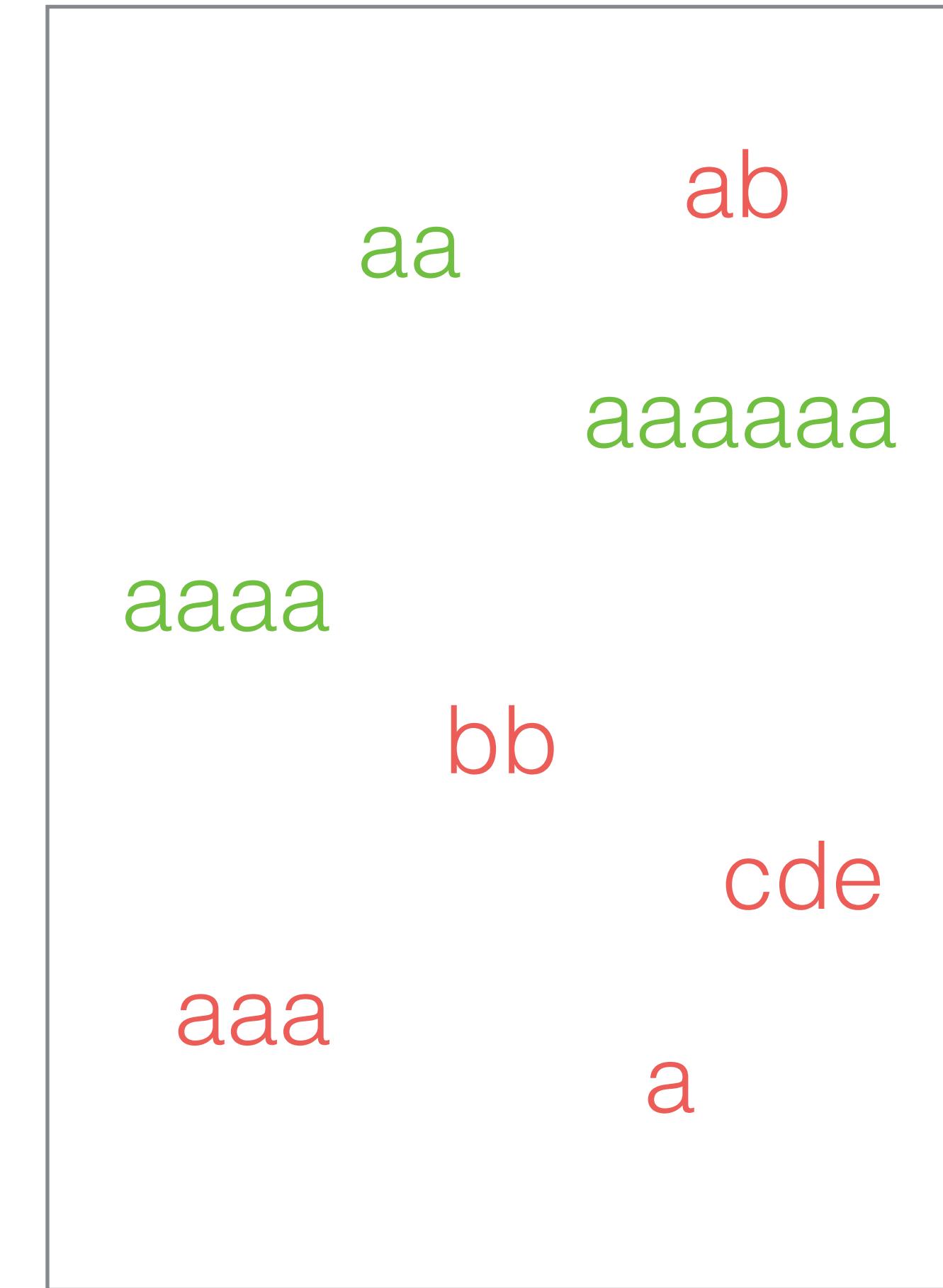
# Formal Grammar



# Formal Grammar



# Formal Grammar

$$\begin{array}{l} S \rightarrow A \\ A \rightarrow aa \\ A \rightarrow Aaa \end{array}$$


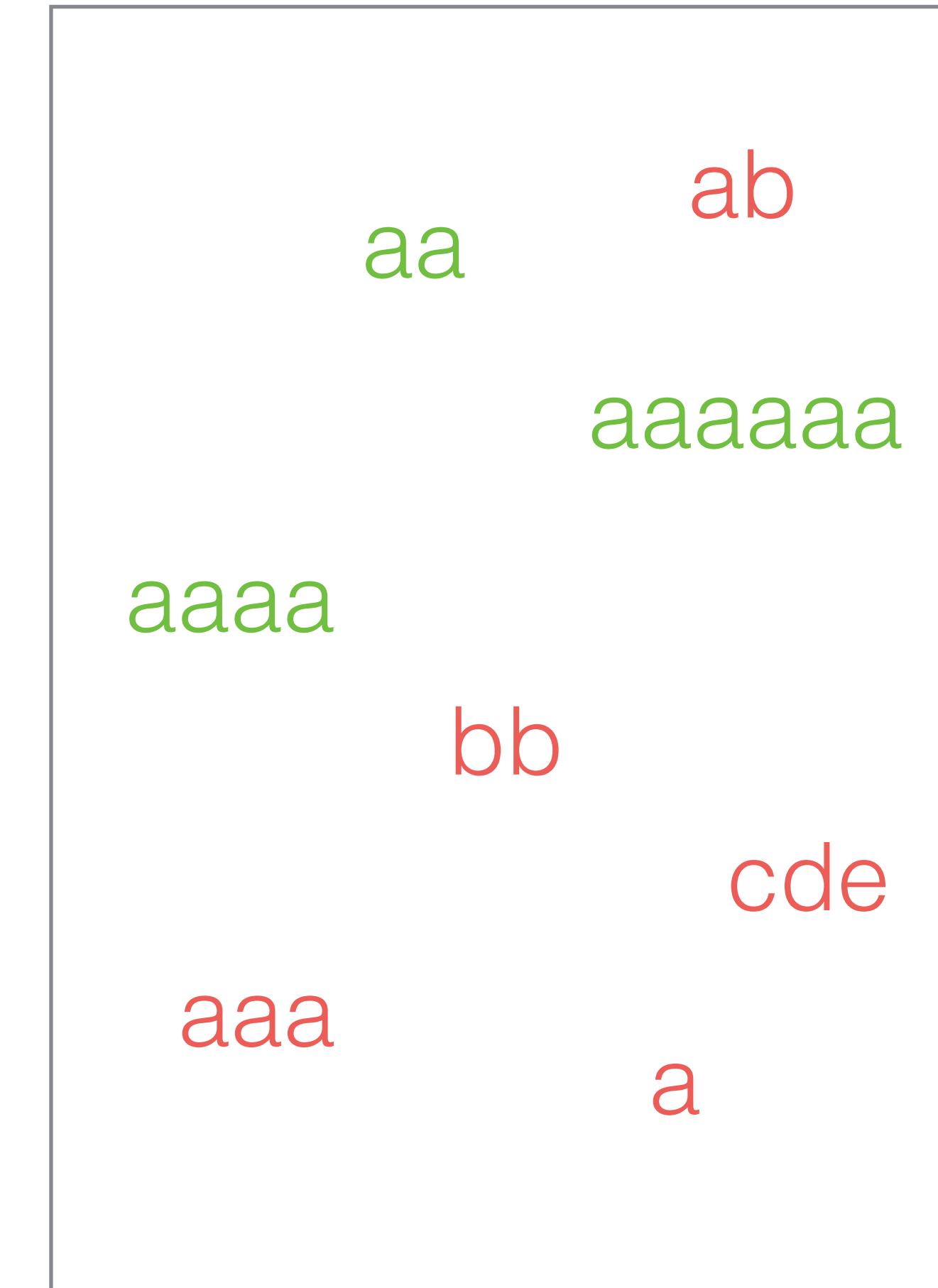
# Formal Grammar

start symbol

$$S \rightarrow A$$

$$A \rightarrow aa$$

$$A \rightarrow Aaa$$



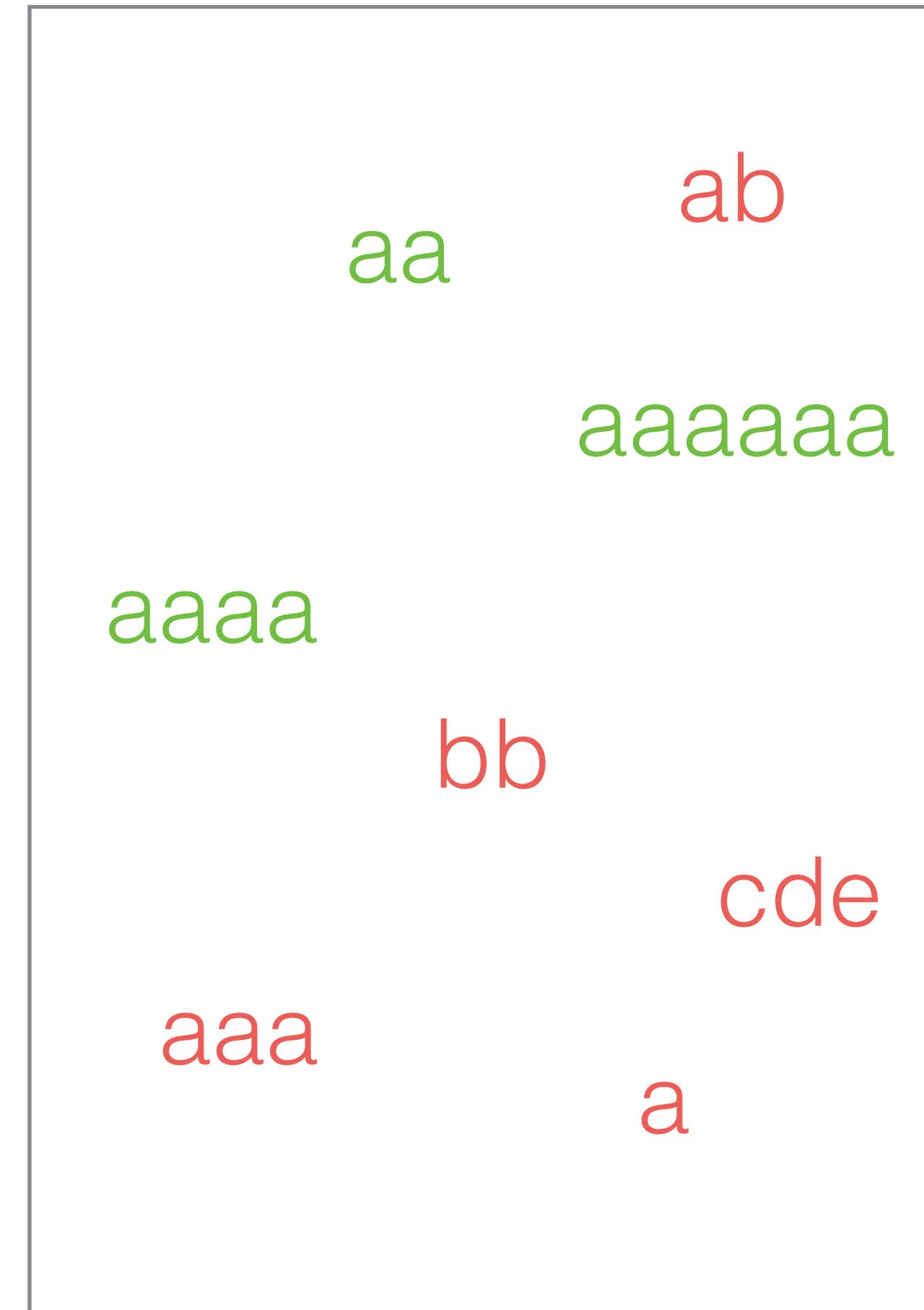
# Formal Grammar

non-terminal symbol

$$S \rightarrow A$$

$$A \rightarrow aa$$

$$A \rightarrow Aaa$$



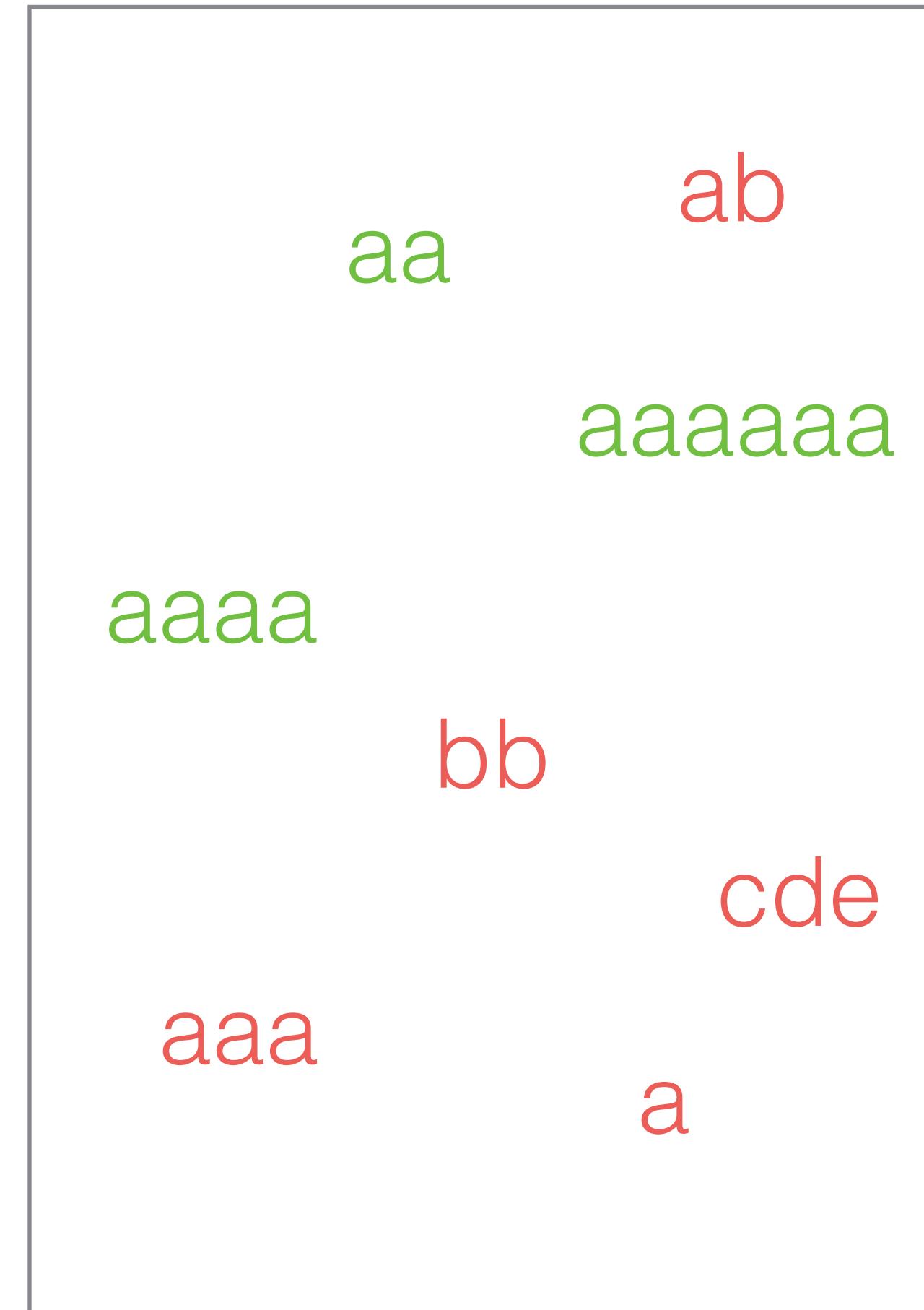
# Formal Grammar

terminal symbol

$$S \rightarrow A$$

$$A \rightarrow aa$$

$$A \rightarrow Aaa$$



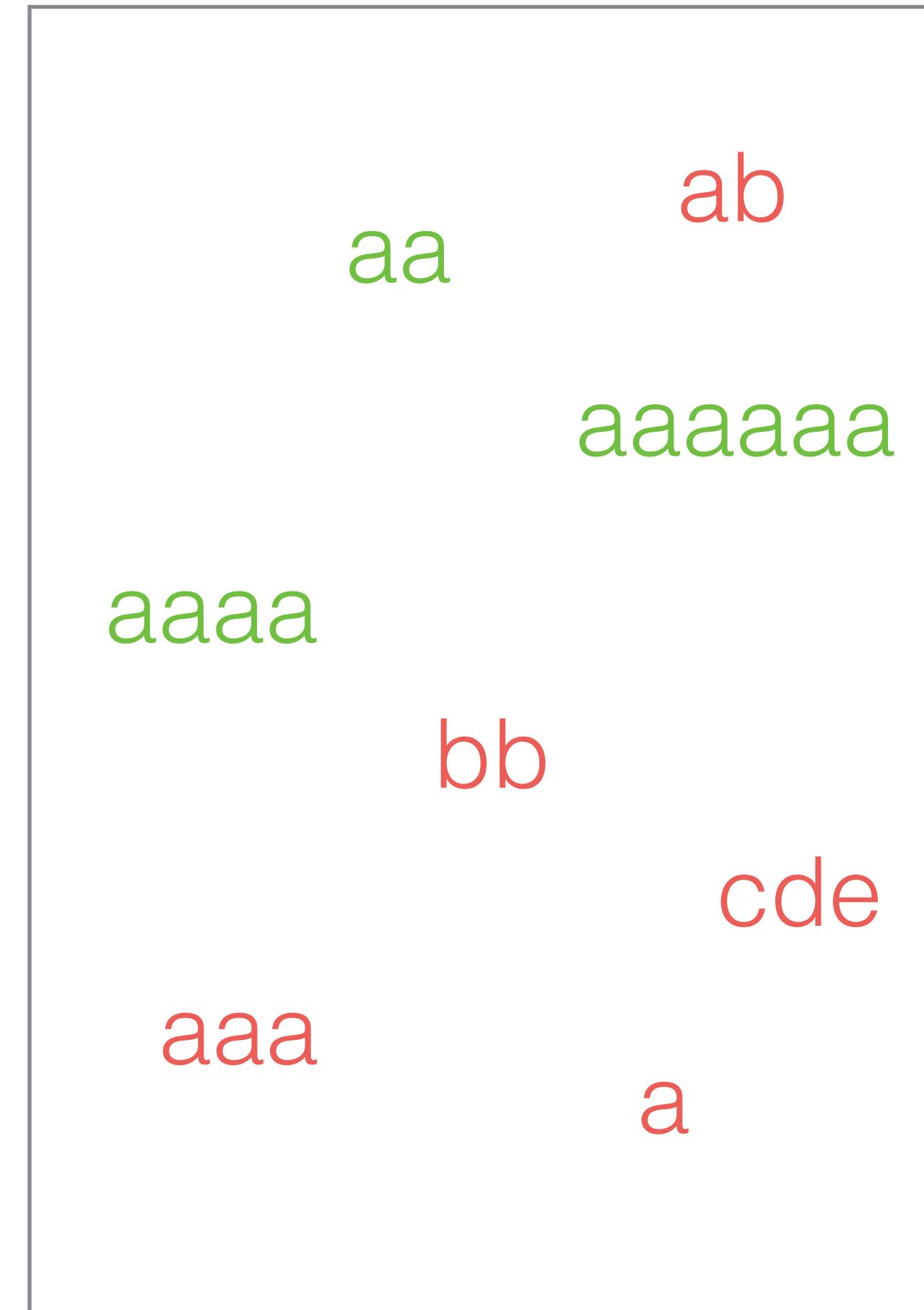
# Formal Grammar

production rule/  
rewrite rule

$$S \rightarrow A$$

$$A \rightarrow aa$$

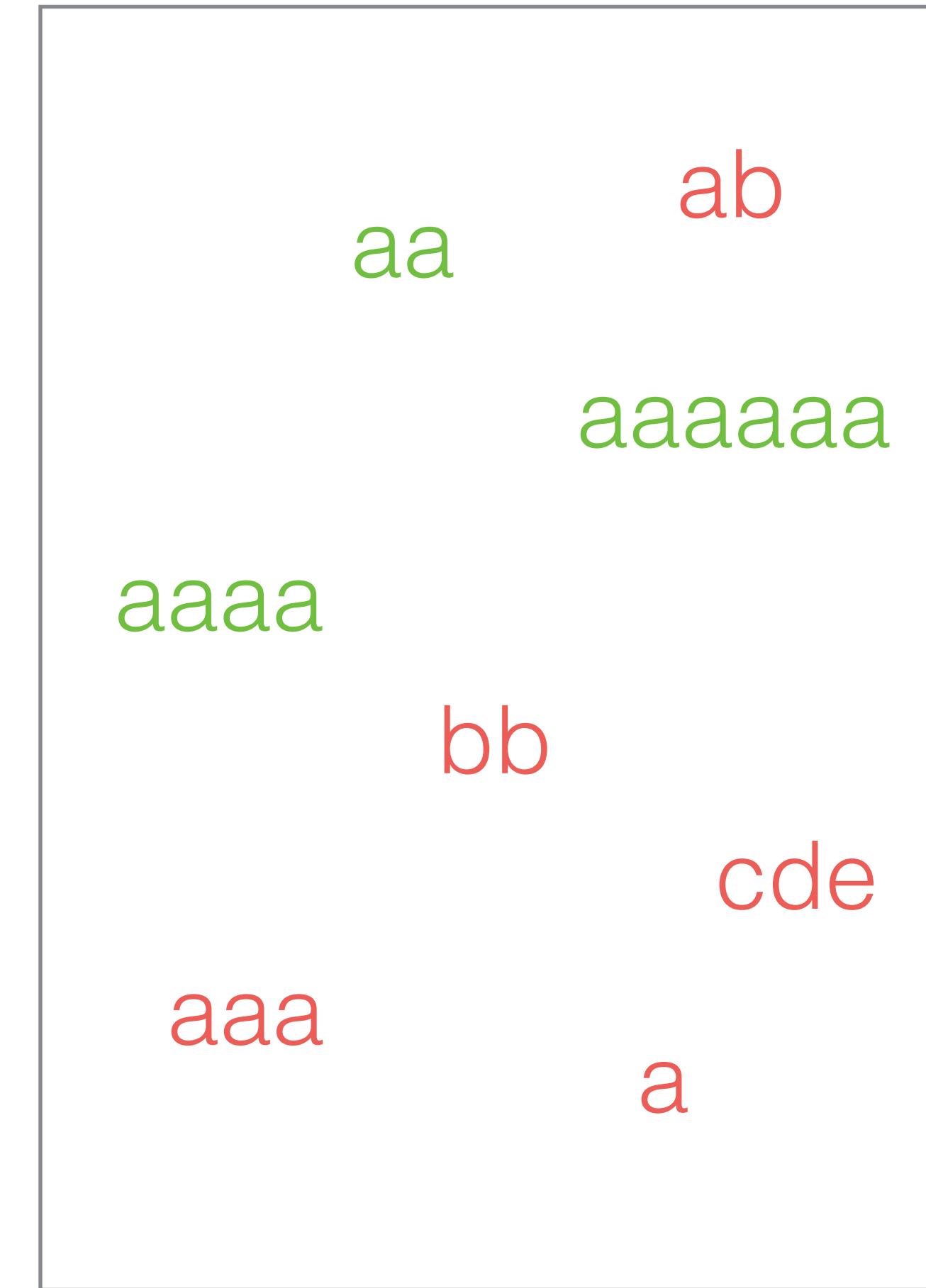
$$A \rightarrow Aaa$$



# Formal Grammar

```
S → A  
A → aa  
A → Aaa
```

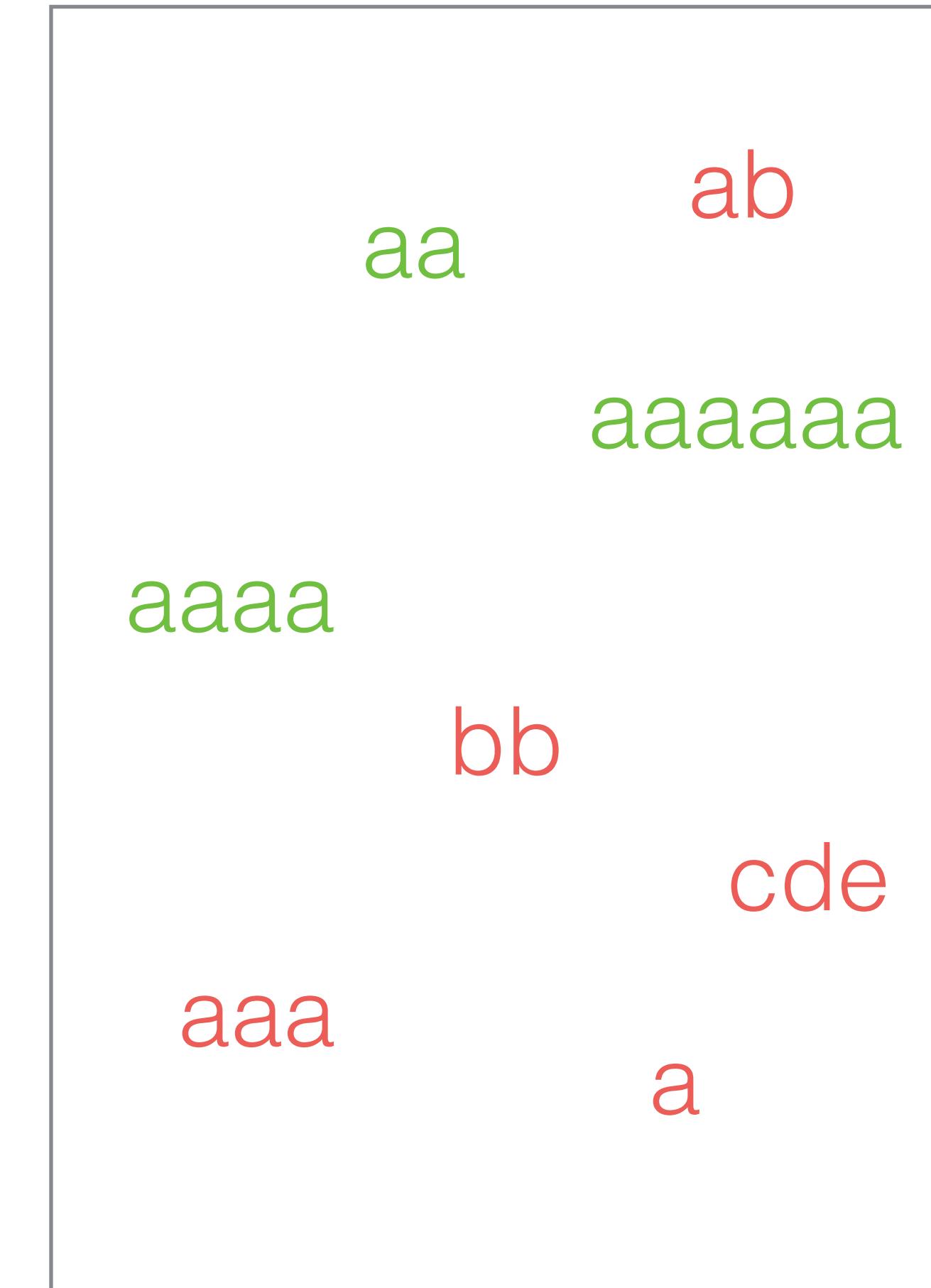
S



# Formal Grammar

```
S → A  
A → aa  
A → Aaa
```

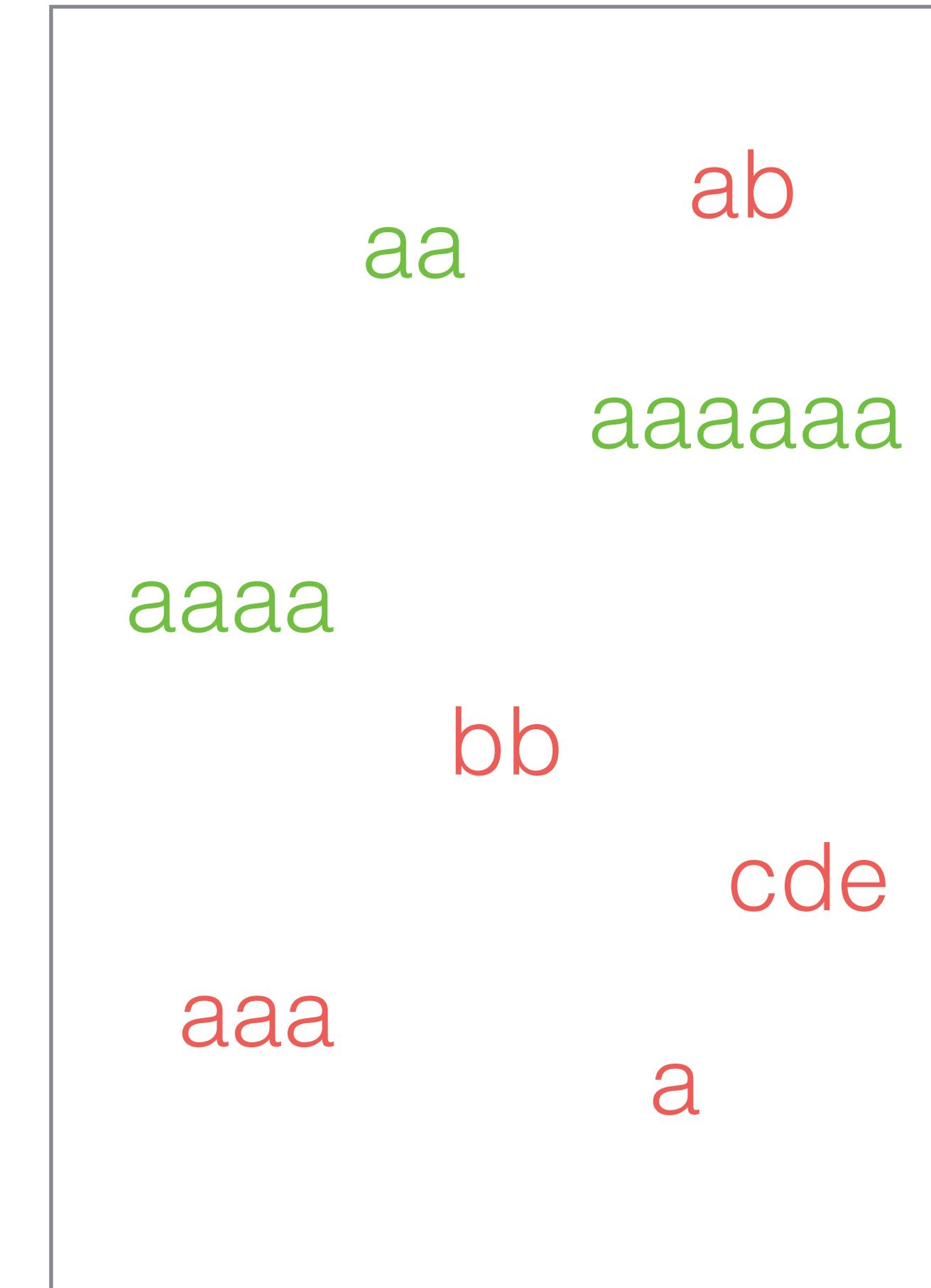
S



# Formal Grammar

```
S → A  
A → aa  
A → Aaa
```

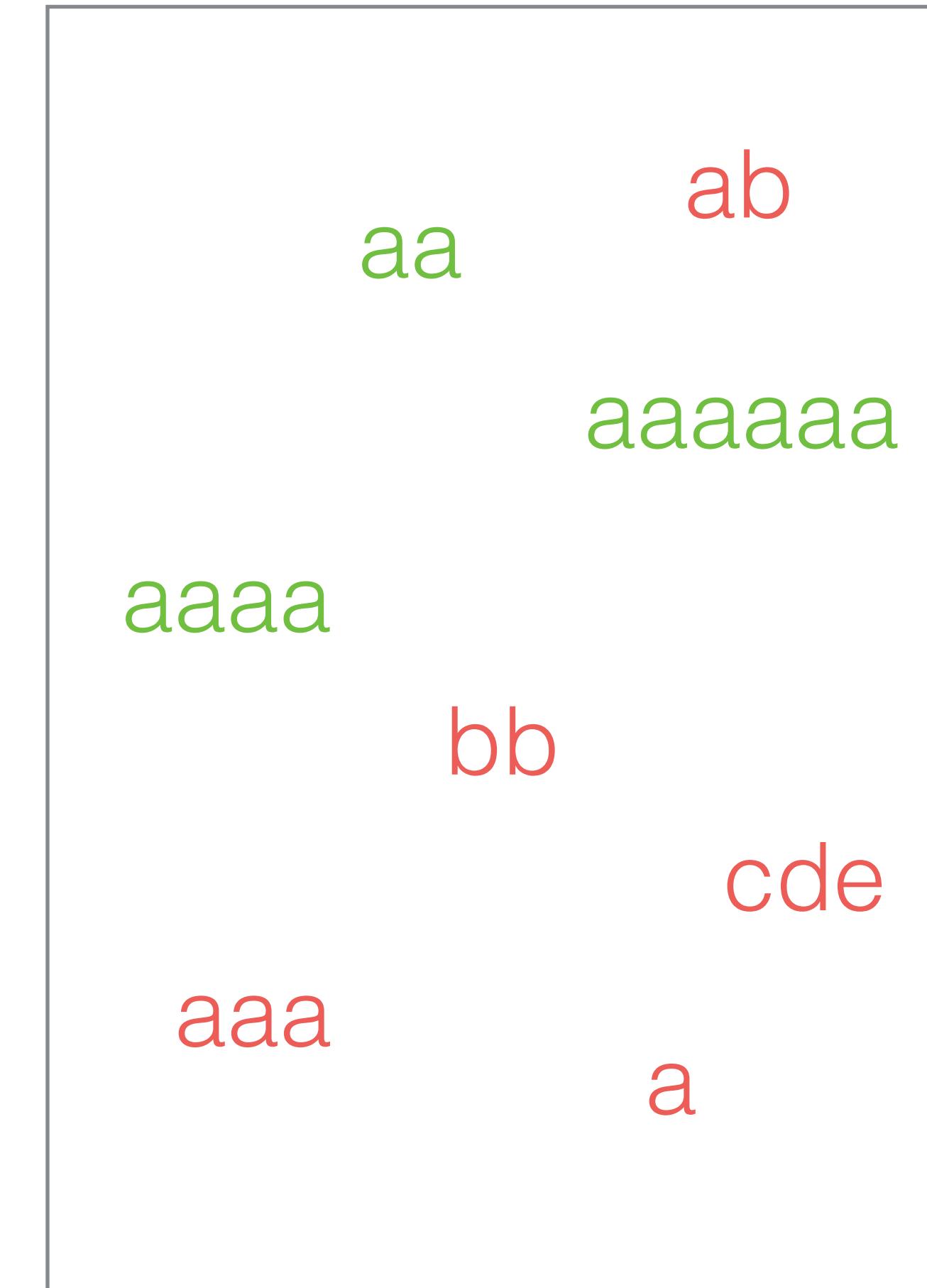
A



# Formal Grammar

```
S → A  
A → aa  
A → Aaa
```

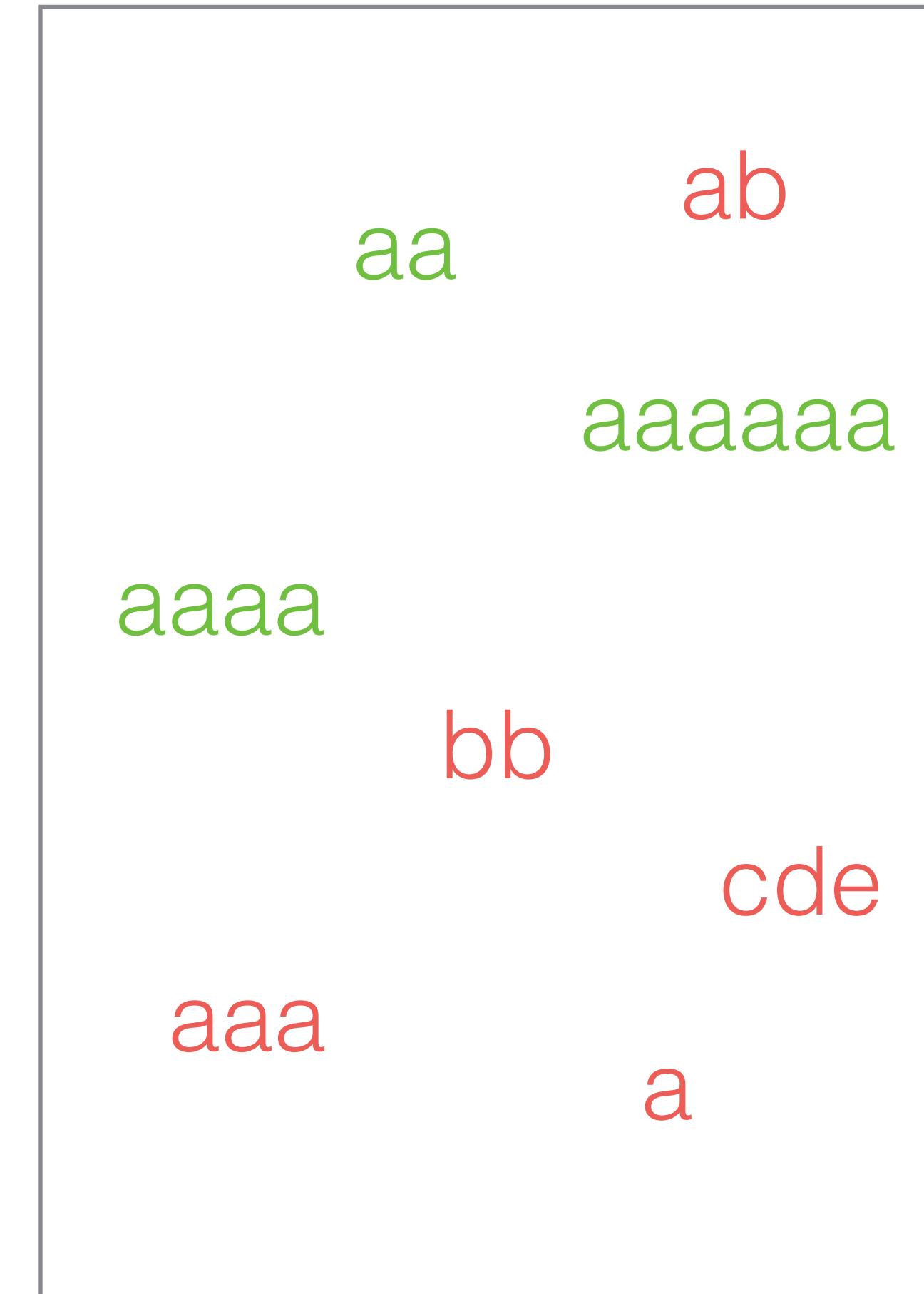
A



# Formal Grammar

```
S → A  
A → aa  
A → Aaa
```

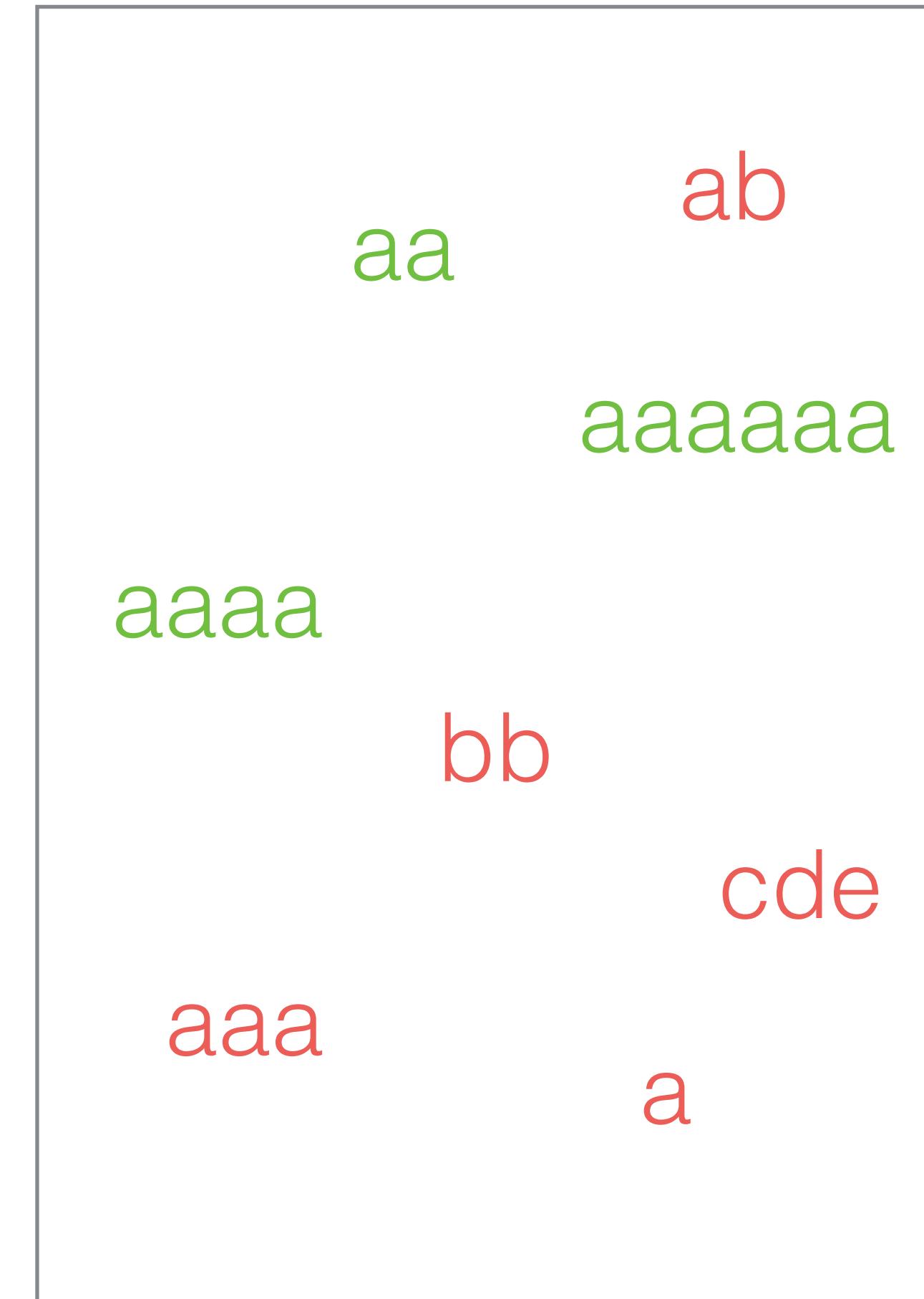
Aaa



# Formal Grammar

```
S → A  
A → aa  
A → Aaa
```

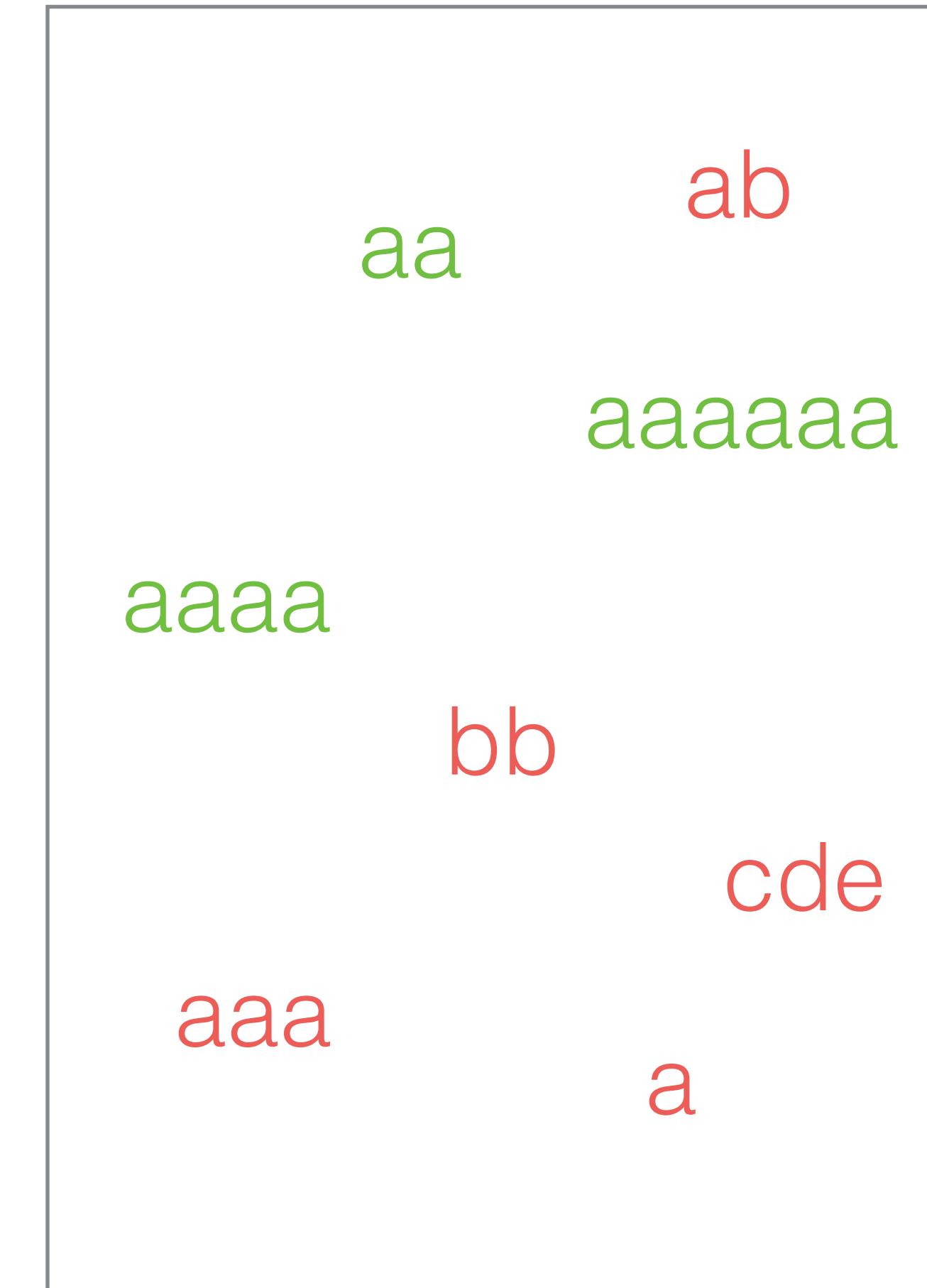
Aaa



# Formal Grammar

```
S → A  
A → aa  
A → Aaa
```

Aaa



# Formal Grammar

```
S → A  
A → aa  
A → Aaa
```

Aaaaa

```
aa  
aaaaaa  
aaaa  
bb  
cde  
aaa  
a
```

# Formal Grammar

```
S → A  
A → aa  
A → Aaa
```

Aaaaa



# Formal Grammar

```
S → A  
A → aa  
A → Aaa
```

aaaaaa

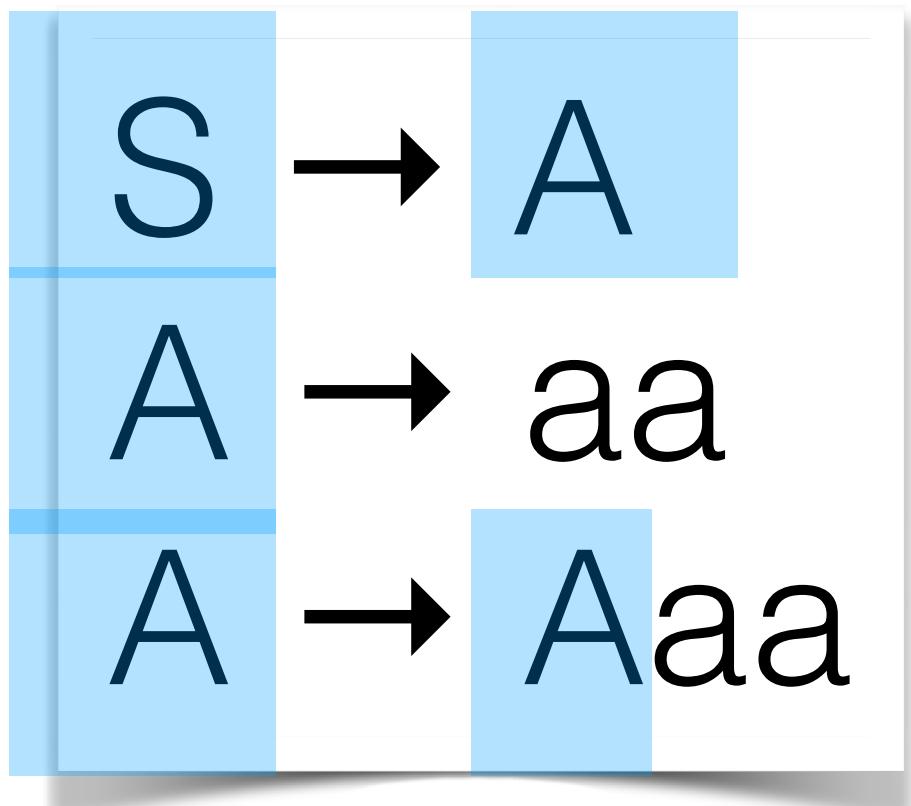


# Formal Grammar

## Definition

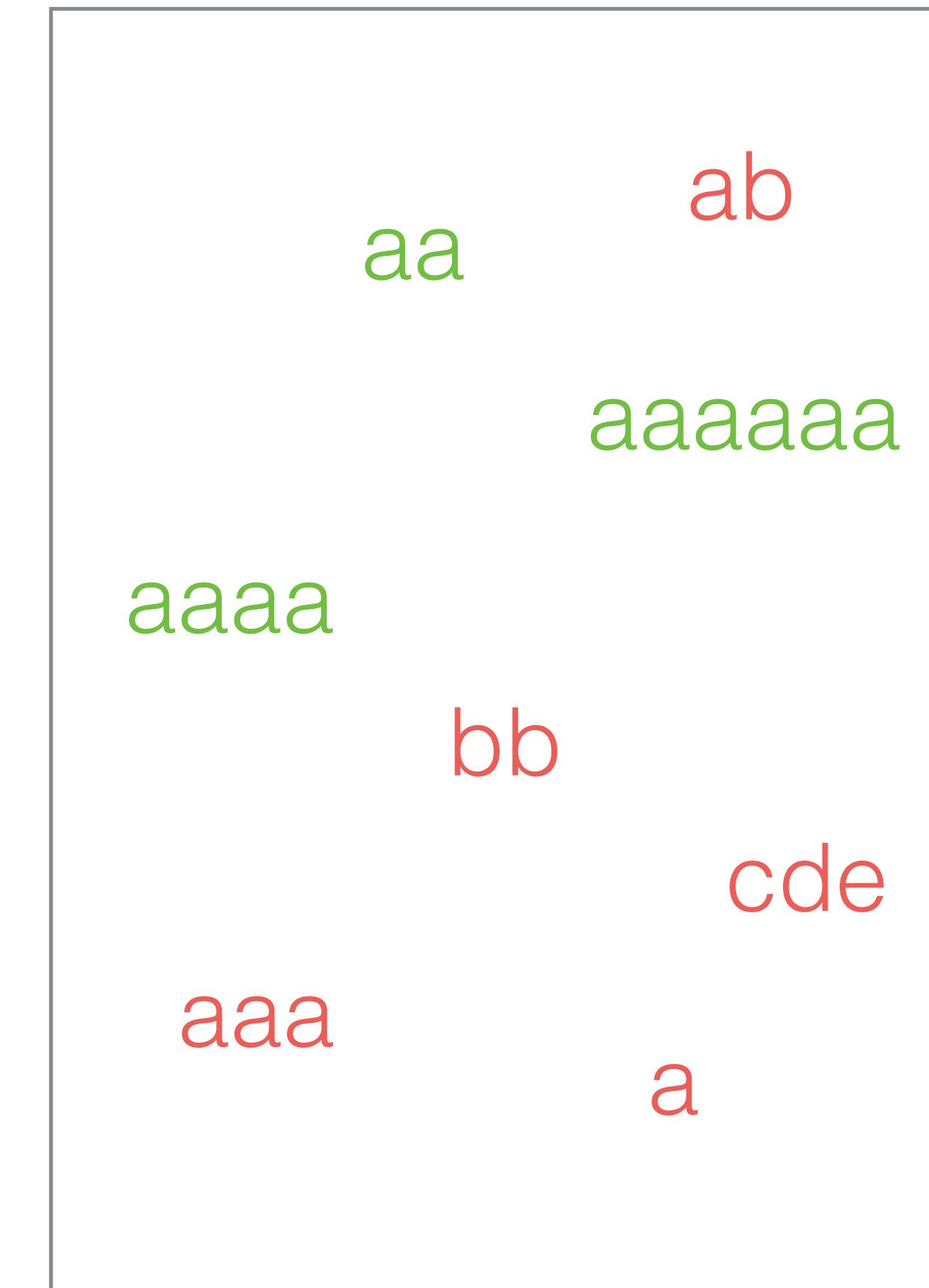
- N: Set of **non-terminal symbols** (can think of these as **variables**)
- $\Sigma$ : Set of **terminal symbols** (disjoint from N)
- R: Set of **production rules** each of the form  $A \rightarrow \beta$
- S: A designated **start symbol** (member of N)

# Formal Grammar

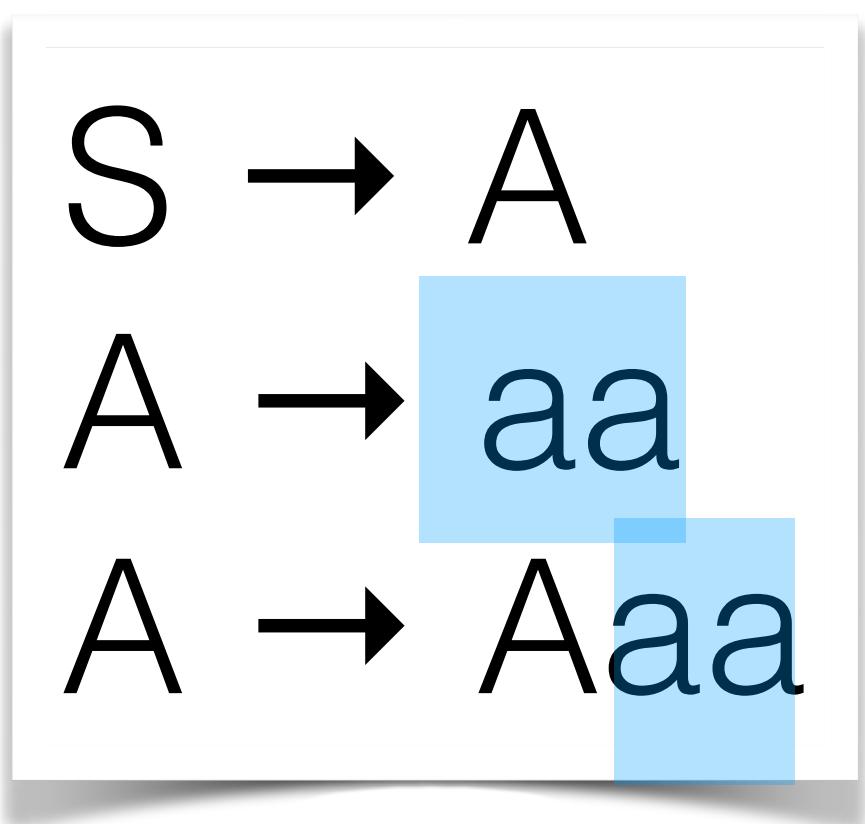


Non-Terminal Symbols

$$N = \{S, A\}$$

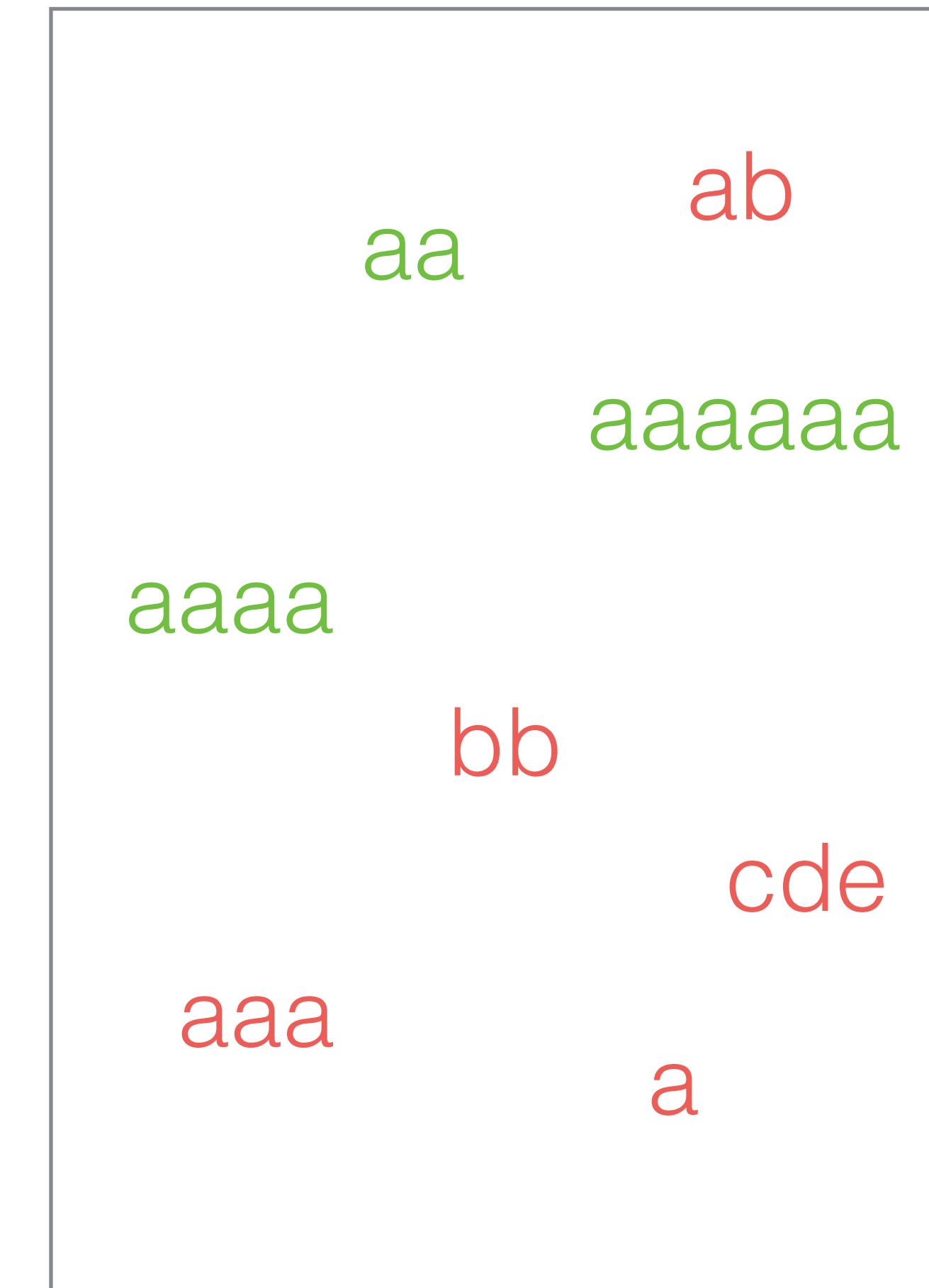


# Formal Grammar



Terminal Symbols

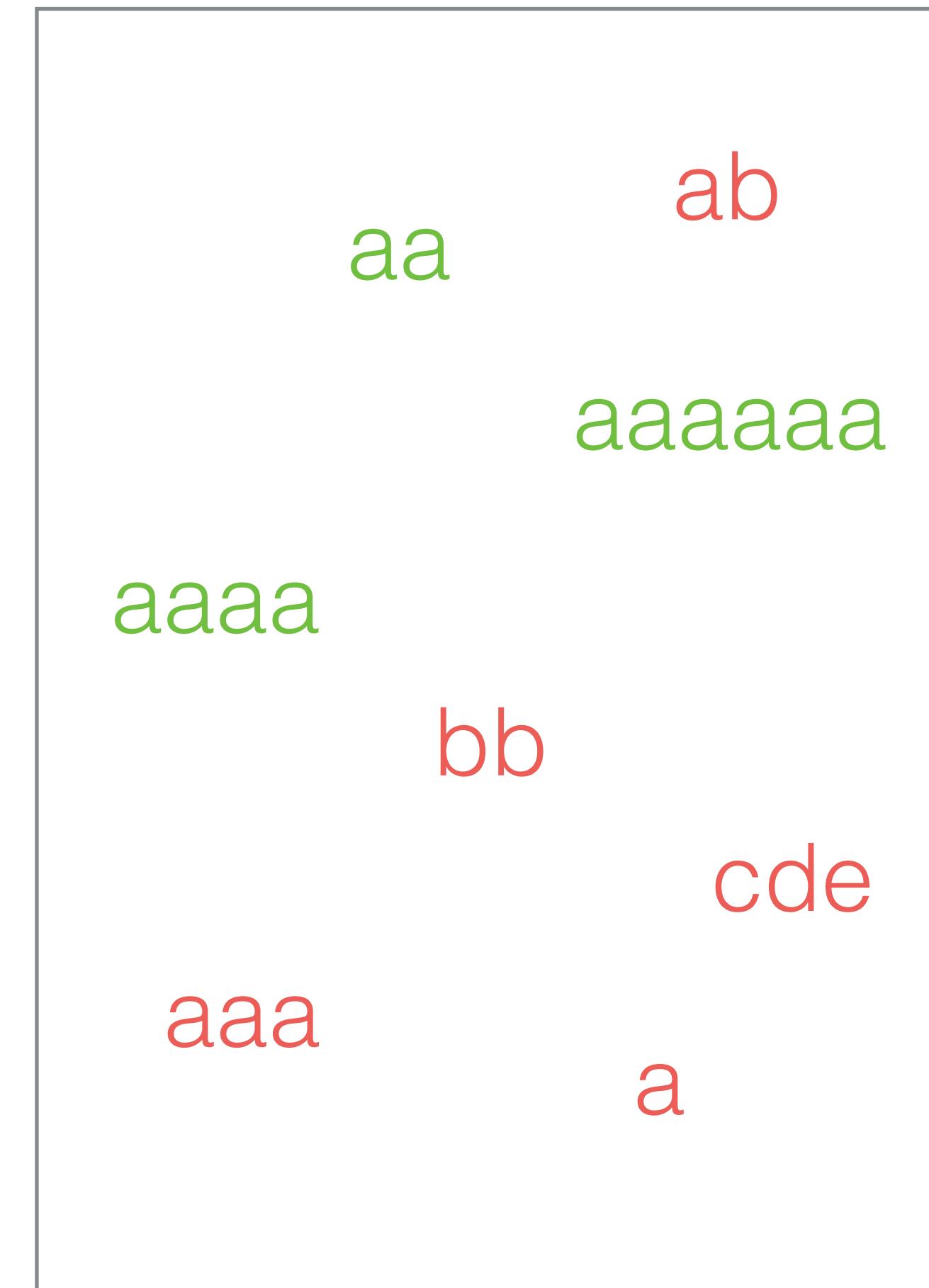
$$\Sigma = \{a\}$$



# Formal Grammar

```
S → A  
A → aa  
A → Aaa
```

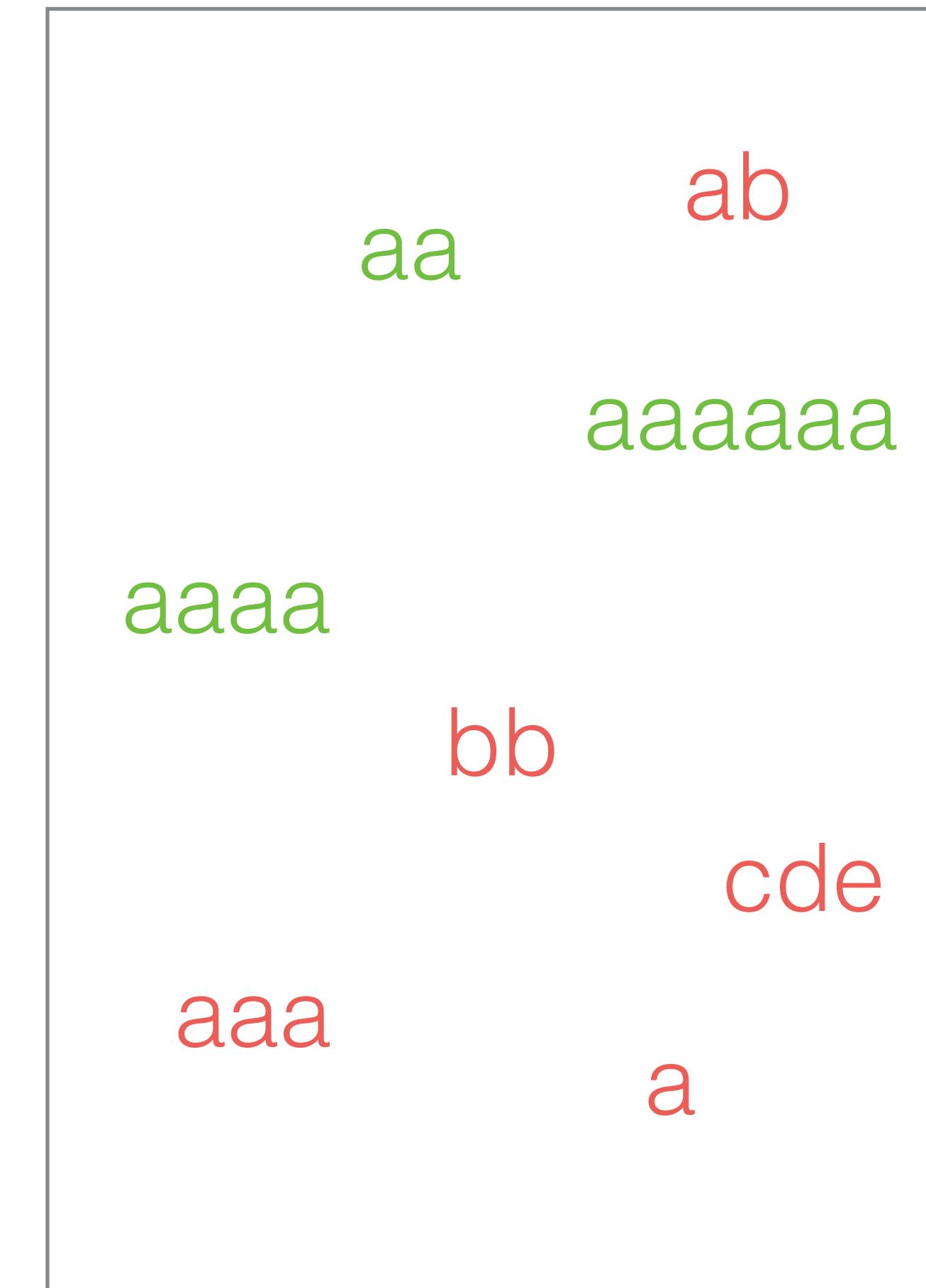
Production Rules



# Formal Grammar

```
S → A  
A → aa  
A → Aaa
```

Start Symbol



# Formal Grammar and Automata

- Grammars and Automata are two ways of modeling the same thing
- Grammars are used to ***generate*** languages
- Automata are used to ***recognize*** languages

# Finite State Automata

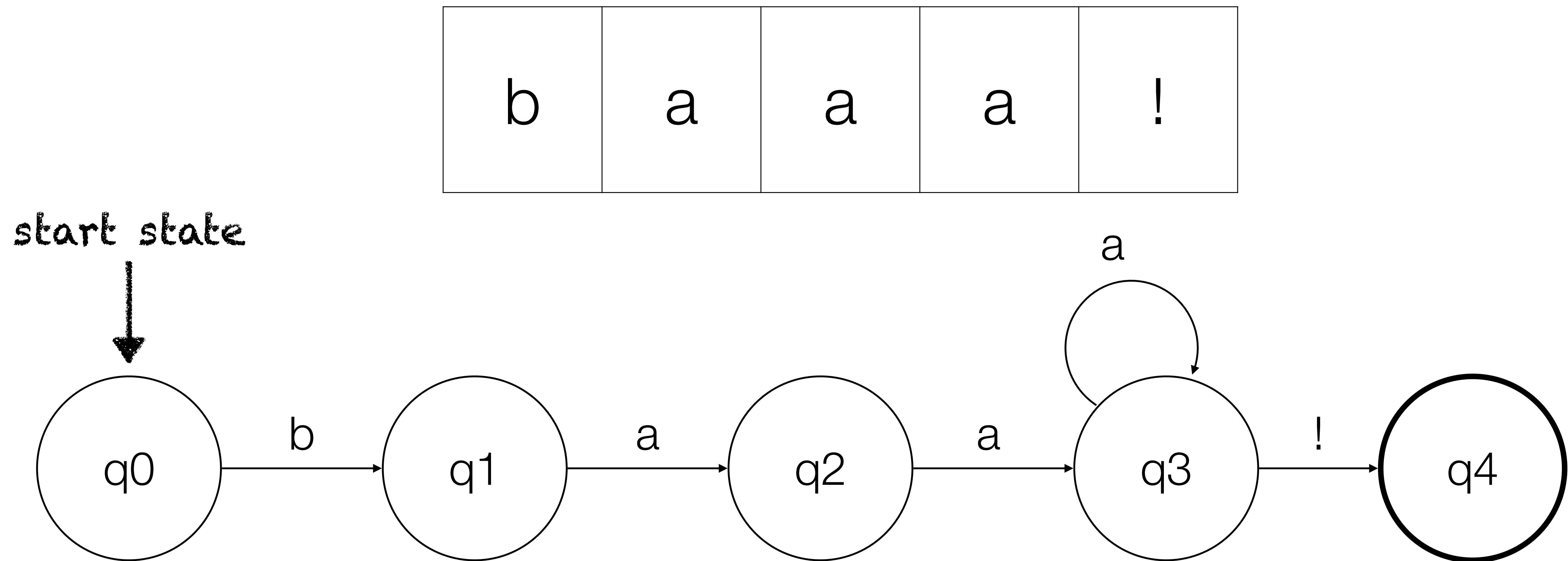
## Simple Example

“sheep” language: contains “b” followed by at least two “a”s followed by “!”

baa!	ba!
baaa!	baba!
baaaaaaaaaa!	abaaa!

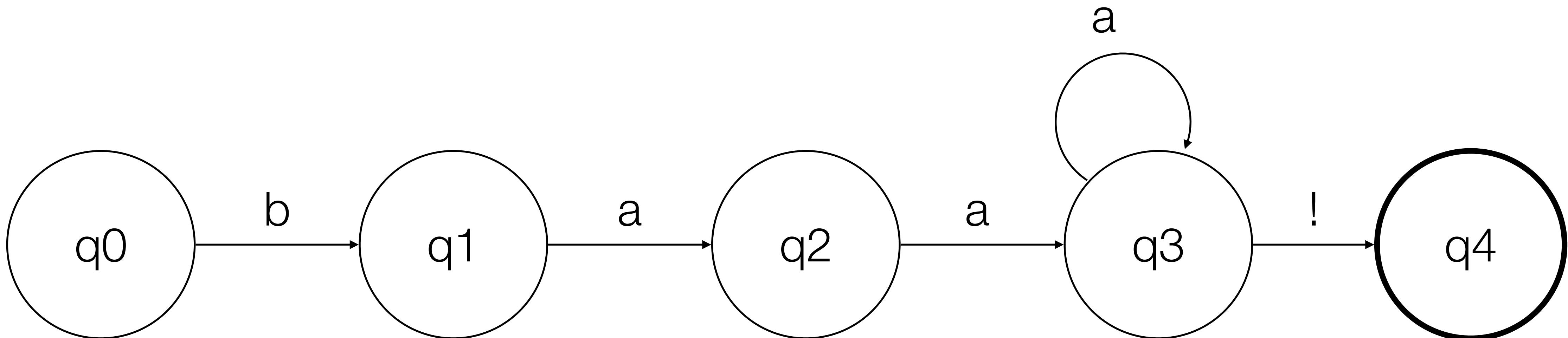
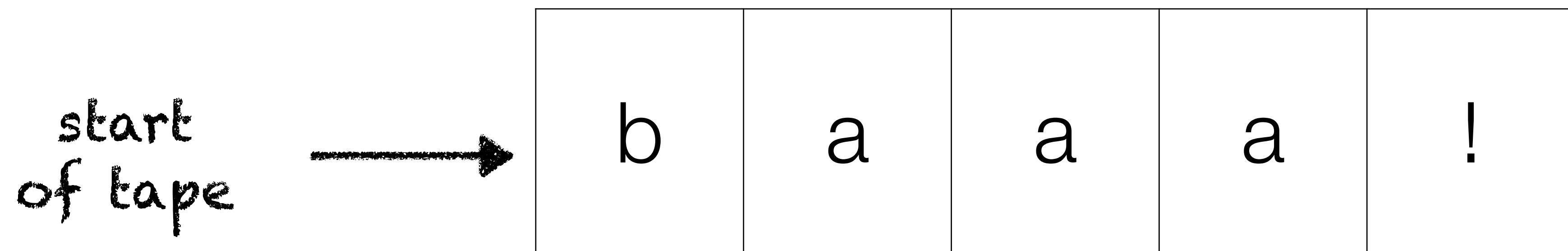
# Finite State Automata

## Simple Example



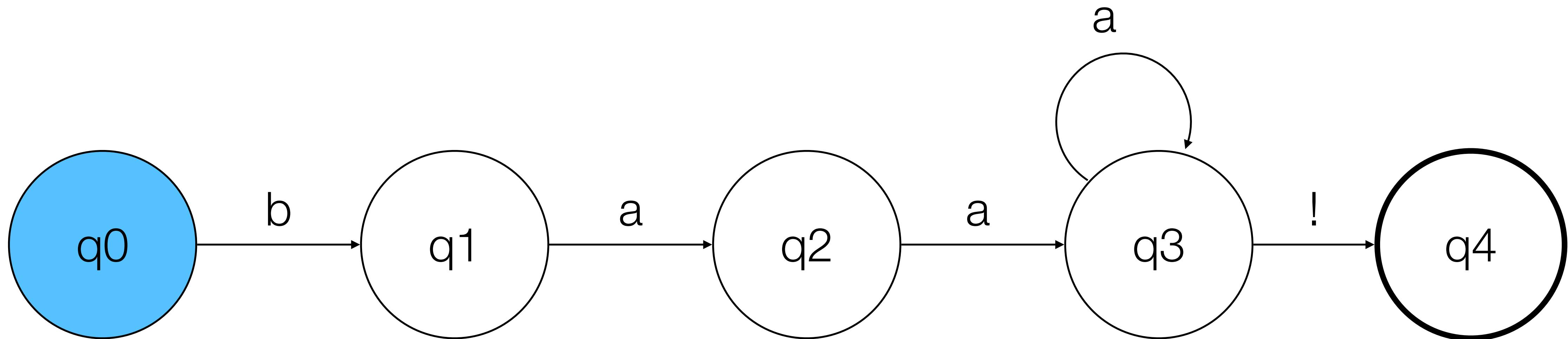
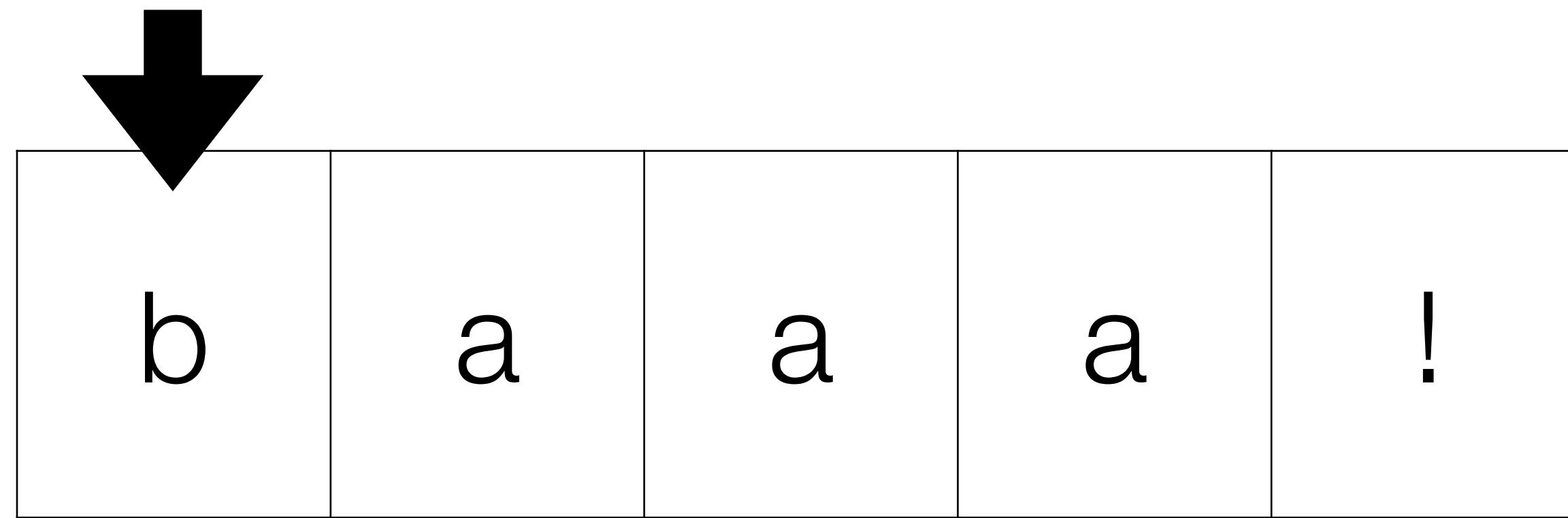
# Finite State Automata

## Simple Example



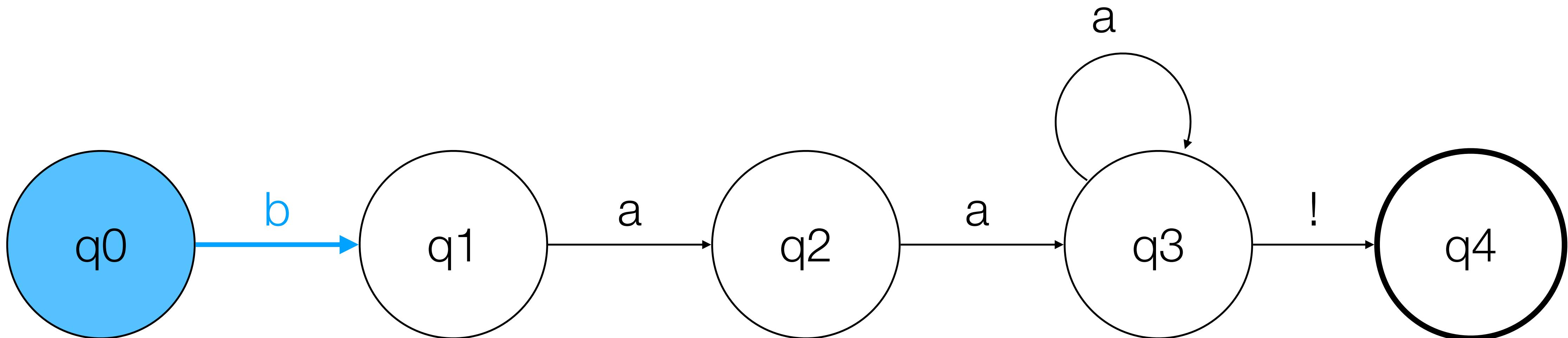
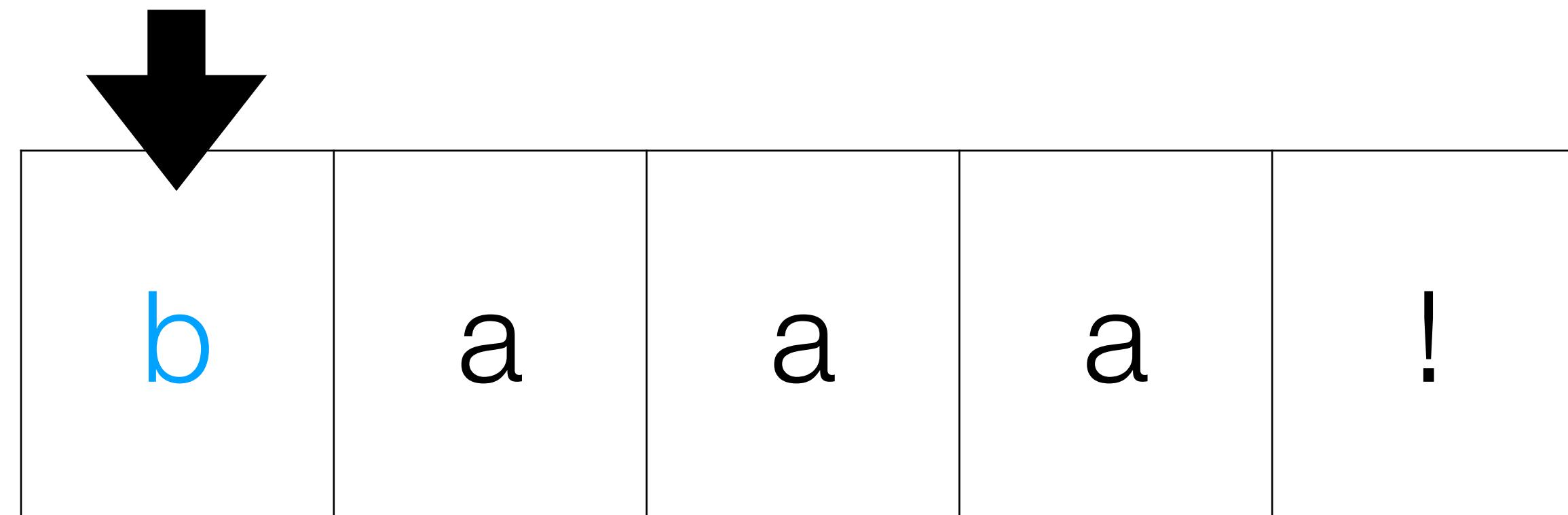
# Finite State Automata

## Simple Example



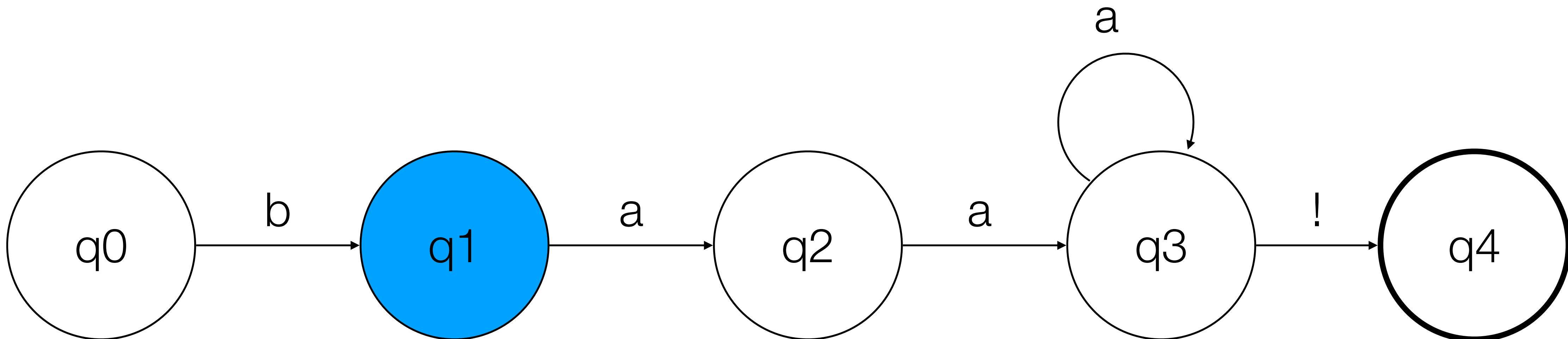
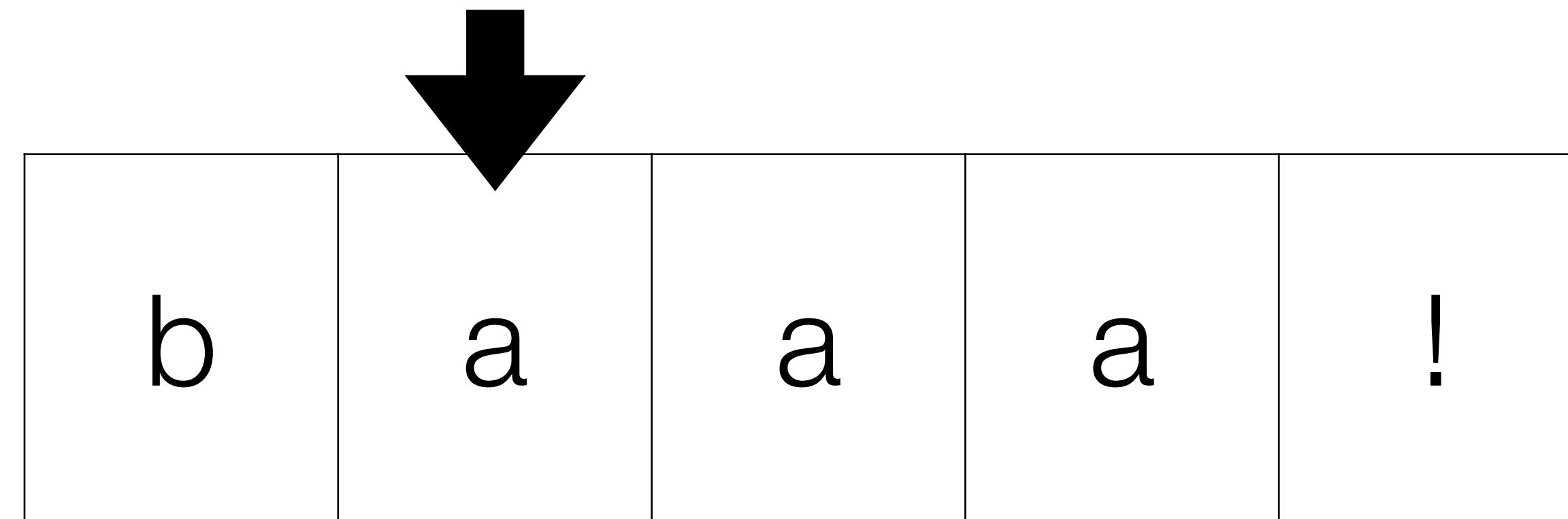
# Finite State Automata

## Simple Example



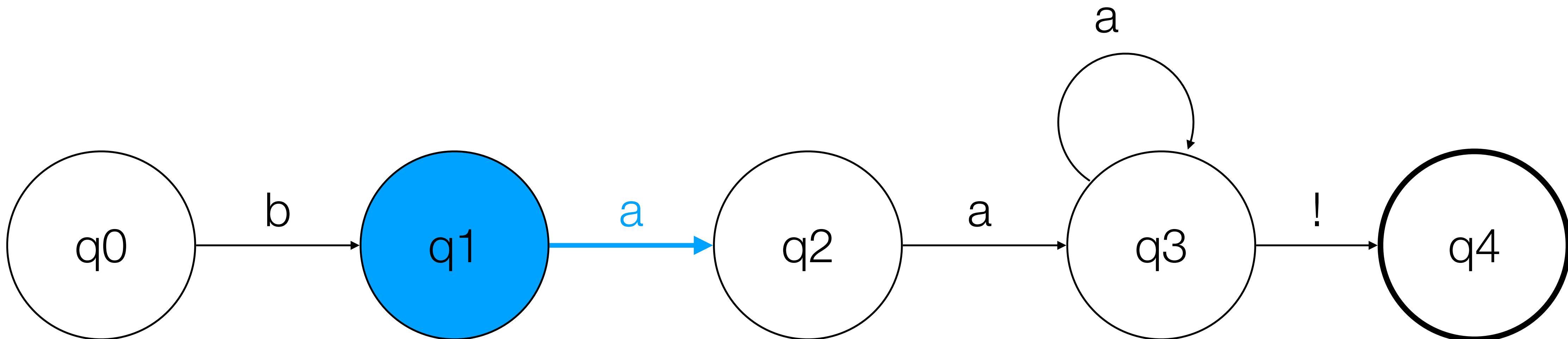
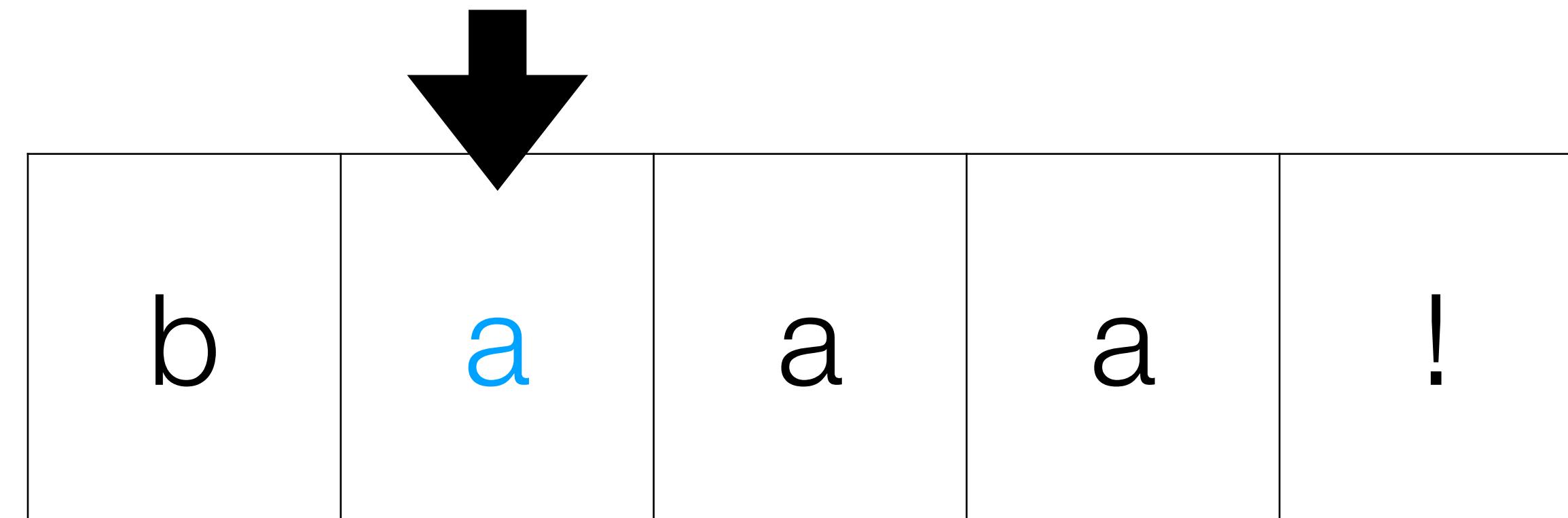
# Finite State Automata

## Simple Example



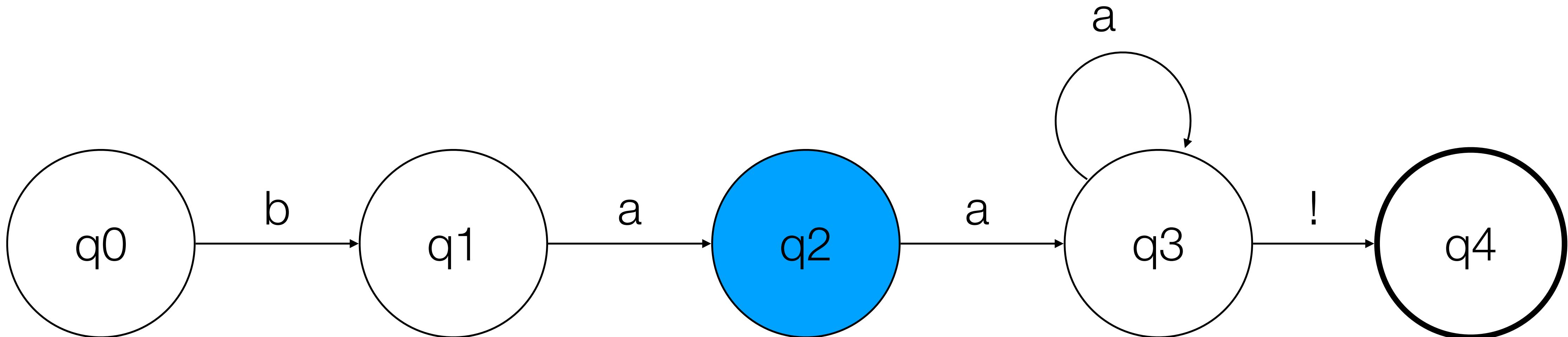
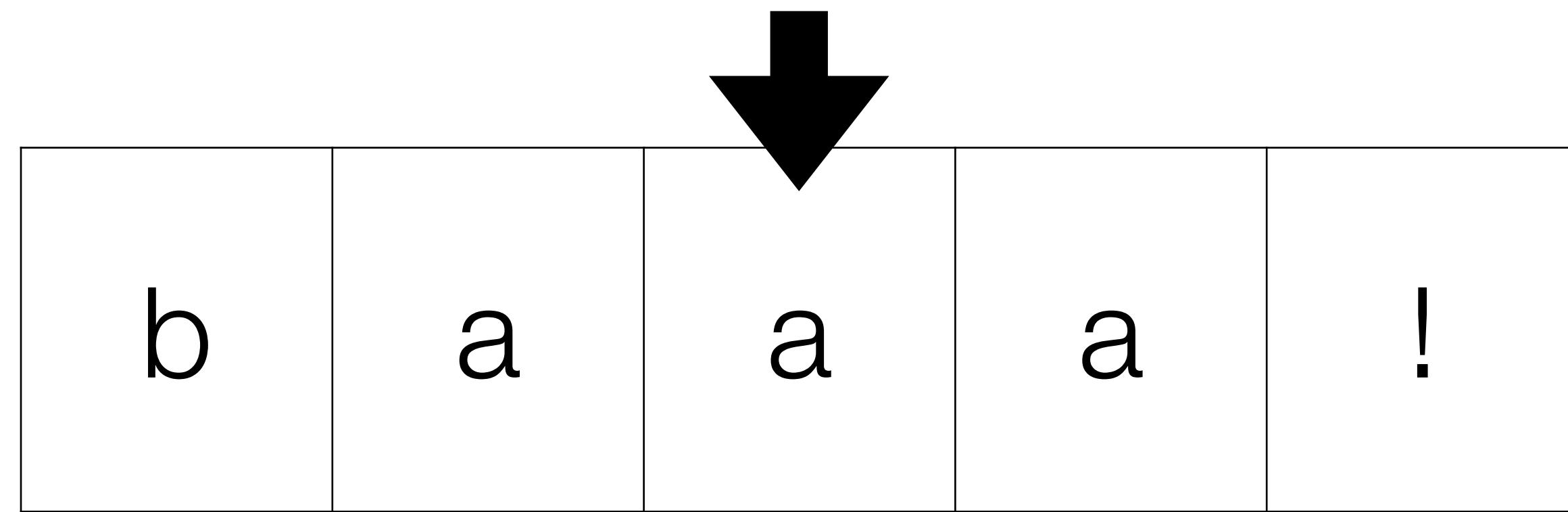
# Finite State Automata

## Simple Example



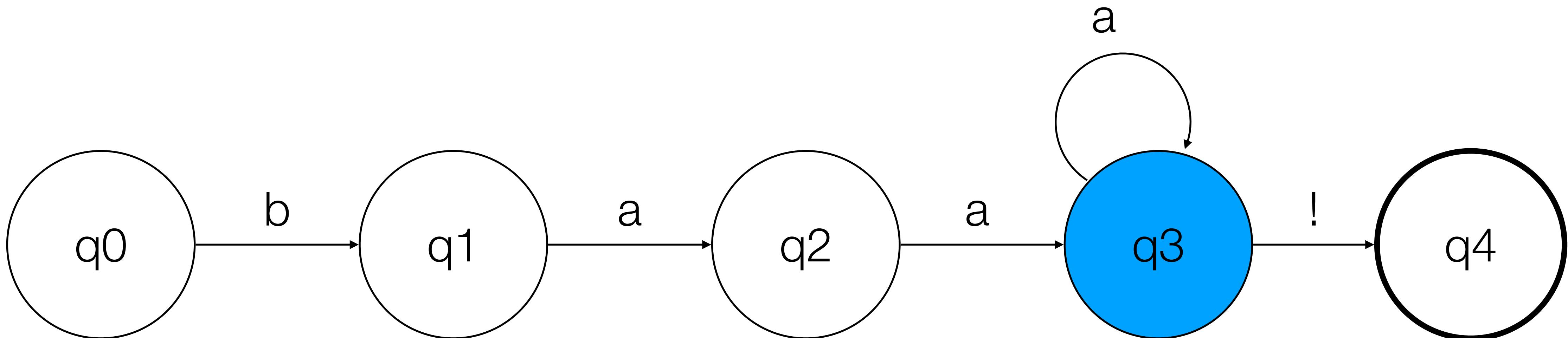
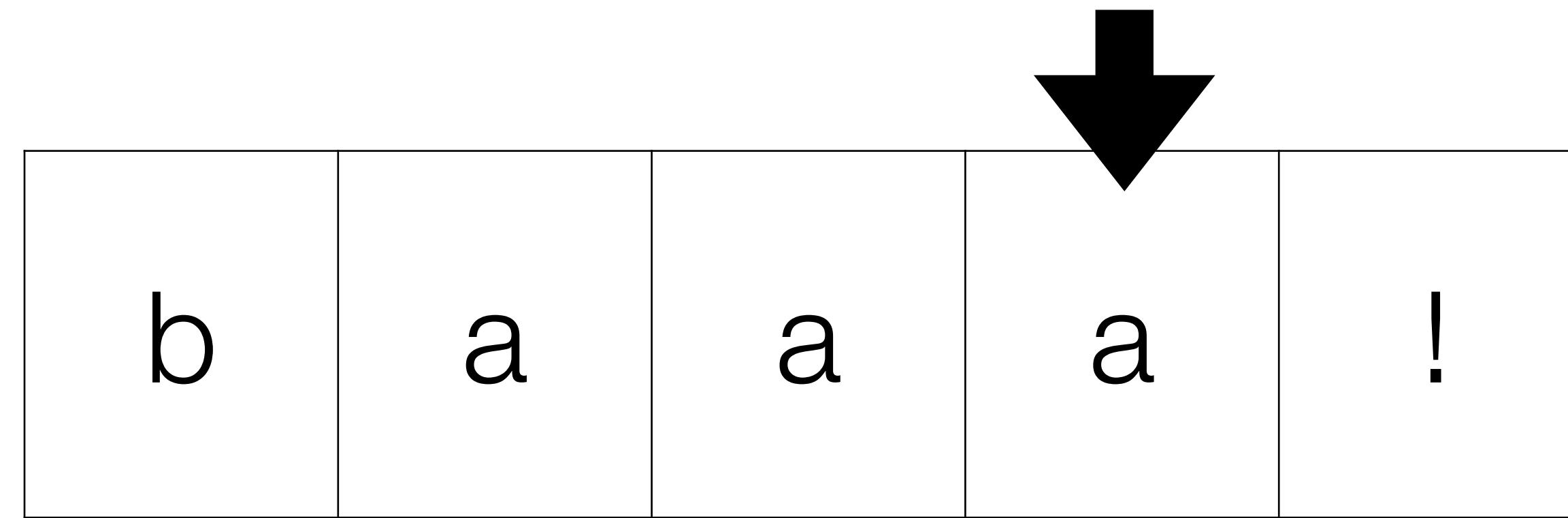
# Finite State Automata

## Simple Example



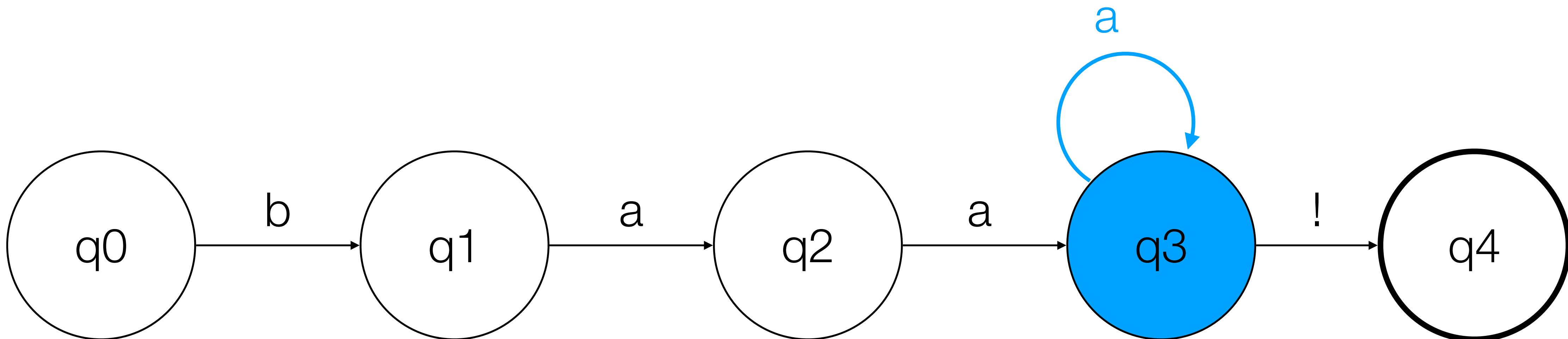
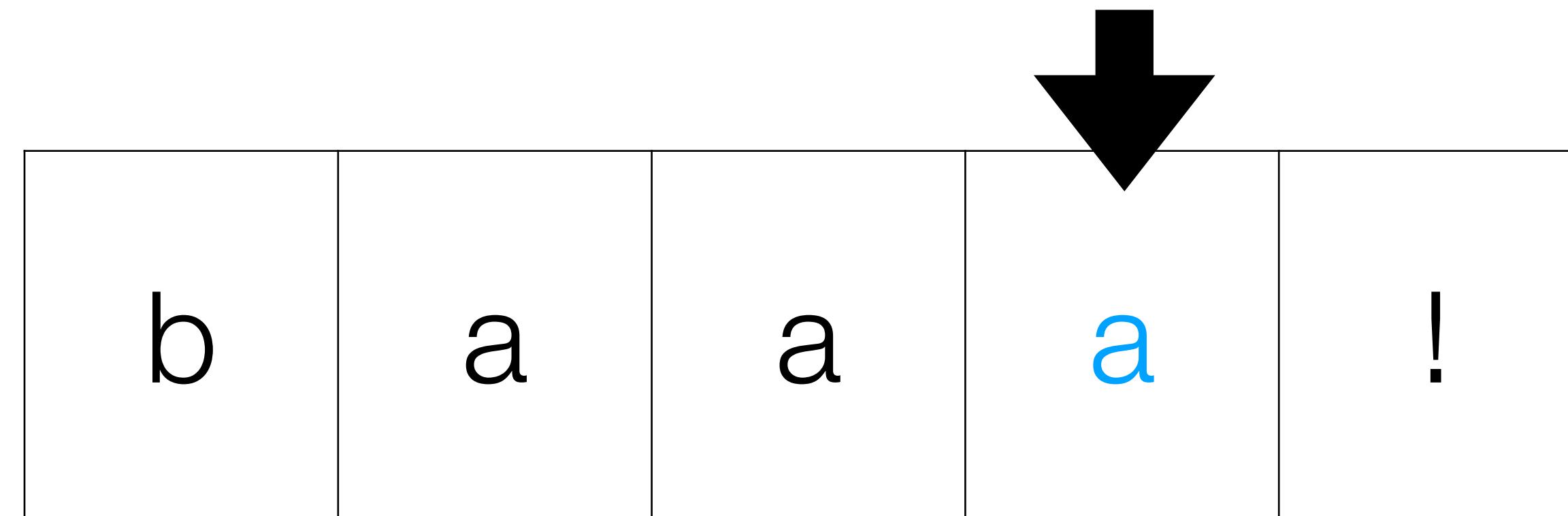
# Finite State Automata

## Simple Example



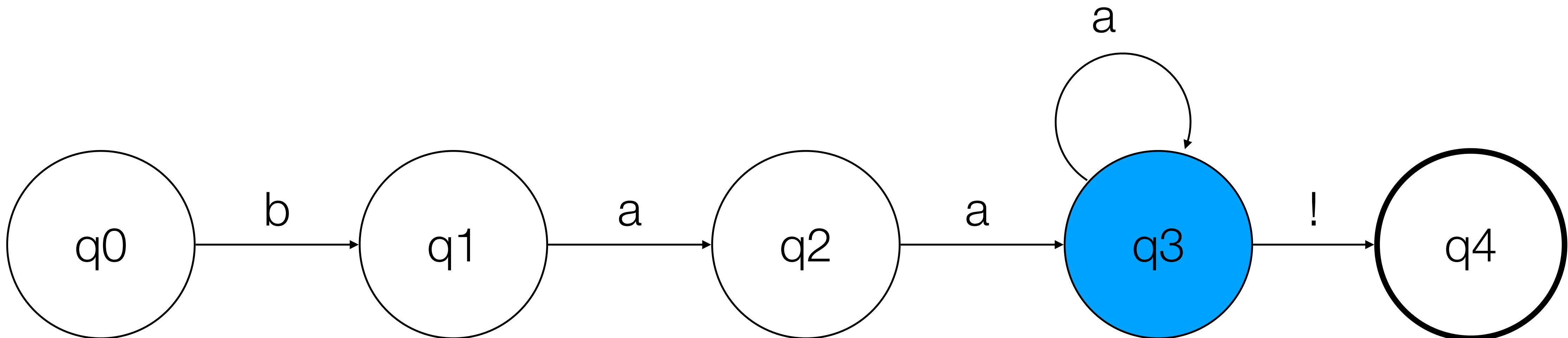
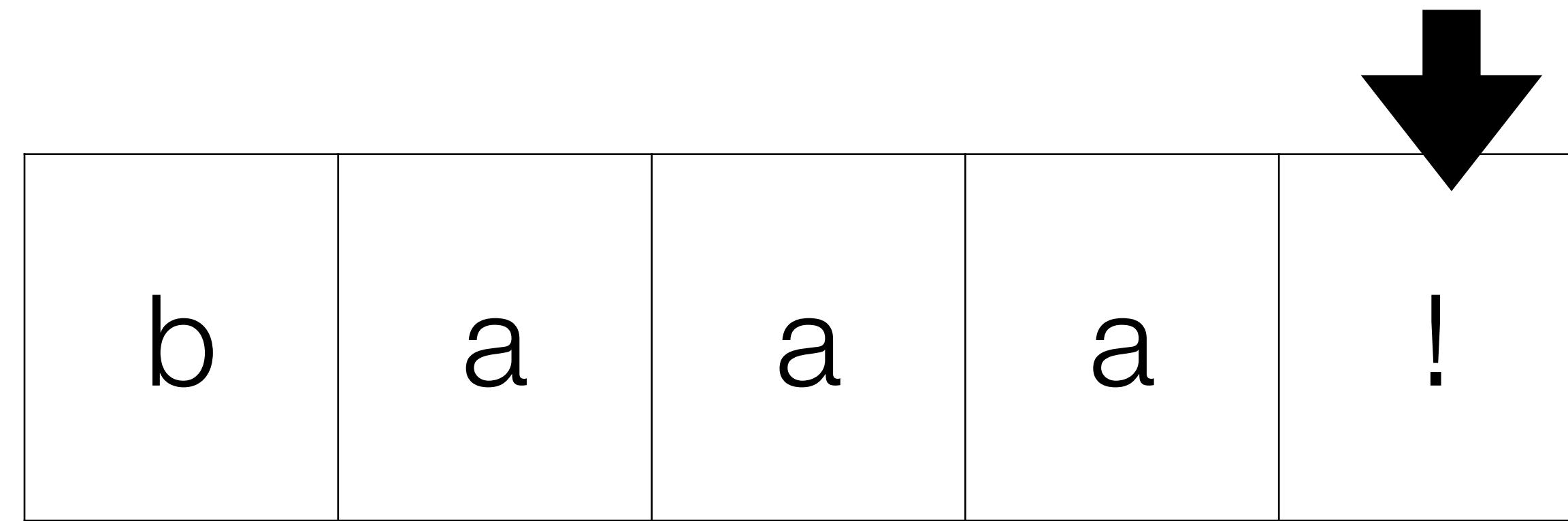
# Finite State Automata

## Simple Example



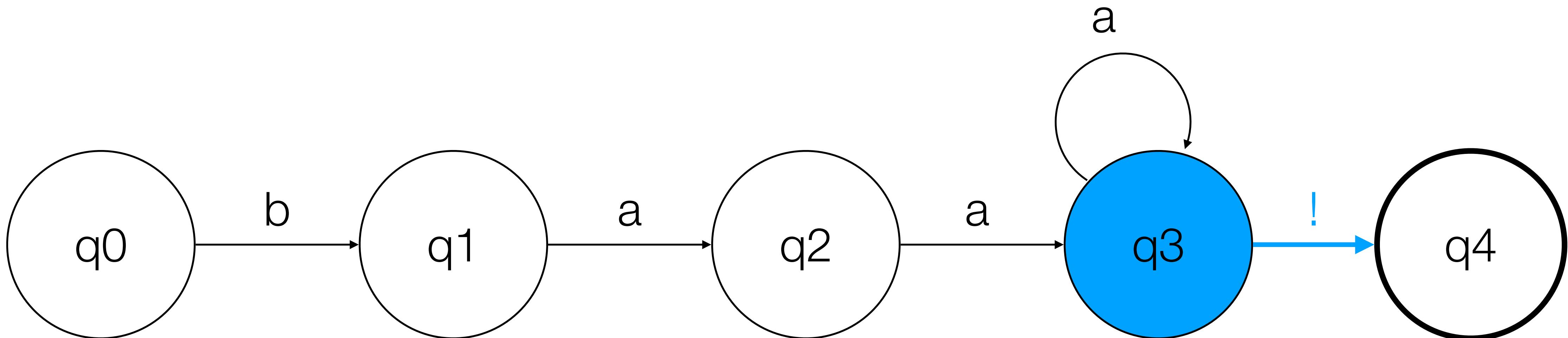
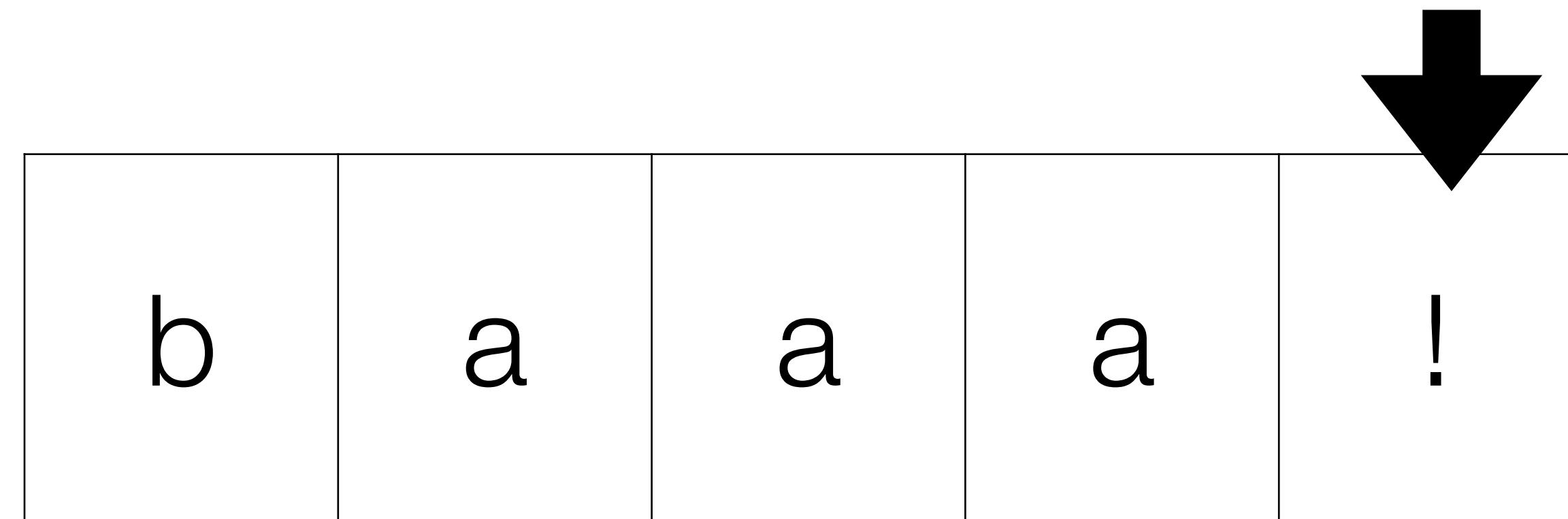
# Finite State Automata

## Simple Example



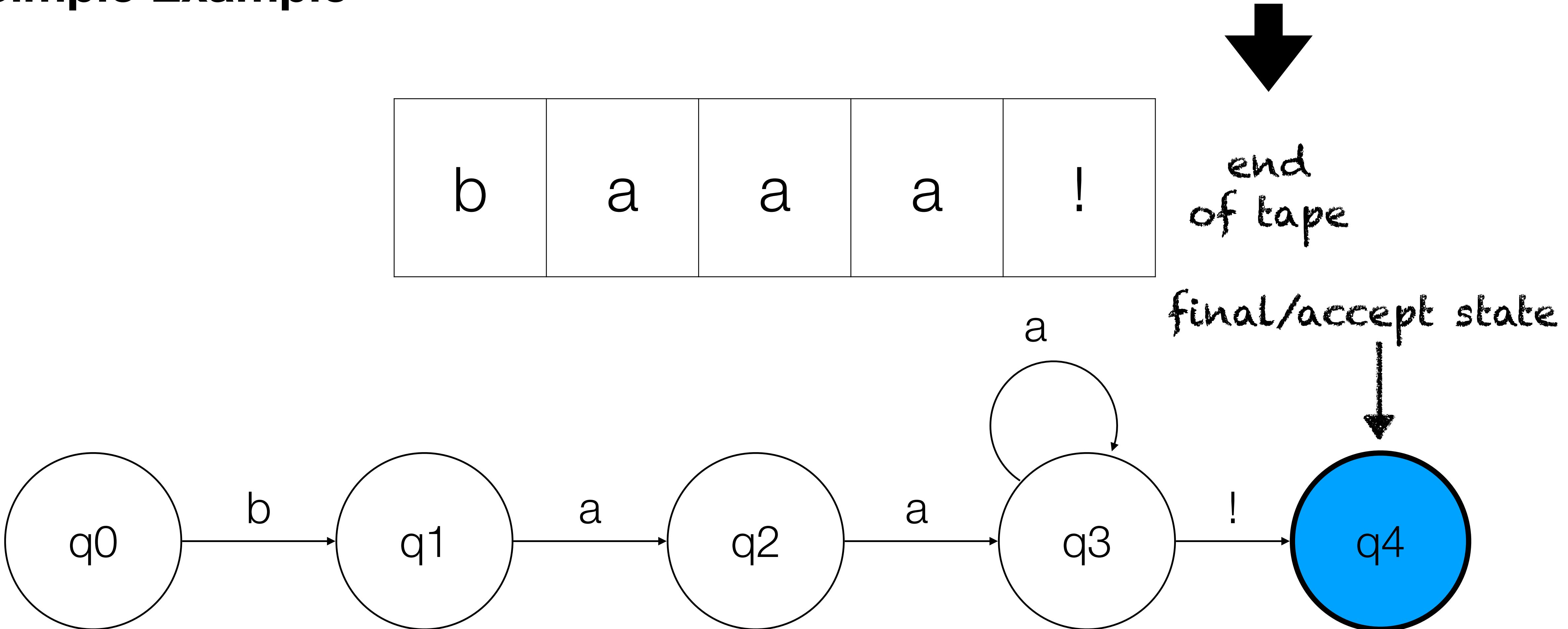
# Finite State Automata

## Simple Example



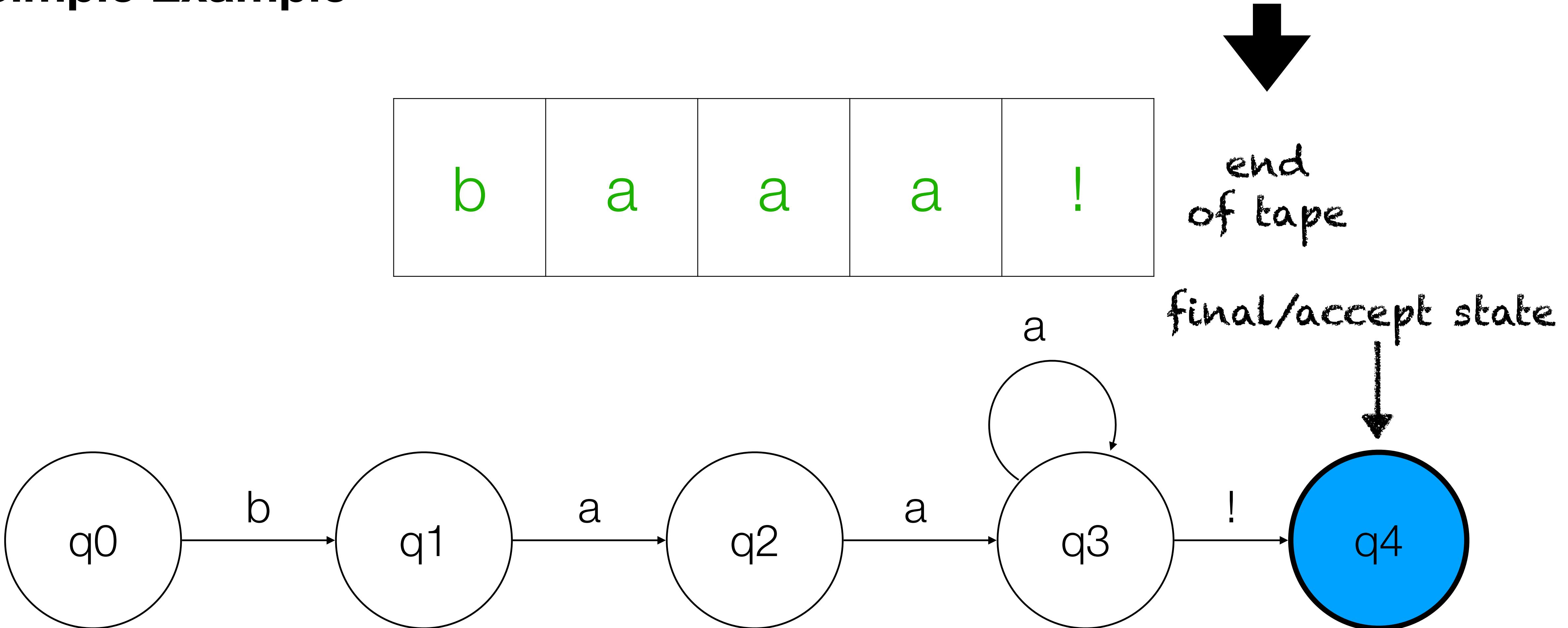
# Finite State Automata

## Simple Example



# Finite State Automata

## Simple Example



# Formal Grammar and Automata

- Regular language:
  - Grammar has the form:  $A \rightarrow Ba$  (i.e., one NT symbol, always at the beginning or end)
  - Can be recognized by an finite-state automaton

# Formal Grammar and Automata

$S \rightarrow \text{Nom sleeps}$

$\text{Nom} \rightarrow \text{Nom by the tree}$

$\text{Nom} \rightarrow \text{DT NN}$

$\text{DT} \rightarrow \{\text{the, a}\}$

$\text{NN} \rightarrow \{\text{cat, dog, bird, ...}\}$

# Formal Grammar and Automata

$S \rightarrow \text{Nom sleeps}$

$\text{Nom} \rightarrow \text{Nom by the tree}$

$\text{Nom} \rightarrow \text{DT cat}$

$\text{Nom} \rightarrow \text{DT dog}$

$\text{Nom} \rightarrow \text{DT bird}$

$\text{DT} \rightarrow \{\text{the, a}\}$

# Formal Grammar and Automata

$S \rightarrow \text{Nom sleeps}$

$\text{Nom} \rightarrow \text{Nom by the tree}$

$\text{Nom} \rightarrow \text{DT NN}$

$\text{DT} \rightarrow \{\text{the, a}\}$

$\text{NN} \rightarrow \{\text{cat, dog, bird, ...}\}$

S

# Formal Grammar and Automata

$S \rightarrow \text{Nom sleeps}$

$\text{Nom} \rightarrow \text{Nom by the tree}$

$\text{Nom} \rightarrow \text{DT NN}$

$\text{DT} \rightarrow \{\text{the, a}\}$

$\text{NN} \rightarrow \{\text{cat, dog, bird, ...}\}$

Nom sleeps

# Formal Grammar and Automata

$S \rightarrow \text{Nom sleeps}$

$\text{Nom} \rightarrow \text{Nom by the tree}$

$\text{Nom} \rightarrow \text{DT NN}$

$\text{DT} \rightarrow \{\text{the, a}\}$

$\text{NN} \rightarrow \{\text{cat, dog, bird, ...}\}$

Nom by the tree sleeps

# Formal Grammar and Automata

$S \rightarrow \text{Nom sleeps}$

$\text{Nom} \rightarrow \text{Nom by the tree}$

$\text{Nom} \rightarrow \text{DT NN}$

$\text{DT} \rightarrow \{\text{the, a}\}$

$\text{NN} \rightarrow \{\text{cat, dog, bird, ...}\}$

Nom by the tree by the tree sleeps

# Formal Grammar and Automata

$S \rightarrow \text{Nom sleeps}$

$\text{Nom} \rightarrow \text{Nom by the tree}$

$\text{Nom} \rightarrow \text{DT NN}$

$\text{DT} \rightarrow \{\text{the, a}\}$

$\text{NN} \rightarrow \{\text{cat, dog, bird, ...}\}$

$\text{DT NN by the tree by the tree sleeps}$

# Formal Grammar and Automata

$S \rightarrow \text{Nom sleeps}$

$\text{Nom} \rightarrow \text{Nom by the tree}$

$\text{Nom} \rightarrow \text{DT NN}$

$\text{DT} \rightarrow \{\text{the, a}\}$

$\text{NN} \rightarrow \{\text{cat, dog, bird, ...}\}$

the cat by the tree by the tree sleeps



# Formal Grammar and Automata

the cat sleeps

the cat by the tree sleeps

the cat by the tree by the tree sleeps

...

# Formal Grammar and Automata

the **cat sleeps**

the **cats** by the tree **sleep**

the **cat** by the tree by the tree **sleeps**

...

# Formal Grammar and Automata

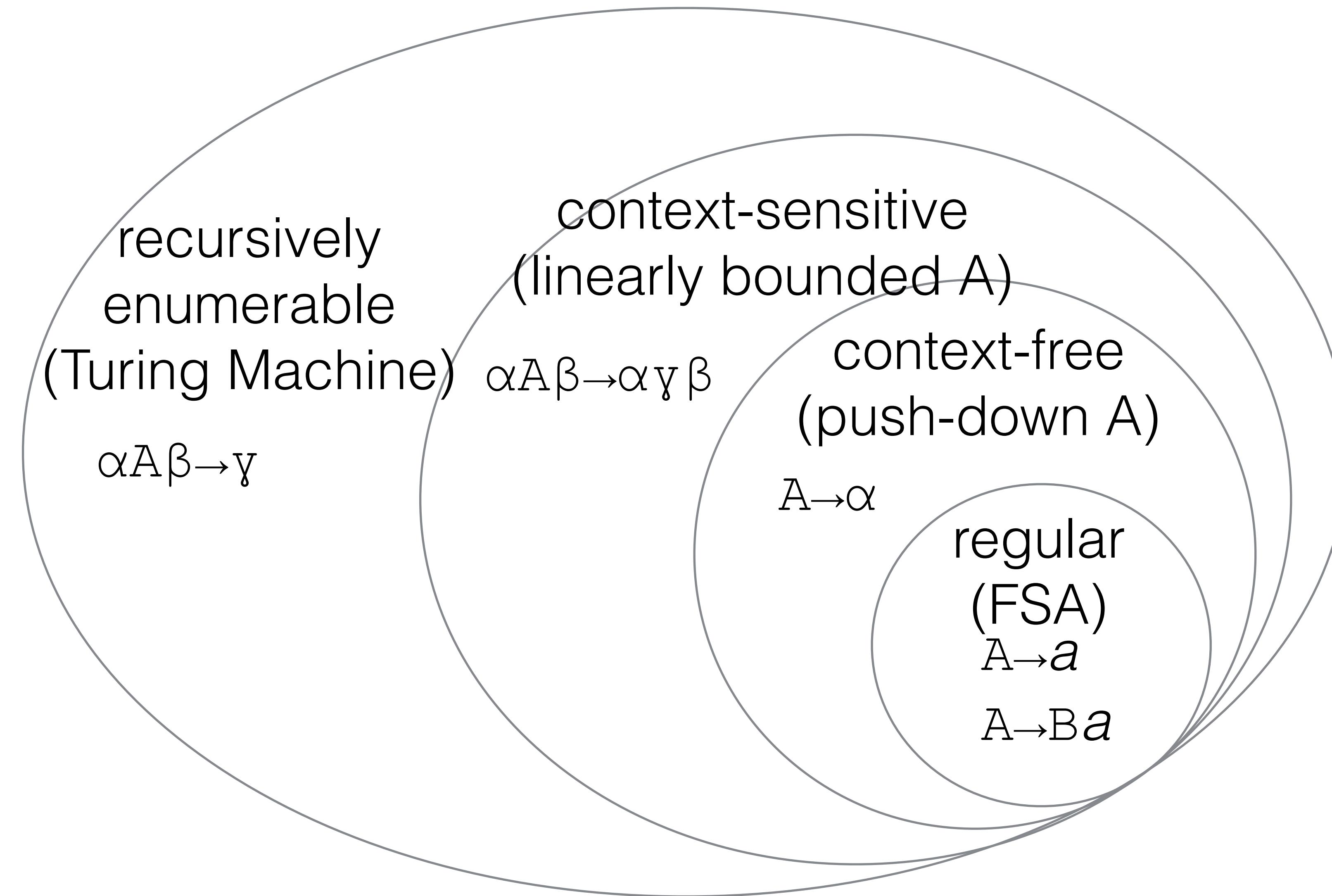
- Regular language: doesn't work when language requires recursion
- Context-Free Languages:
  - Grammars have form  $A \rightarrow a$  (where  $a$  can be any mix of terminals and non-terminals)
  - Language can be recognized by a pushdown automaton (an automaton that uses a stack to maintain memory)
  - “context free” because rule can be applied to the nonterminal regardless of its context

# Formal Grammar and Automata

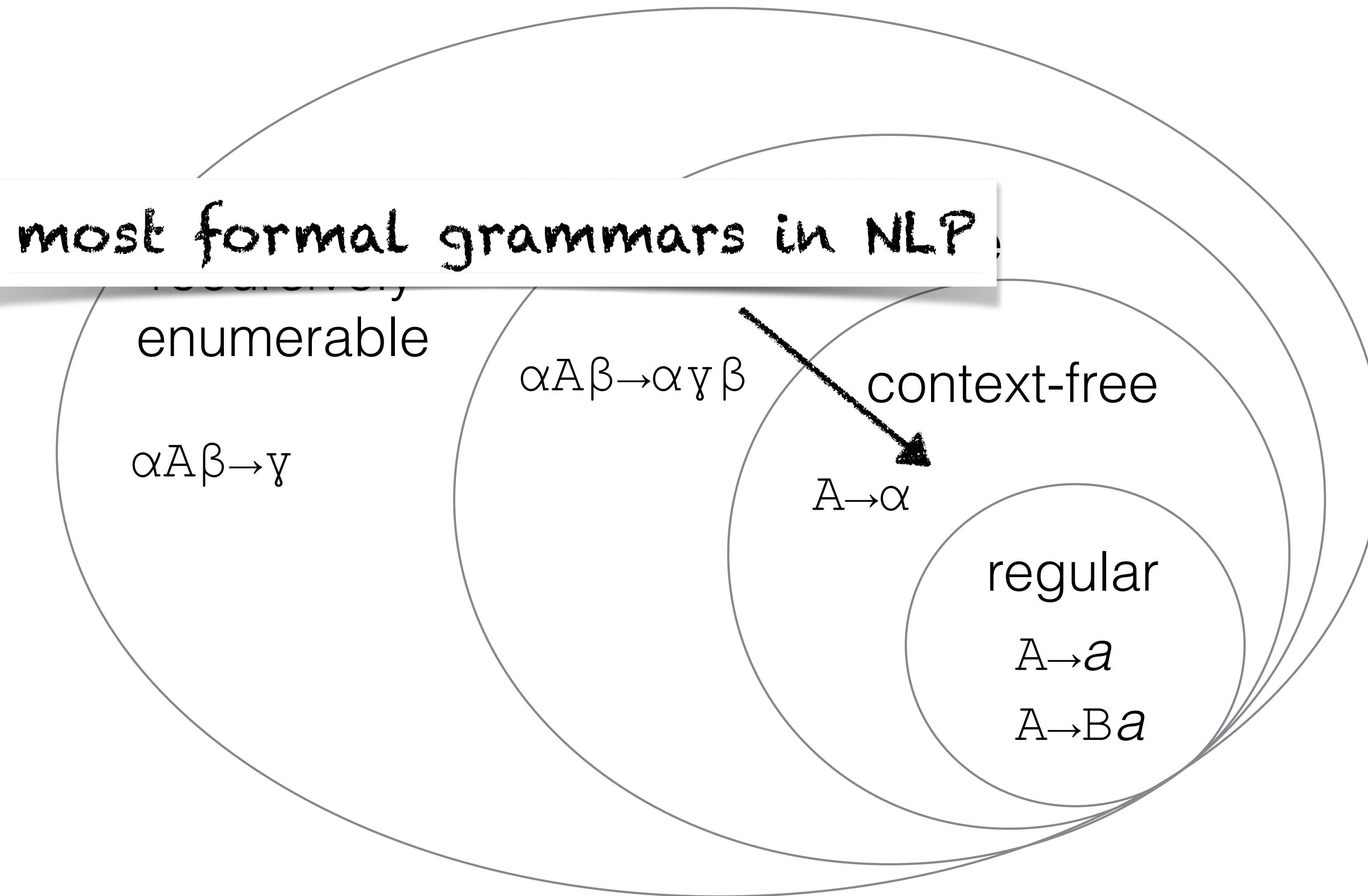
```
S → NPs PP VBs
S → NPp PP VBp
NPs → Noms PP
Noms → DT NNs
PP → PREP DT NN
```

...

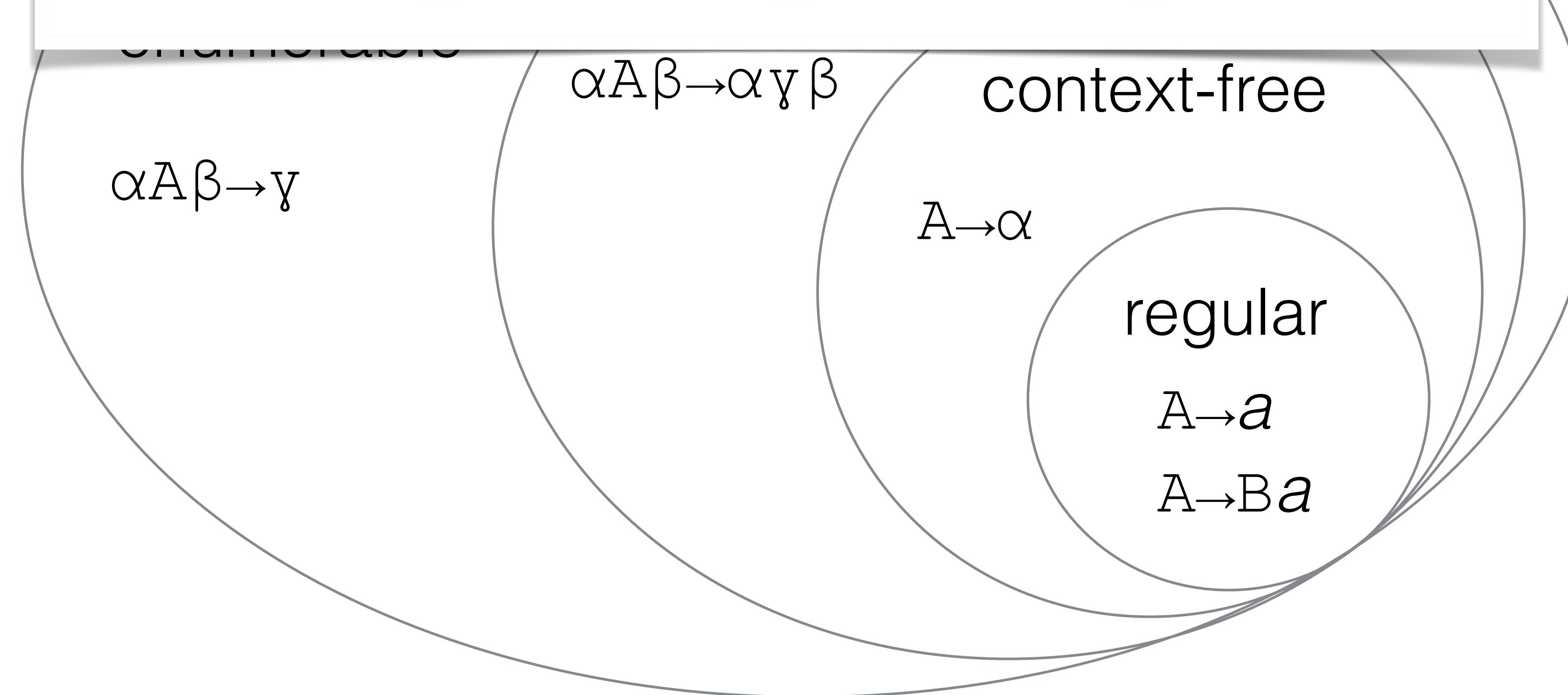
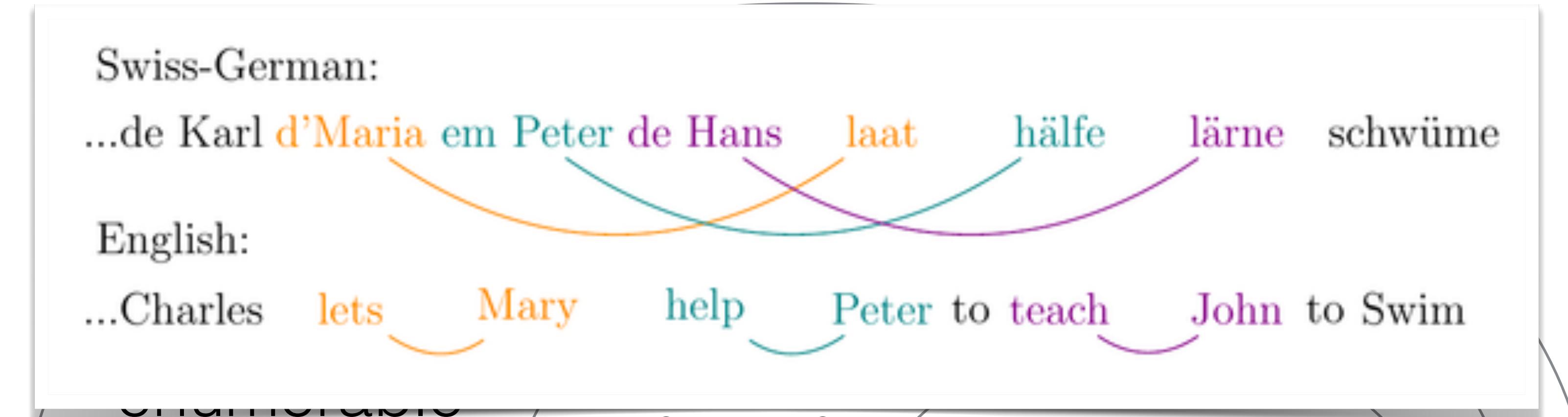
# Chomsky Hierarchy



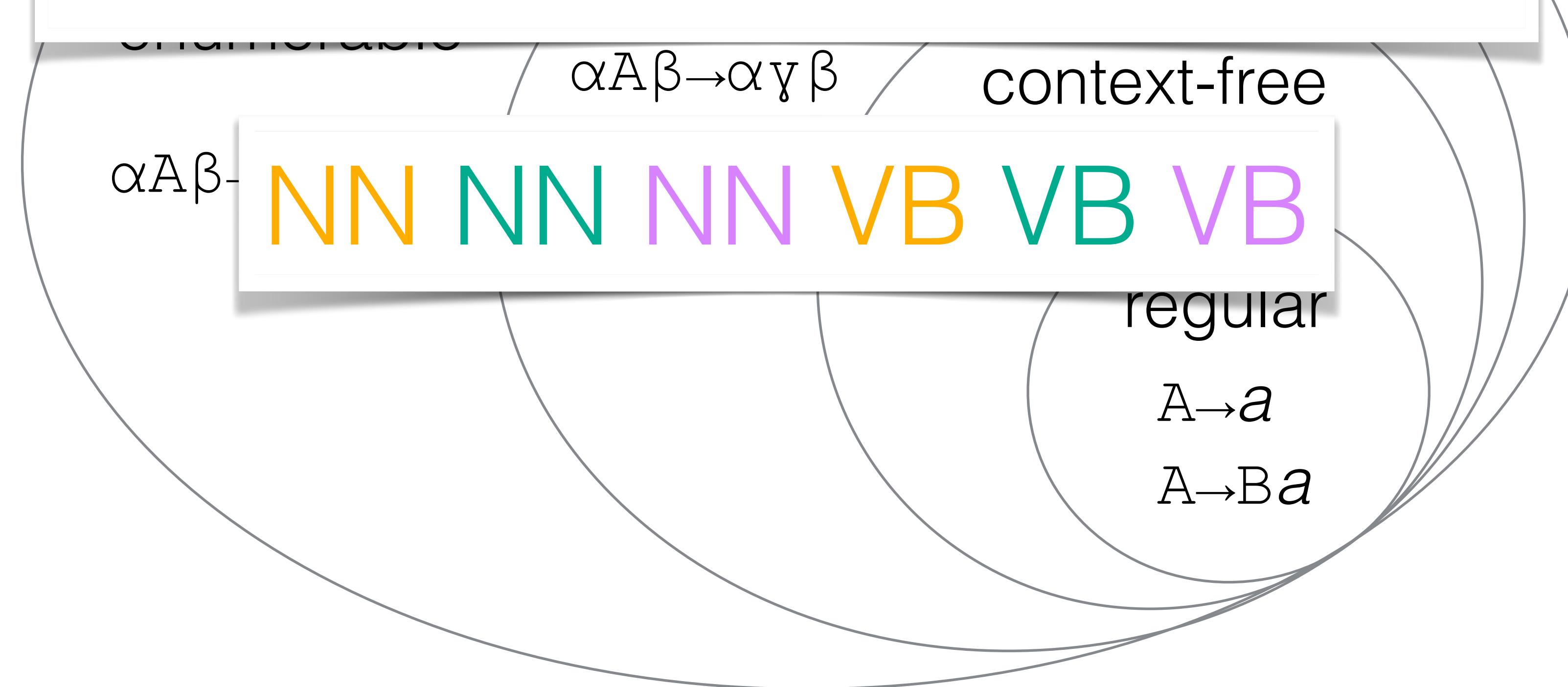
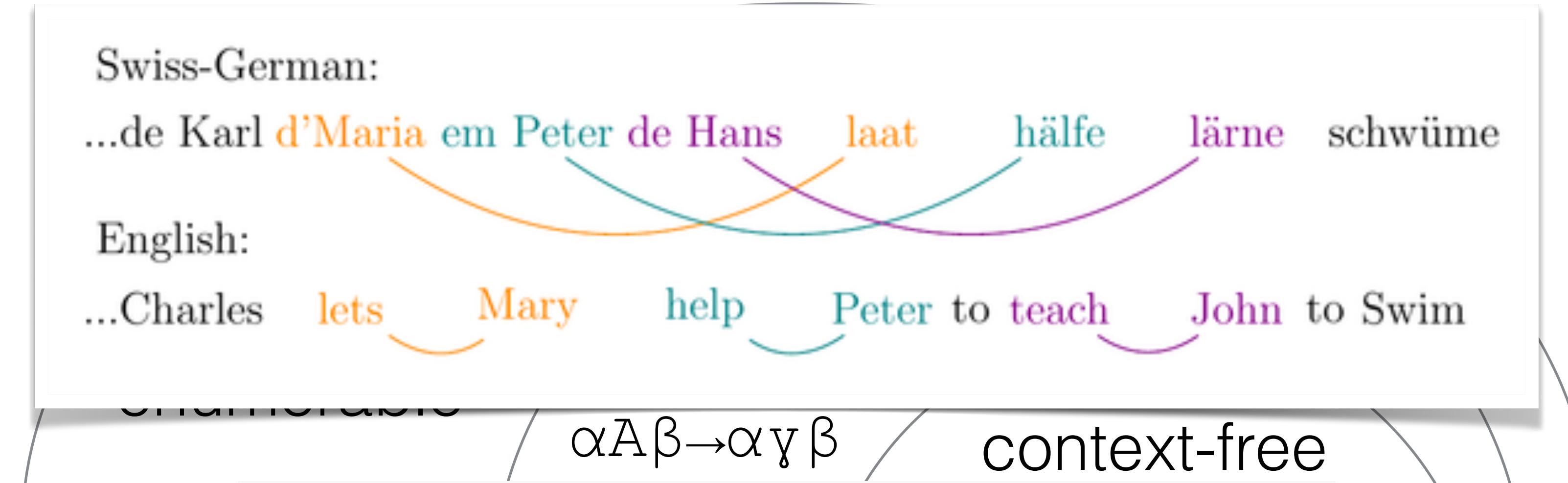
# Chomsky Hierarchy



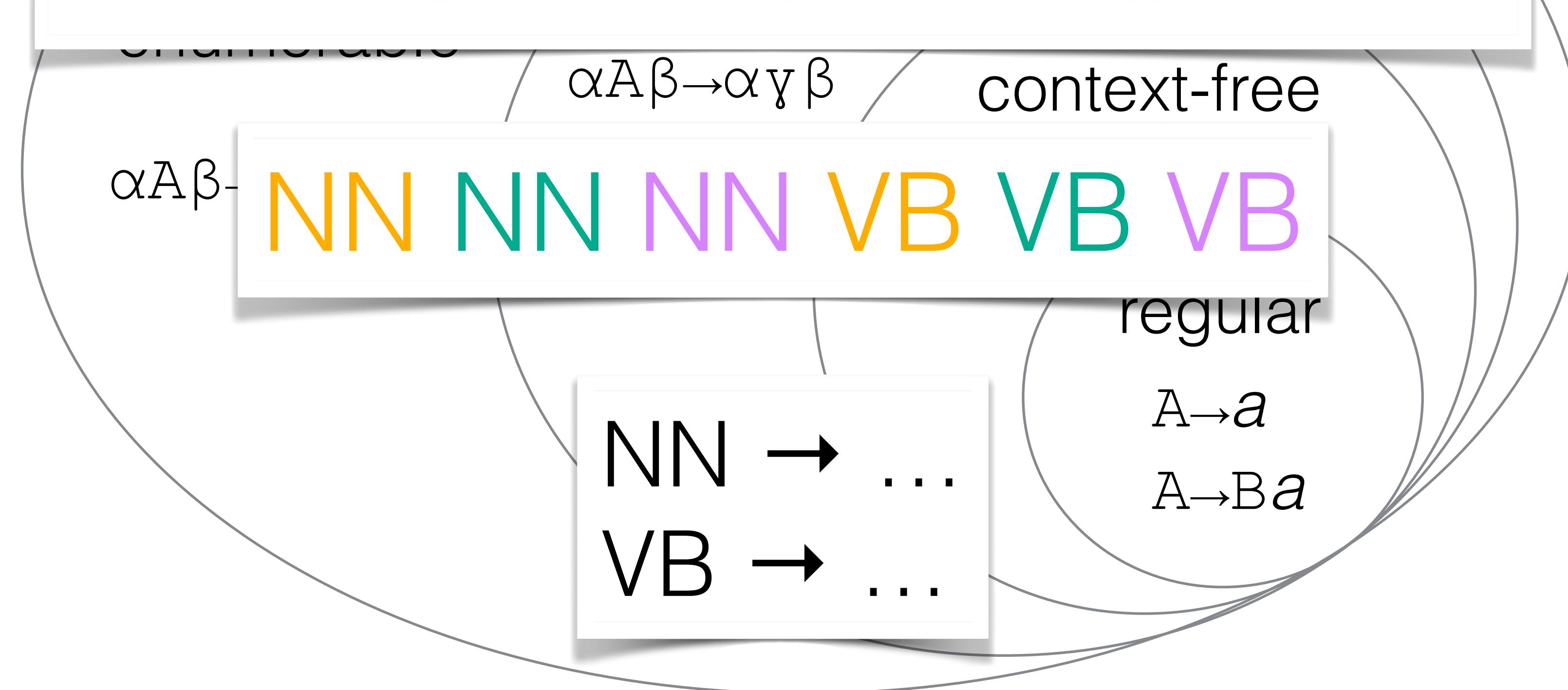
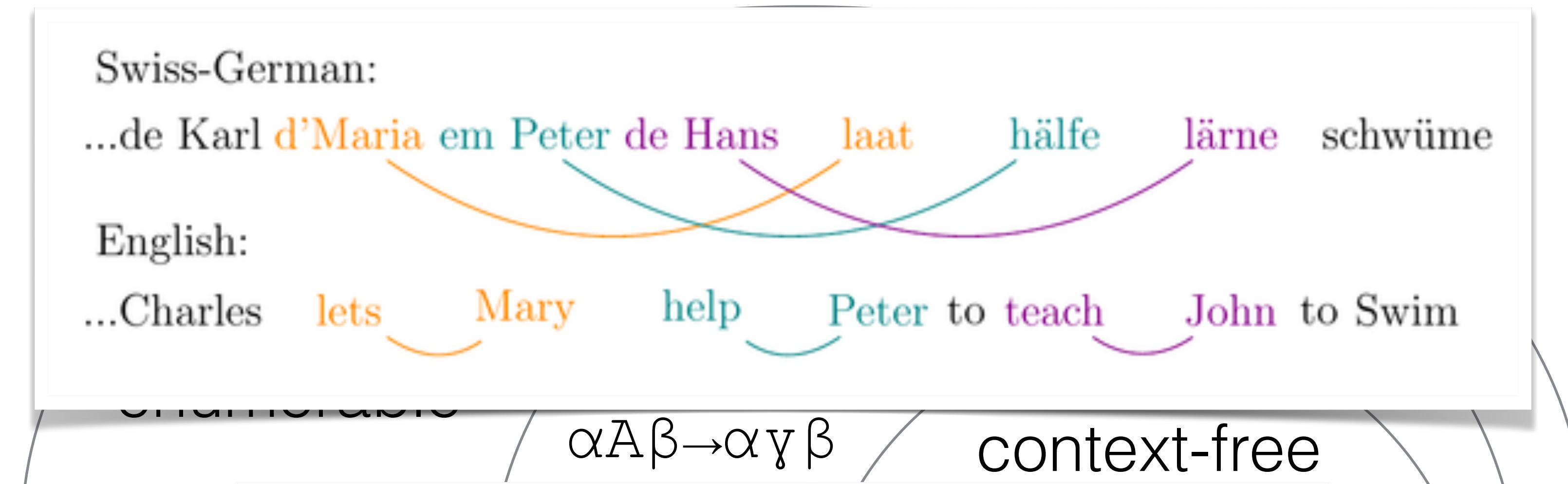
# Chomsky Hierarchy



# Chomsky Hierarchy



# Chomsky Hierarchy



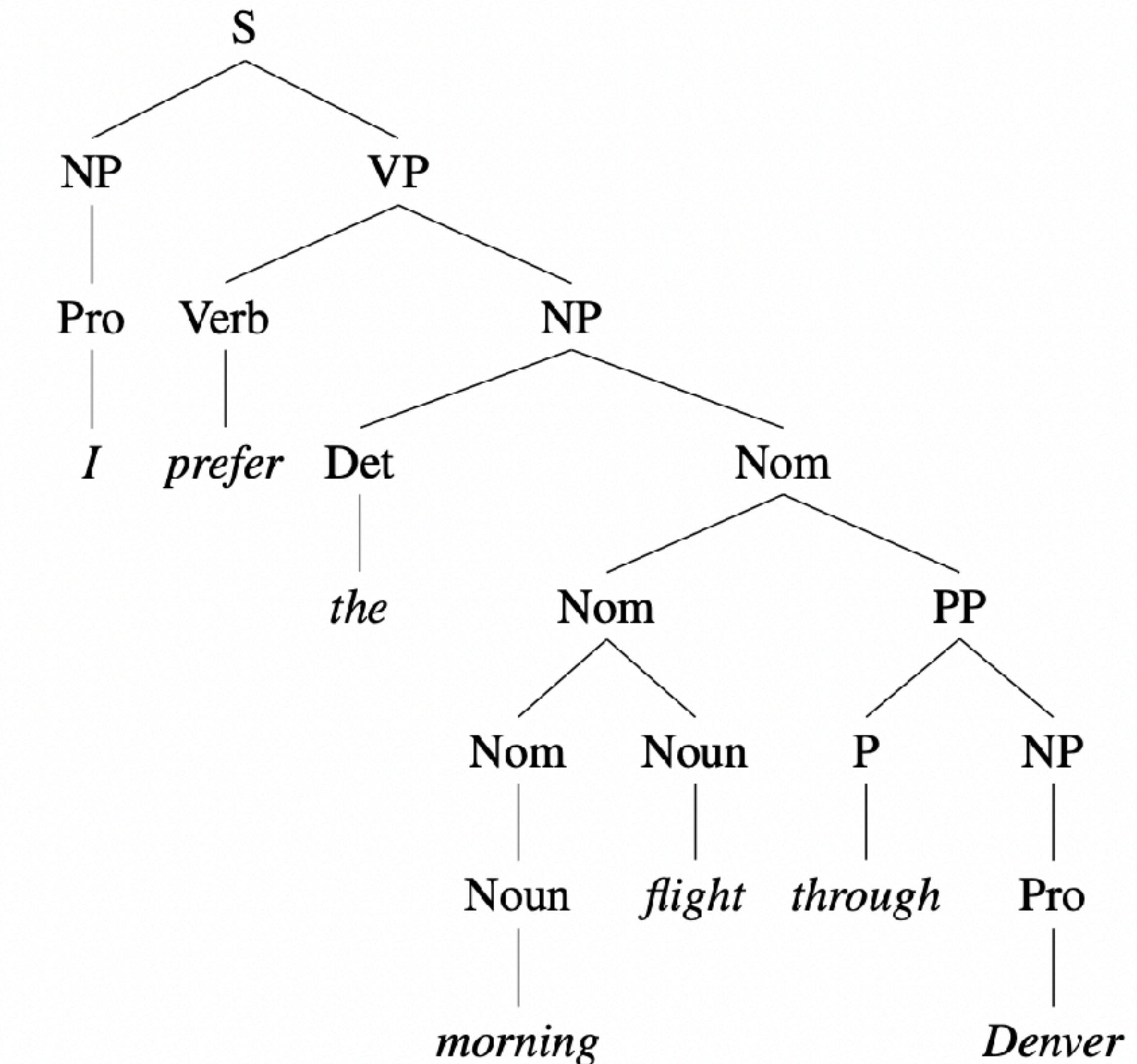


# Topics

- Formal Grammars
- **Constituency Parsing with the CKY Algorithm**
- Dependency Parsing with the Shift-Reduce Algorithm

# Constituency Grammar

- AKA “Phrase Structure Grammar”
- Organize sentences into phrases that are functionally equivalent (w.r.t, the grammar)
- I.e., phrases can be swapped in and out and maintain grammaticality
  - (Note: grammatical != meaningful)



# Constituency Grammar

The bunny

slept soundly under the shade of the umbrella

The cat

played

The dog with one eye and a green handkerchief

jumped and twirled

# Constituency Grammar

The bunny

jumped and twirled

The cat

slept soundly under the shade of the umbrella

The dog with one eye and a green handkerchief

played

S → NP VP

S → Aux NP VP

S → VP

NP → Pronoun

NP → Proper-Noun

NP → Det Nominal

Nominal → Noun

Nominal → Nominal Noun

Nominal → Nominal PP

VP → Verb

VP → Verb NP

VP → Verb NP PP

VP → Verb PP

VP → VP PP

PP → Preposition NP

Det → that | this | a

Noun → book | flight | meal | money

Verb → book | include | prefer

Pronoun → I | she | me

Proper-Noun → Houston | NWA

Aux → does

Preposition → from | to | on | near | through

S → NP VP **Grammar**

S → Aux NP VP

S → VP

NP → Pronoun

NP → Proper-Noun

NP → Det Nominal

Nominal → Noun

Nominal → Nominal Noun

Nominal → Nominal PP

VP → Verb

VP → Verb NP

VP → Verb NP PP

VP → Verb PP

VP → VP PP

PP → Preposition NP

Det → that | this | a

Noun → book | flight | meal | money

Verb → book | include | prefer

Pronoun → I | she | me

Proper-Noun → Houston | NWA

Aux → does

Preposition → from | to | on | near | through

**Lexicon**

S → NP VP **Grammar**

S → Aux NP VP

S → VP

NP → Pronoun

NP → Proper-Noun

NP → Det Nominal

Nominal → Noun

Nominal → Nominal Noun

Nominal → Nominal PP

VP → Verb

VP → Verb NP

VP → Verb NP PP

VP → Verb PP

VP → VP PP

PP → Preposition NP

Det → that | this | a

Noun → book | flight | meal | money

Verb → book | include | prefer

Pronoun → I | she | me

Proper-Noun → Houston | NWA

Aux → does

Preposition → from | to | on | near | through

**Lexicon**

*book      that      flight*

S → NP VP **Grammar**

S → Aux NP VP

S → VP

NP → Pronoun

NP → Proper-Noun

NP → Det Nominal

Nominal → Noun

Nominal → Nominal Noun

Nominal → Nominal PP

VP → Verb

VP → Verb NP

VP → Verb NP PP

VP → Verb PP

VP → VP PP

PP → Preposition NP

**Det** → **that** | this | a

**Noun** → book | **flight** | meal | money

**Verb** → **book** | include | prefer

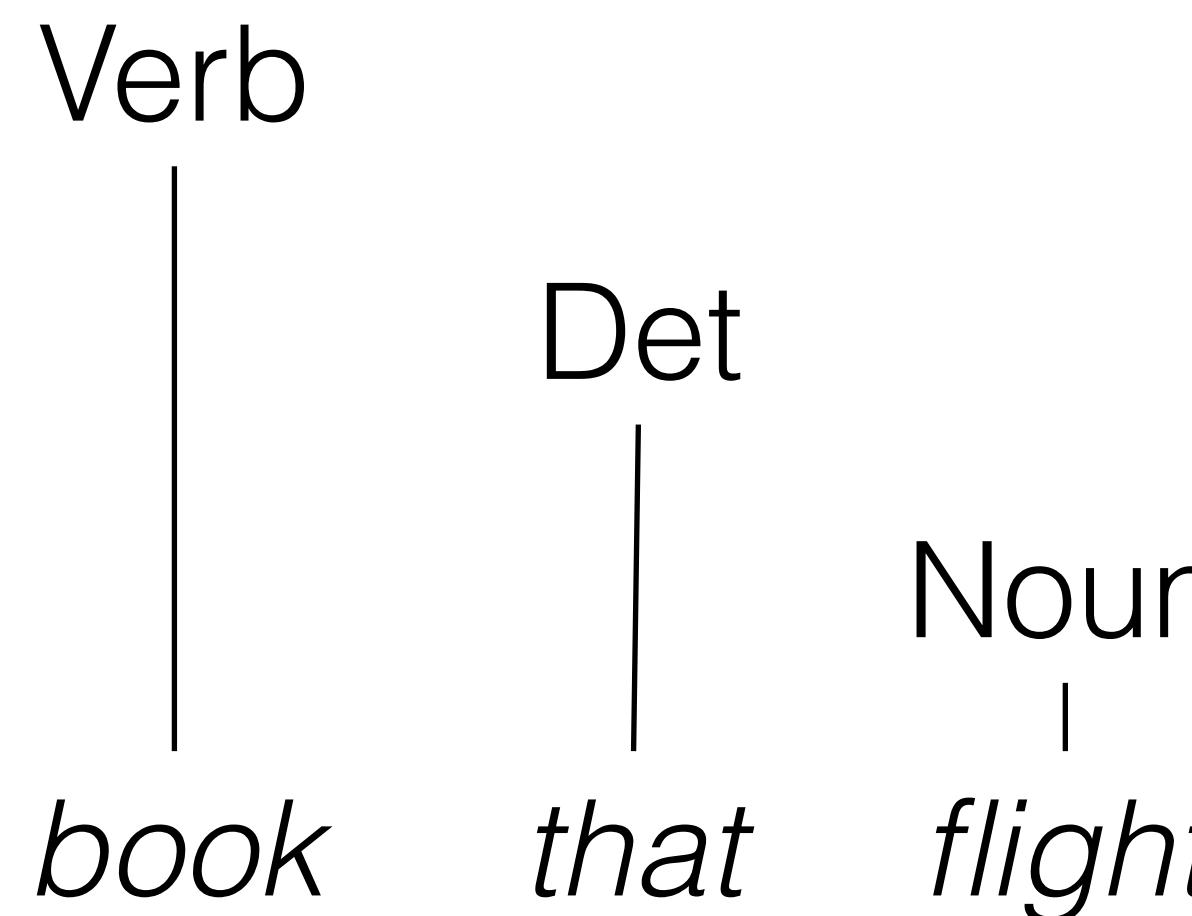
Pronoun → I | she | me

Proper-Noun → Houston | NWA

Aux → does

Preposition → from | to | on | near | through

**Lexicon**



$S \rightarrow NP VP$  **Grammar**

$S \rightarrow Aux NP VP$

$S \rightarrow VP$

$NP \rightarrow Pronoun$

$NP \rightarrow Proper-Noun$

$NP \rightarrow Det Nominal$

### **Nominal → Noun**

$Nominal \rightarrow Nominal\ Noun$

$Nominal \rightarrow Nominal\ PP$

$VP \rightarrow Verb$

$VP \rightarrow Verb\ NP$

$VP \rightarrow Verb\ NP\ PP$

$VP \rightarrow Verb\ PP$

$VP \rightarrow VP\ PP$

$PP \rightarrow Preposition\ NP$

$Det \rightarrow that | this | a$

$Noun \rightarrow book | flight | meal | money$

$Verb \rightarrow book | include | prefer$

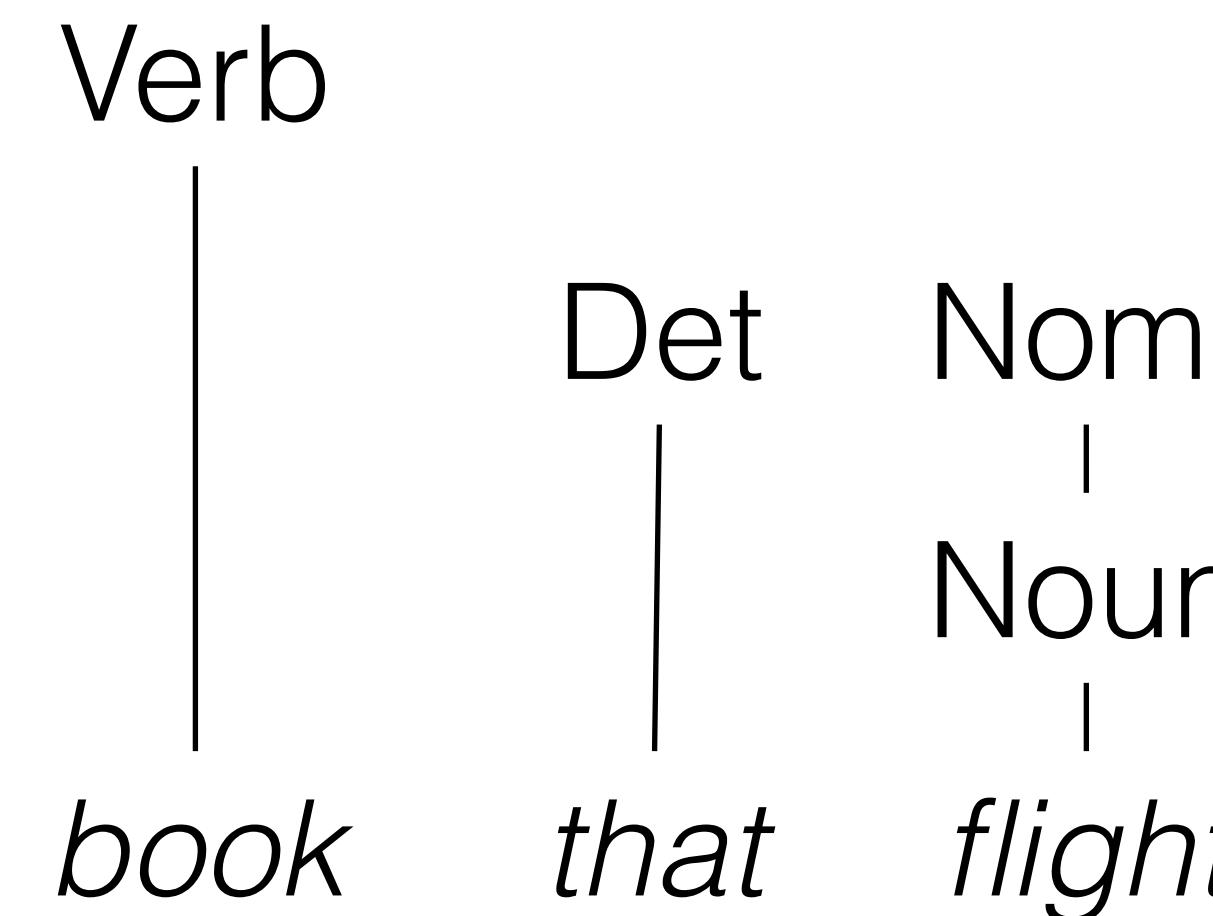
$Pronoun \rightarrow I | she | me$

$Proper-Noun \rightarrow Houston | NWA$

$Aux \rightarrow does$

$Preposition \rightarrow from | to | on | near | through$

### **Lexicon**



S → NP VP **Grammar**

S → Aux NP VP

S → VP

NP → Pronoun

NP → Proper-Noun

**NP → Det Nominal**

Nominal → Noun

Nominal → Nominal Noun

Nominal → Nominal PP

VP → Verb

VP → Verb NP

VP → Verb NP PP

VP → Verb PP

VP → VP PP

PP → Preposition NP

Det → that | this | a

Noun → book | flight | meal | money

Verb → book | include | prefer

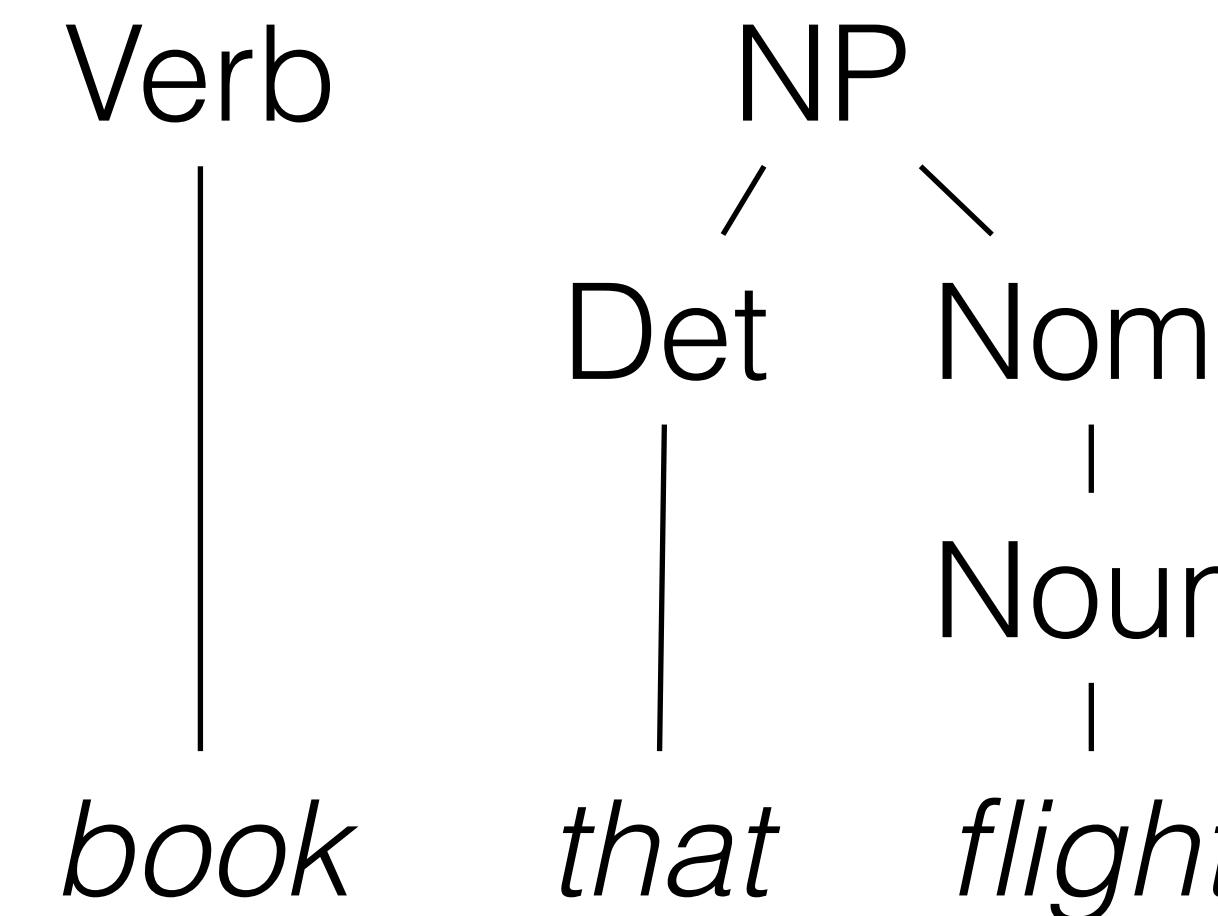
Pronoun → I | she | me

Proper-Noun → Houston | NWA

Aux → does

Preposition → from | to | on | near | through

**Lexicon**



$S \rightarrow NP\ VP$  **Grammar**

$S \rightarrow Aux\ NP\ VP$

$S \rightarrow VP$

$NP \rightarrow Pronoun$

$NP \rightarrow Proper-Noun$

$NP \rightarrow Det\ Nominal$

$Nominal \rightarrow Noun$

$Nominal \rightarrow Nominal\ Noun$

$Nominal \rightarrow Nominal\ PP$

$VP \rightarrow Verb$

**$VP \rightarrow Verb\ NP$**

$VP \rightarrow Verb\ NP\ PP$

$VP \rightarrow Verb\ PP$

$VP \rightarrow VP\ PP$

$PP \rightarrow Preposition\ NP$

$Det \rightarrow that\ | this\ | a$

$Noun \rightarrow book\ | flight\ | meal\ | money$

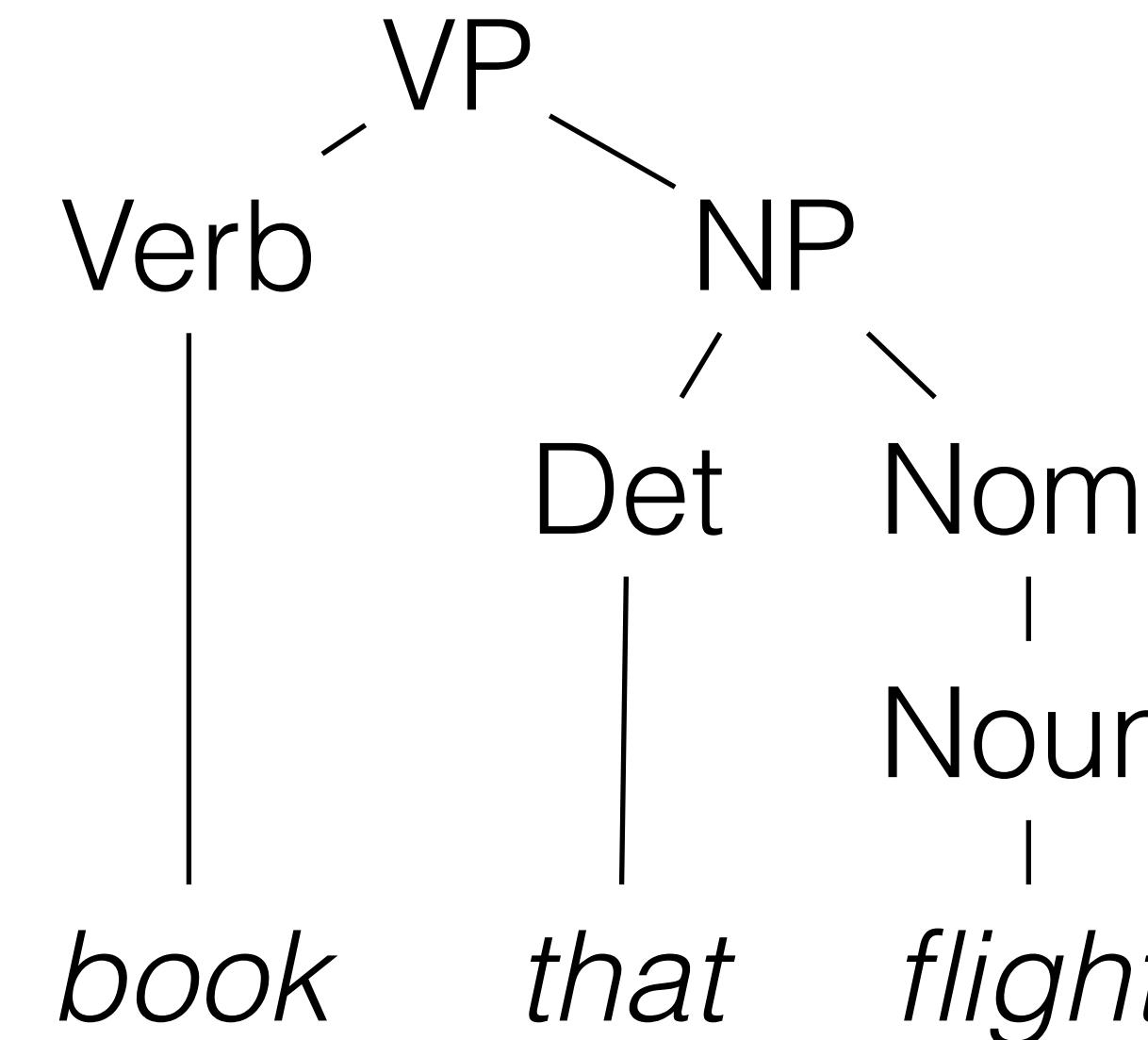
$Verb \rightarrow book\ | include\ | prefer$

$Pronoun \rightarrow I\ | she\ | me$

$Proper-Noun \rightarrow Houston\ | NWA$

$Aux \rightarrow does$

$Preposition \rightarrow from\ | to\ | on\ | near\ | through$



$S \rightarrow NP\ VP$  **Grammar**

$S \rightarrow Aux\ NP\ VP$

**$S \rightarrow VP$**

$NP \rightarrow Pronoun$

$NP \rightarrow Proper-Noun$

$NP \rightarrow Det\ Nominal$

$Nominal \rightarrow Noun$

$Nominal \rightarrow Nominal\ Noun$

$Nominal \rightarrow Nominal\ PP$

$VP \rightarrow Verb$

$VP \rightarrow Verb\ NP$

$VP \rightarrow Verb\ NP\ PP$

$VP \rightarrow Verb\ PP$

$VP \rightarrow VP\ PP$

$PP \rightarrow Preposition\ NP$

$Det \rightarrow that\ | this\ | a$

$Noun \rightarrow book\ | flight\ | meal\ | money$

$Verb \rightarrow book\ | include\ | prefer$

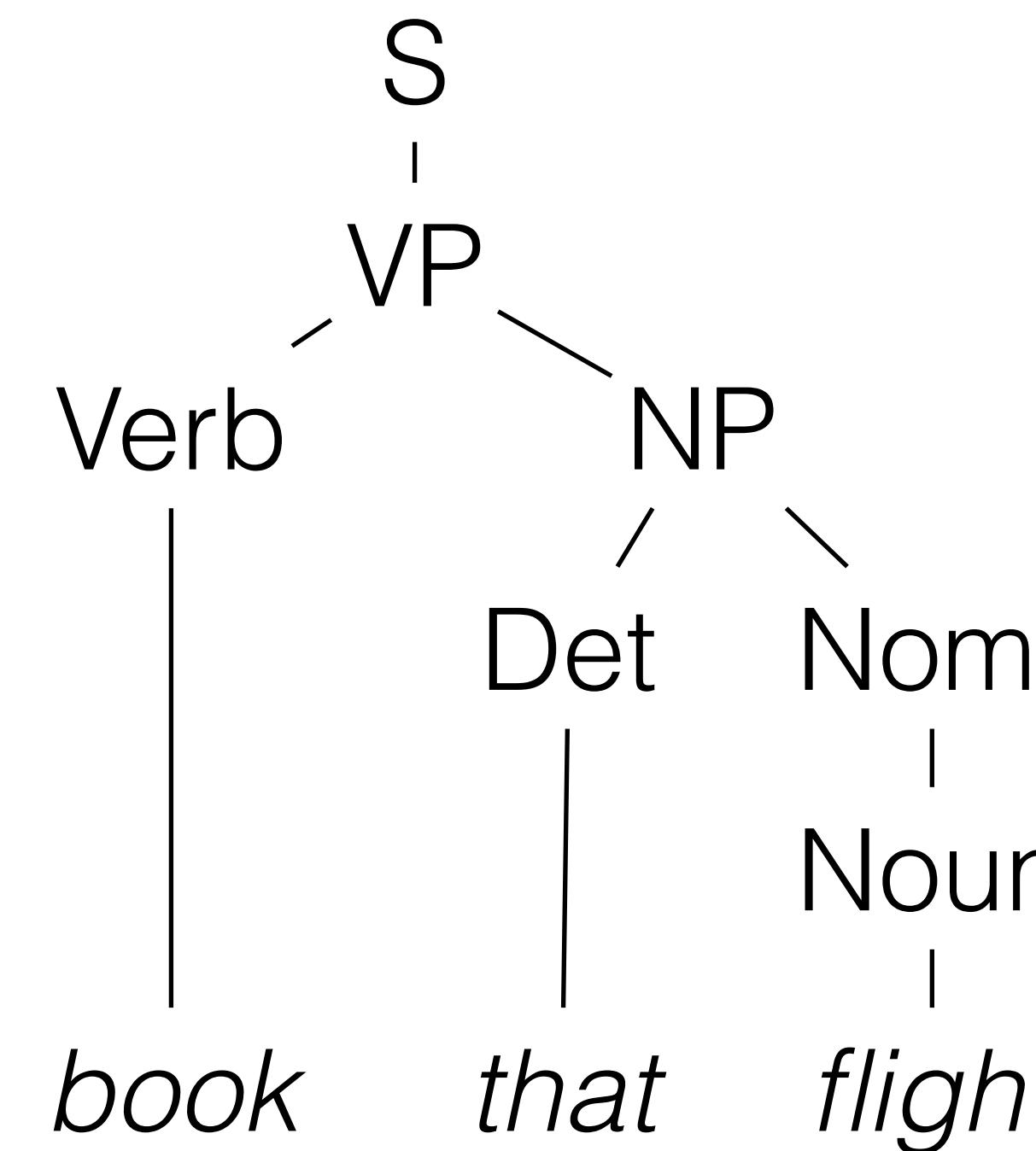
$Pronoun \rightarrow I\ | she\ | me$

$Proper-Noun \rightarrow Houston\ | NWA$

$Aux \rightarrow does$

$Preposition \rightarrow from\ | to\ | on\ | near\ | through$

**Lexicon**



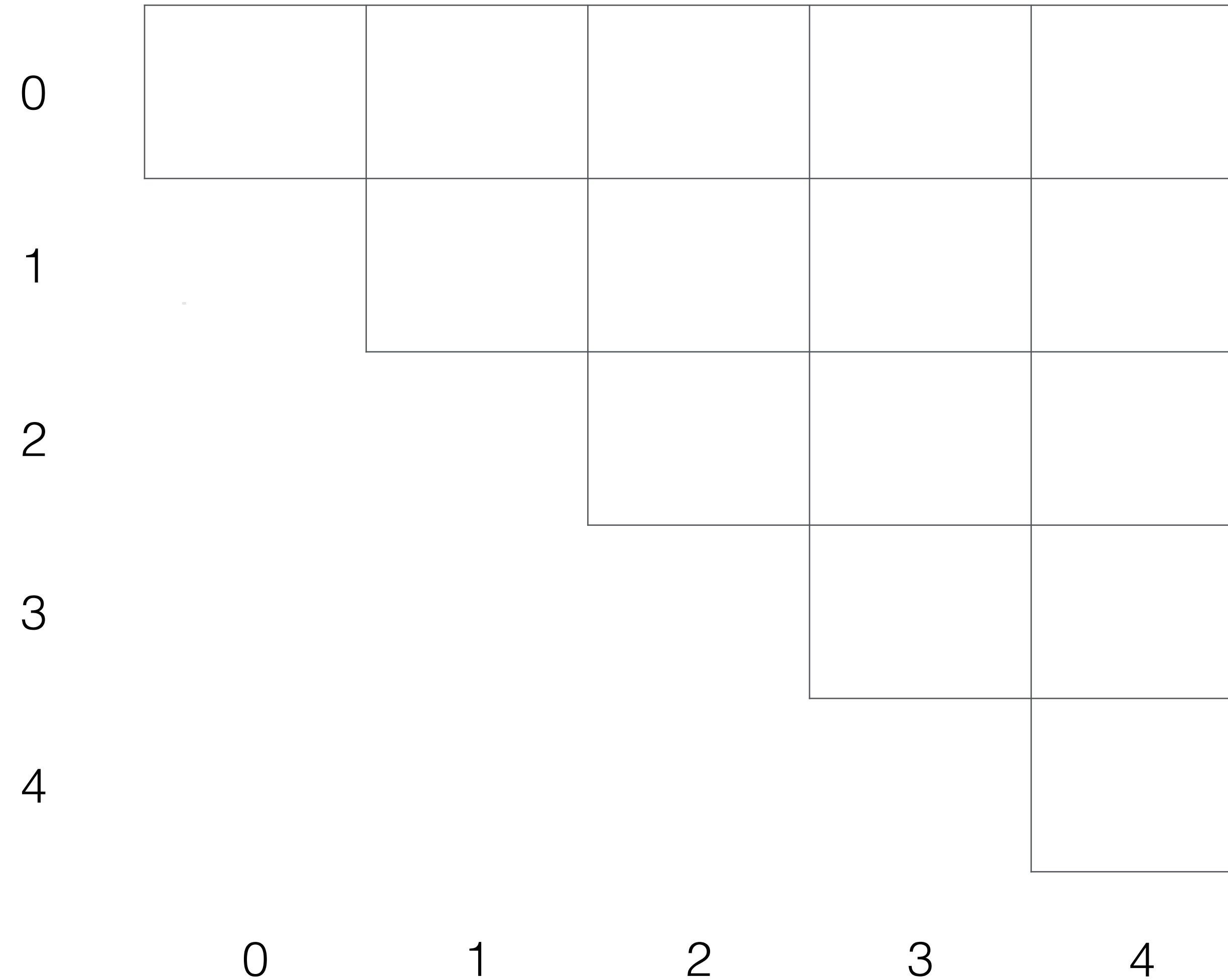


# CKY Parsing

- Dynamic Programming Algorithm → breaks problem into sub problems to solve more efficiently
- Basic idea: find a parse for words i through j by combining parses for words i through k with parses for works k through j

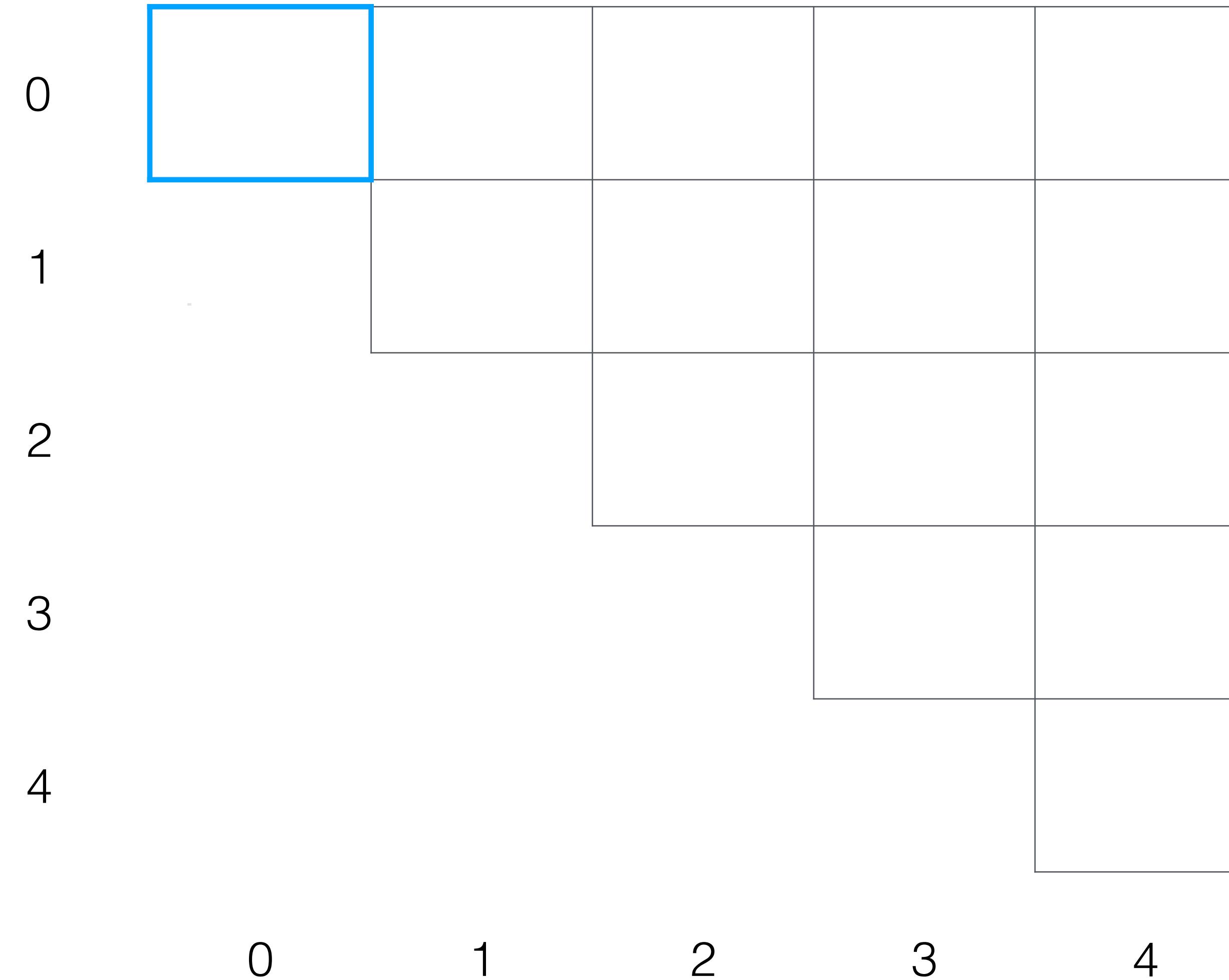
# CKY Parsing

0 book 1 the 2 flight 3 through 4 Houston 5



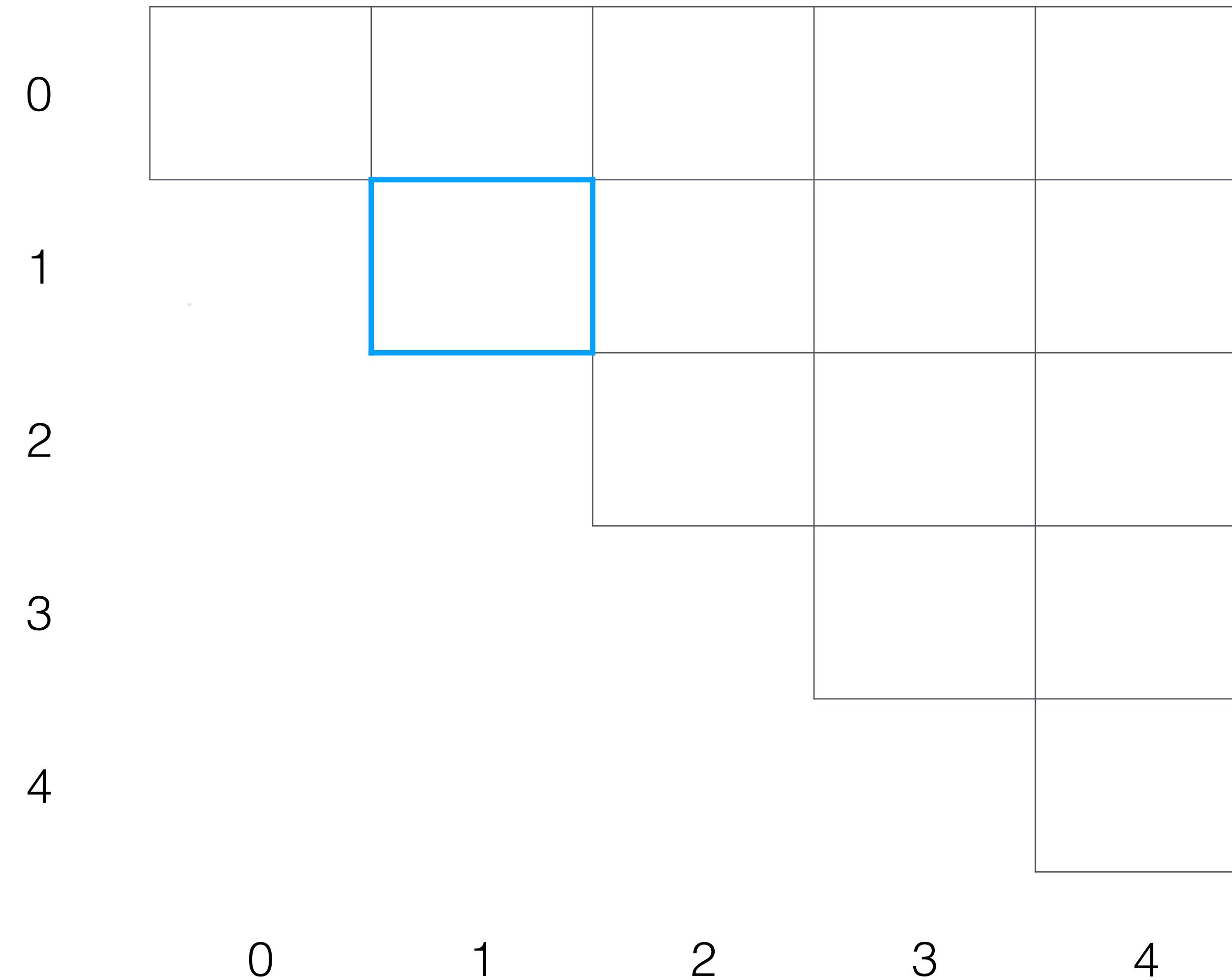
# CKY Parsing

0 book 1 the 2 flight 3 through 4 Houston 5



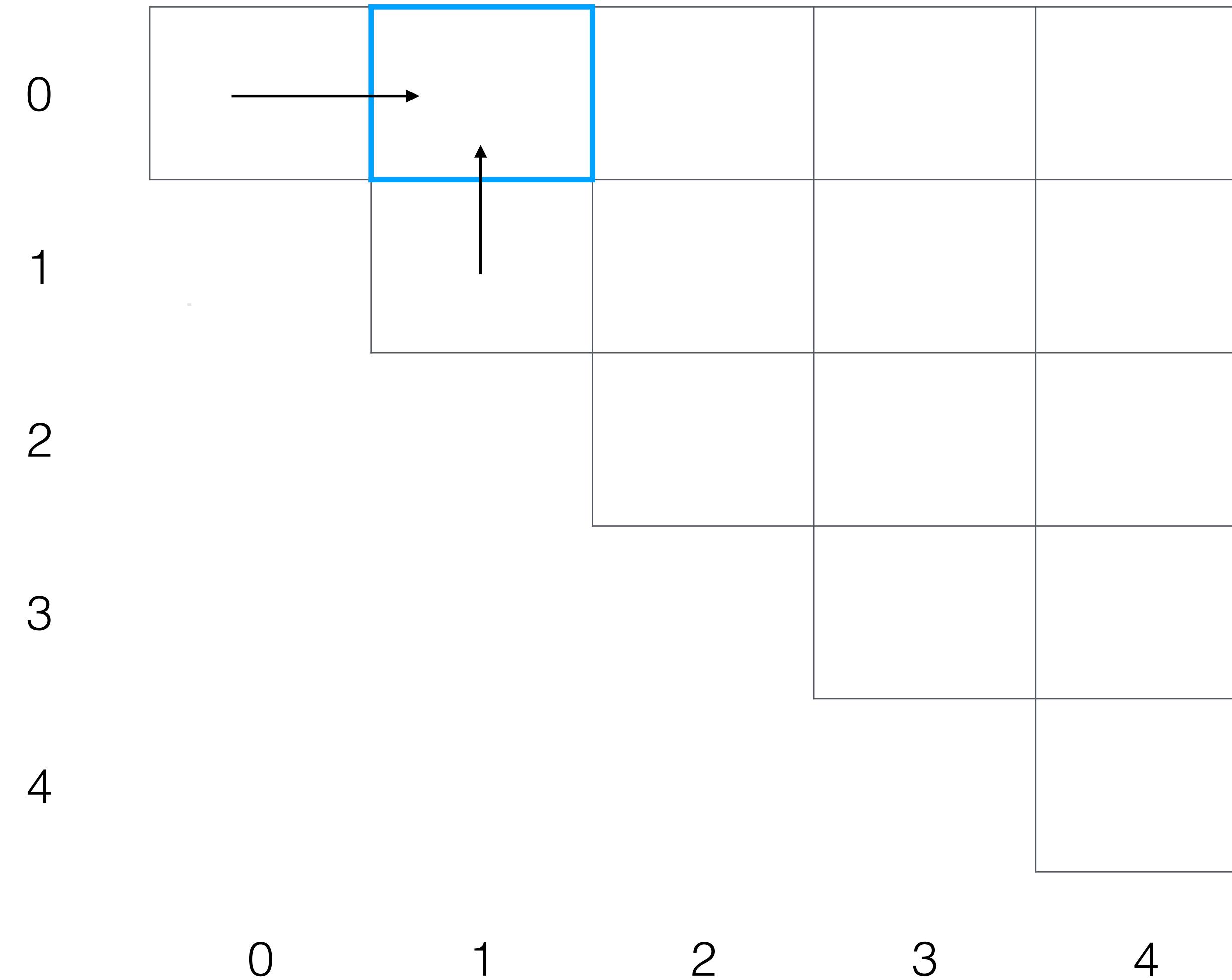
# CKY Parsing

0 book 1 the 2 flight 3 through 4 Houston 5



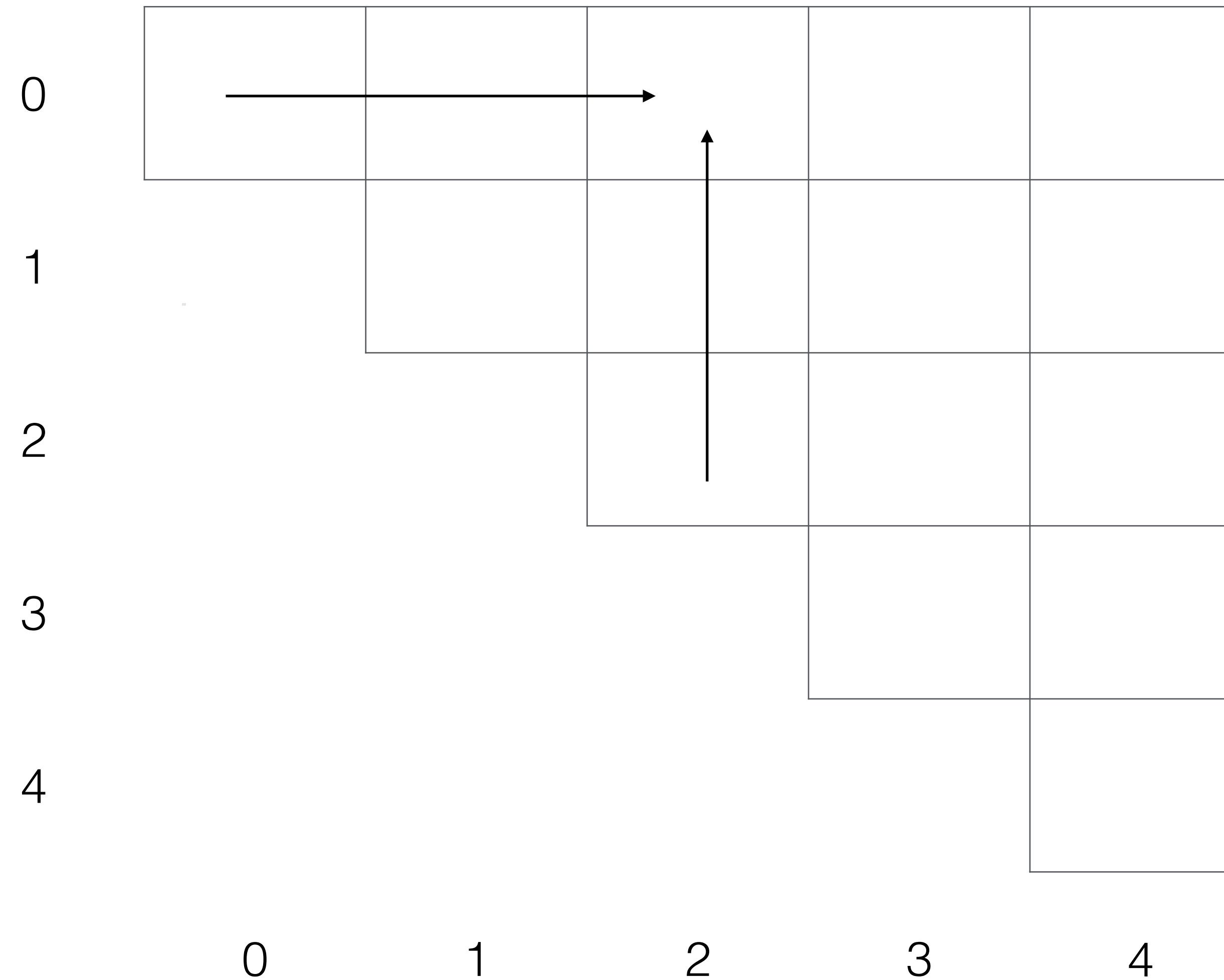
# CKY Parsing

0 book 1 the 2 flight 3 through 4 Houston 5



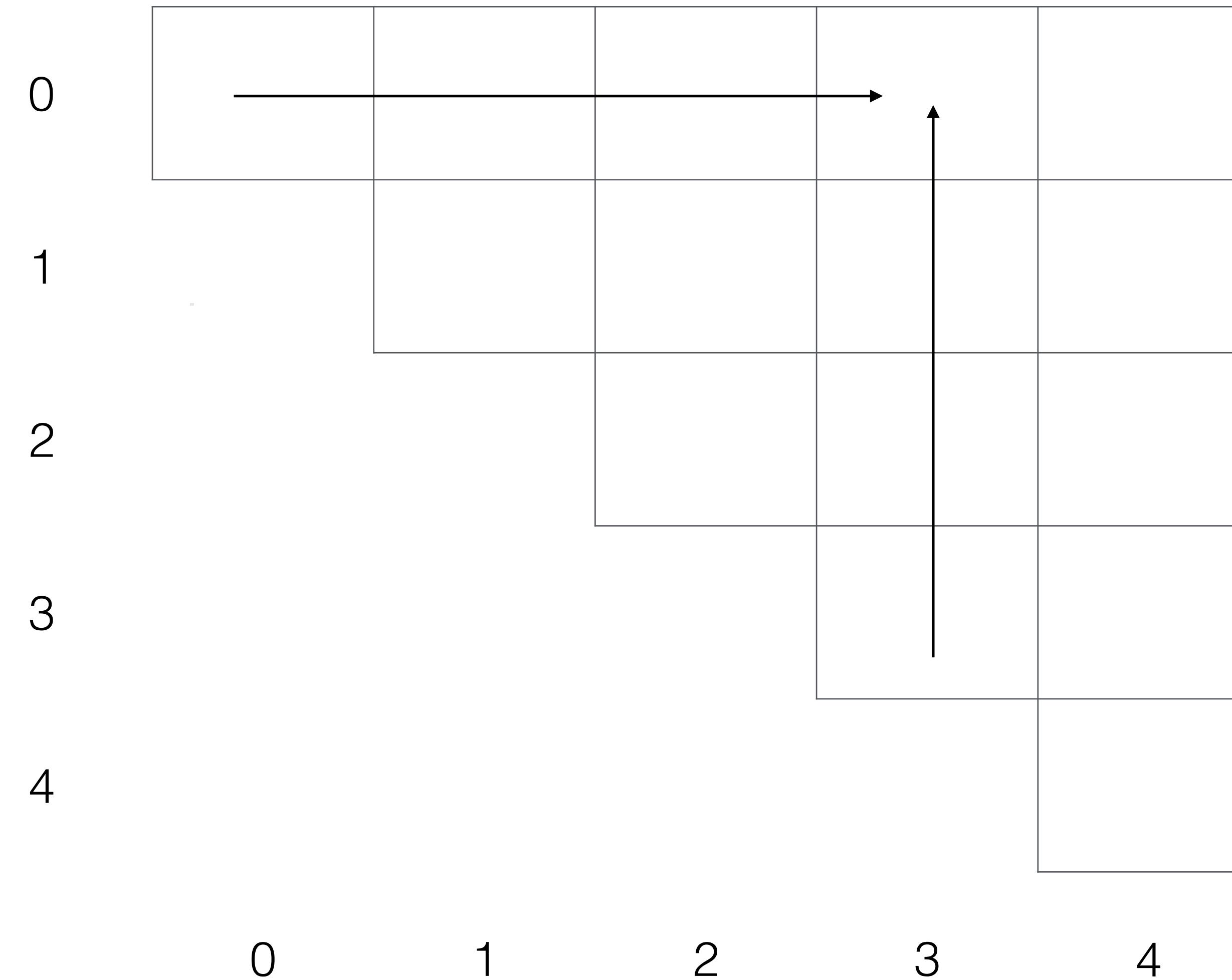
# CKY Parsing

0 book 1 the 2 flight 3 through 4 Houston 5



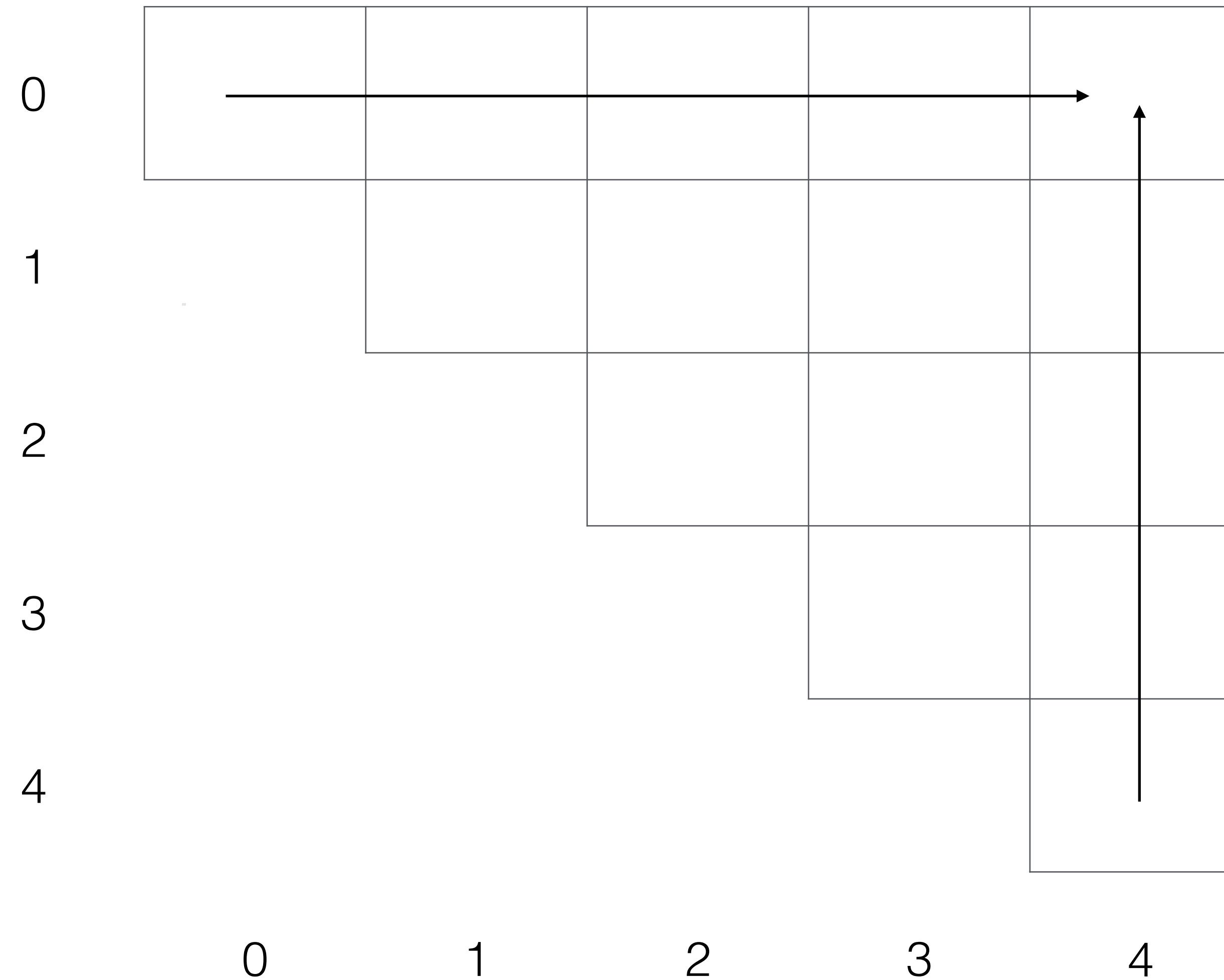
# CKY Parsing

0 book 1 the 2 flight 3 through 4 Houston 5



# CKY Parsing

0 book 1 the 2 flight 3 through 4 Houston 5

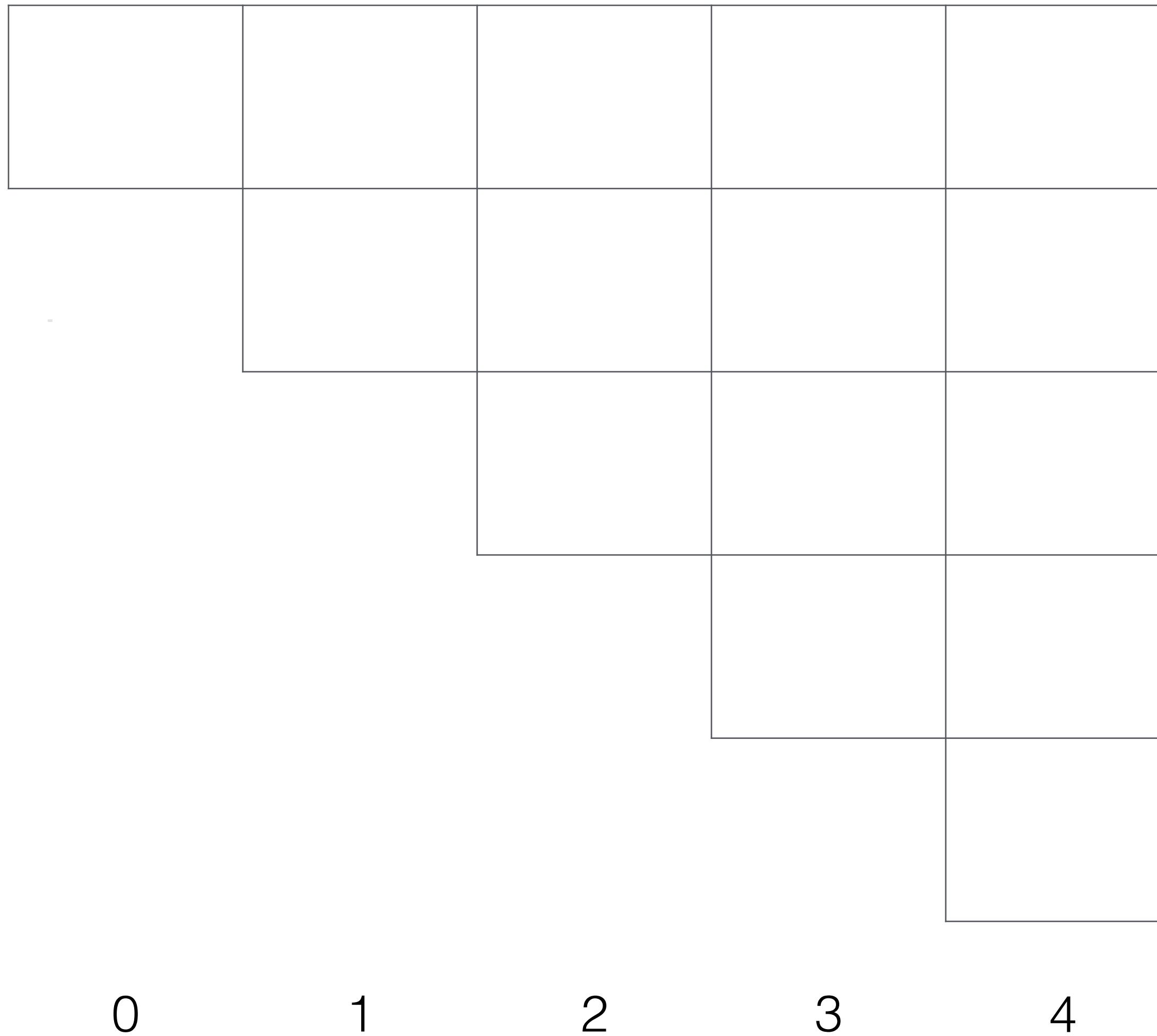


```

S → NP VP
S → book | prefer
S → Verb NP
S → X2 PP
S → Verb PP
S → VP PP
NP → TWA | Houston
NP → Det Nominal
Nominal → book | flight
Nominal → Nominal PP
VP → book | prefer
VP → Verb NP
VP → X2 PP
X2 → Verb NP
VP → Verb PP
VP → VP PP
PP → Prep. NP
Det → that | this | a
Noun → book | flight
Verb → book | prefer
Prop-N → Houston | NWA
Aux → does
Prep. → from | to | through

```

0 book 1 the 2 flight 3 through 4 Houston 5



$S \rightarrow NP VP$

**S → book | prefer**

$S \rightarrow Verb\ NP$

$S \rightarrow X2\ PP$

$S \rightarrow Verb\ PP$

$S \rightarrow VP\ PP$

$NP \rightarrow TWA\mid Houston$

$NP \rightarrow Det\ Nominal$

**Nominal → book | flight**

$Nominal \rightarrow Nominal\ PP$

**VP → book | prefer**

$VP \rightarrow Verb\ NP$

$VP \rightarrow X2\ PP$

$X2 \rightarrow Verb\ NP$

$VP \rightarrow Verb\ PP$

$VP \rightarrow VP\ PP$

$PP \rightarrow Prep.\ NP$

$Det \rightarrow that\mid this\mid a$

**Noun → book | flight**

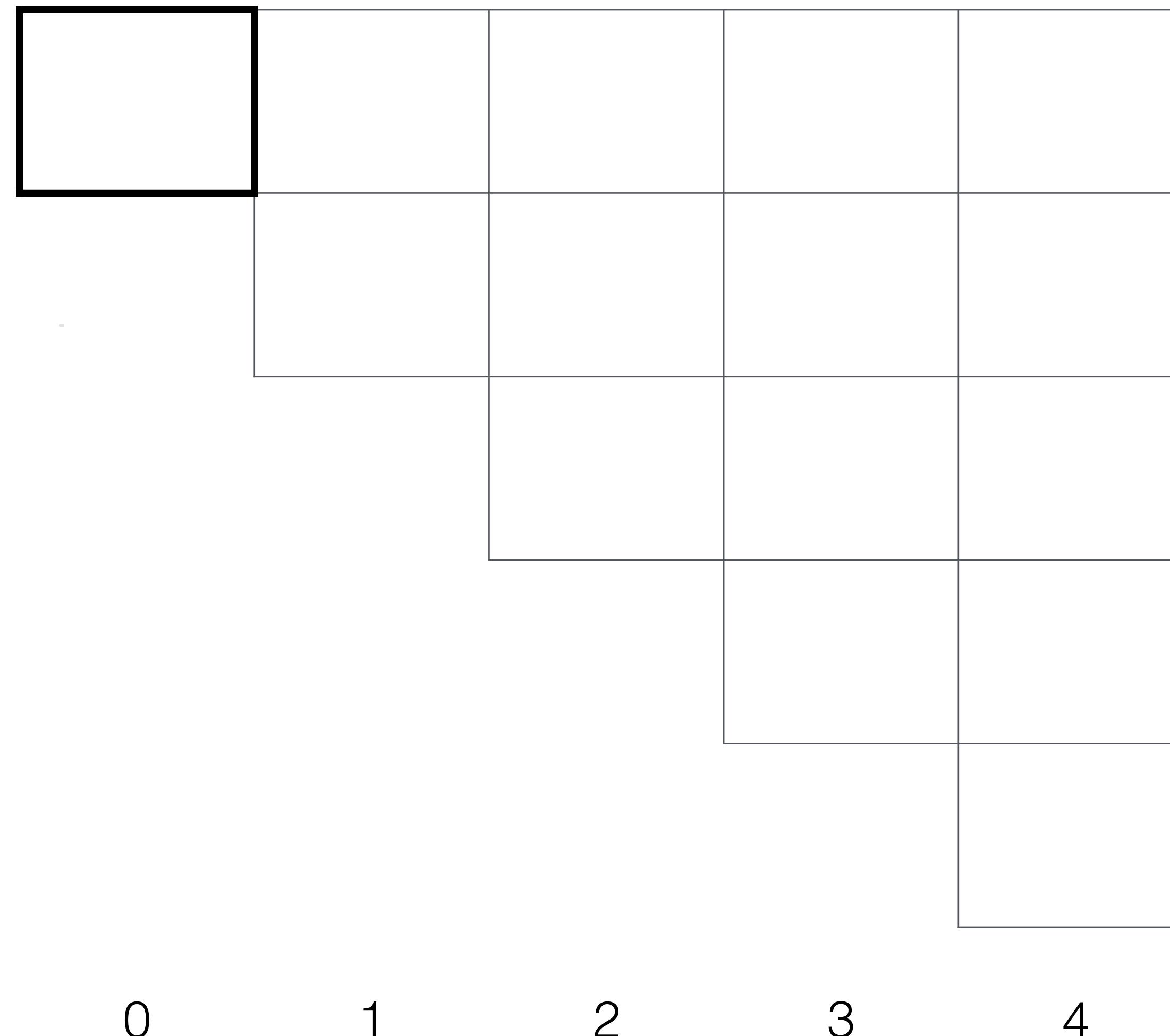
**Verb → book | prefer**

$Prop-N \rightarrow Houston\mid NWA$

$Aux \rightarrow does$

$Prep. \rightarrow from\mid to\mid through$

0 book 1 the 2 flight 3 through 4 Houston 5



$S \rightarrow NP\ VP$

**S → book | prefer**

$S \rightarrow Verb\ NP$

$S \rightarrow X2\ PP$

$S \rightarrow Verb\ PP$

$S \rightarrow VP\ PP$

$NP \rightarrow TWA\ | Houston$

$NP \rightarrow Det\ Nominal$

**Nominal → book | flight**

$Nominal \rightarrow Nominal\ PP$

**VP → book | prefer**

$VP \rightarrow Verb\ NP$

$VP \rightarrow X2\ PP$

$X2 \rightarrow Verb\ NP$

$VP \rightarrow Verb\ PP$

$VP \rightarrow VP\ PP$

$PP \rightarrow Prep.\ NP$

$Det \rightarrow that\ | this\ | a$

**Noun → book | flight**

**Verb → book | prefer**

$Prop-N \rightarrow Houston\ | NWA$

$Aux \rightarrow does$

$Prep. \rightarrow from\ | to\ | through$

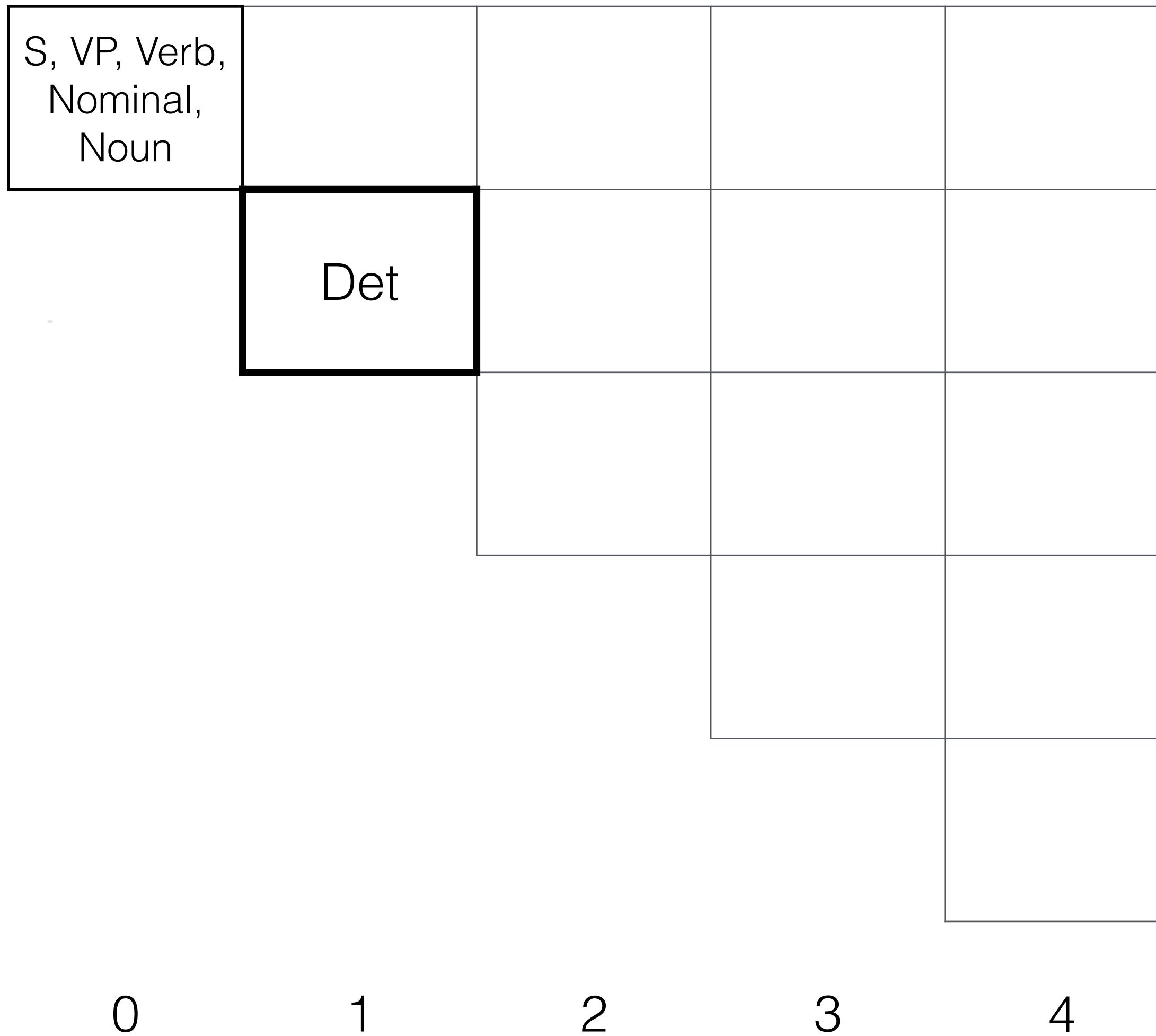
0 **book** 1 the 2 flight 3 through 4 Houston 5

S, VP, Verb,  
Nominal,  
Noun

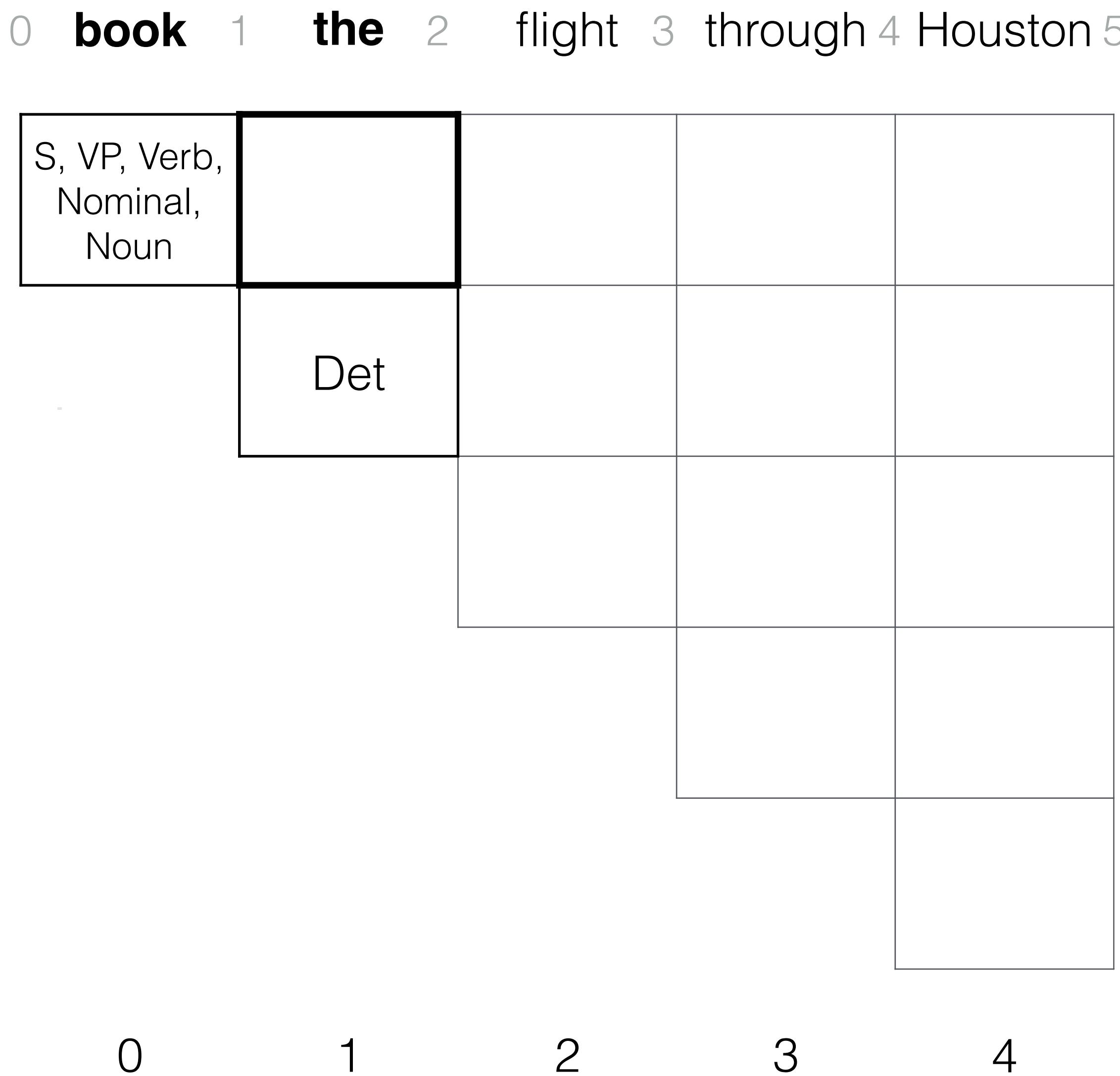
0 1 2 3 4

S → NP VP	
S → book   prefer	
S → Verb NP	
S → X2 PP	
S → Verb PP	0
S → VP PP	
NP → TWA   Houston	
NP → Det Nominal	
Nominal → book   flight	1
Nominal → Nominal PP	
VP → book   prefer	
VP → Verb NP	
VP → X2 PP	2
X2 → Verb NP	
VP → Verb PP	
VP → VP PP	3
PP → Prep. NP	
<b>Det</b> → that   <b>the</b>   a	
Noun → book   flight	4
Verb → book   prefer	
Prop-N → Houston   NWA	
Aux → does	
Prep. → from   to   through	

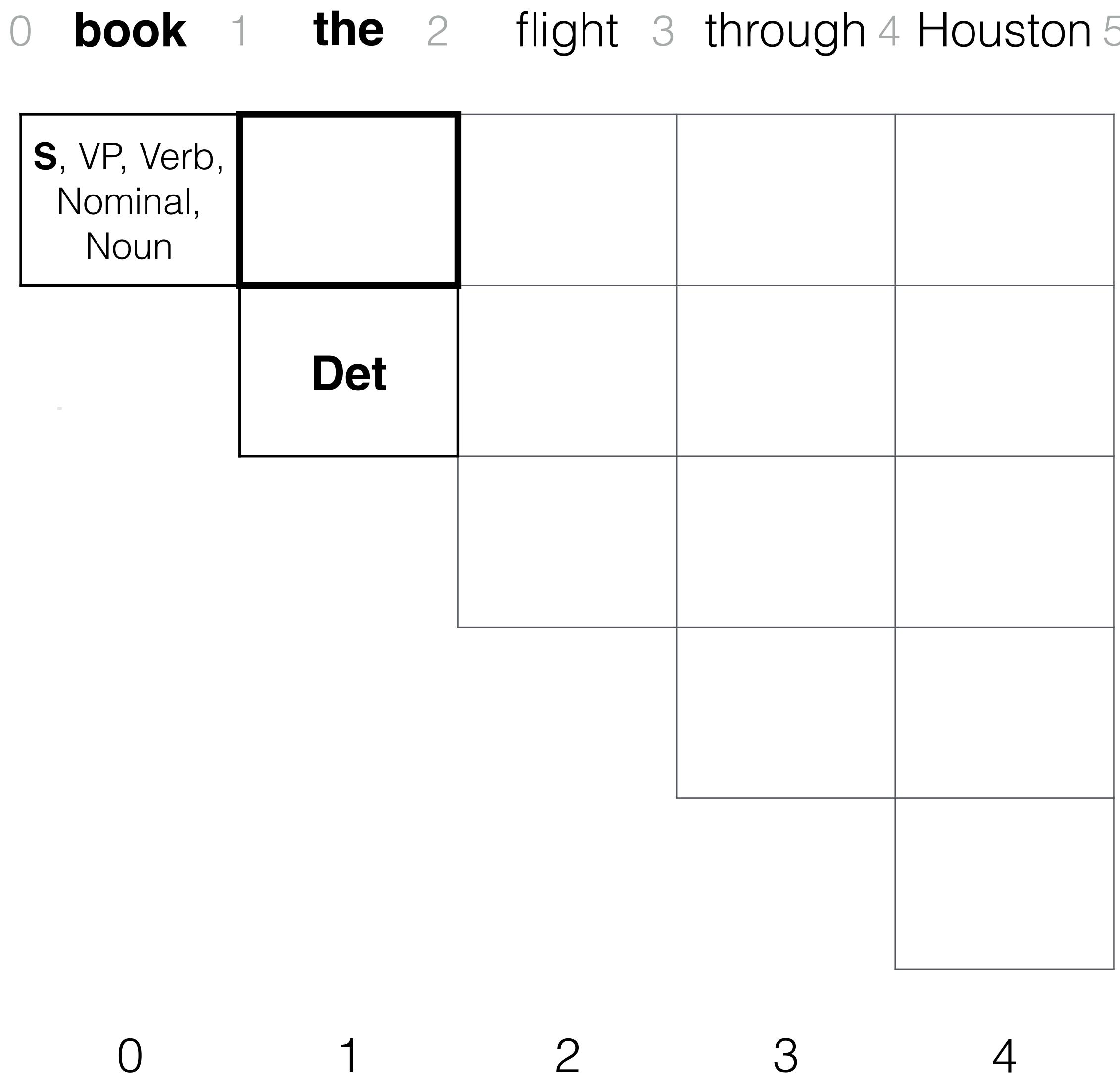
0 book 1 **the** 2 flight 3 through 4 Houston 5



S → NP VP	
S → book   prefer	0
S → Verb NP	
S → X2 PP	
S → Verb PP	
S → VP PP	
NP → TWA   Houston	0
NP → Det Nominal	
Nominal → book   flight	1
Nominal → Nominal PP	
VP → book   prefer	
VP → Verb NP	
VP → X2 PP	2
X2 → Verb NP	
VP → Verb PP	
VP → VP PP	3
PP → Prep. NP	
Det → that   the   a	
Noun → book   flight	4
Verb → book   prefer	
Prop-N → Houston   NWA	
Aux → does	
Prep. → from   to   through	

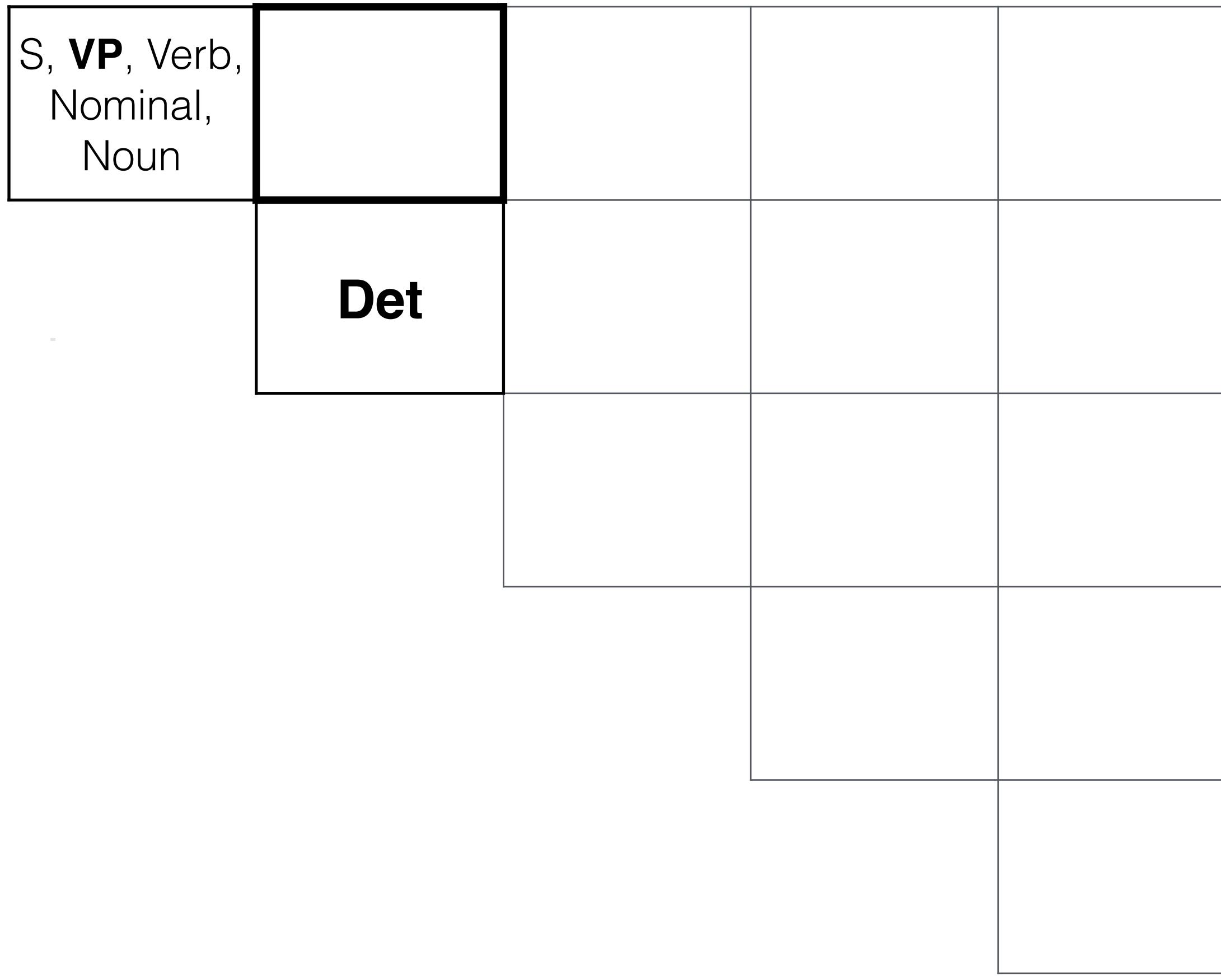


S → NP VP	
S → book   prefer	
S → Verb NP	
S → X2 PP	
S → Verb PP	0
S → VP PP	
NP → TWA   Houston	
NP → Det Nominal	
Nominal → book   flight	1
Nominal → Nominal PP	
VP → book   prefer	
VP → Verb NP	
VP → X2 PP	2
X2 → Verb NP	
VP → Verb PP	
VP → VP PP	3
PP → Prep. NP	
Det → that   the   a	
Noun → book   flight	4
Verb → book   prefer	
Prop-N → Houston   NWA	
Aux → does	
Prep. → from   to   through	

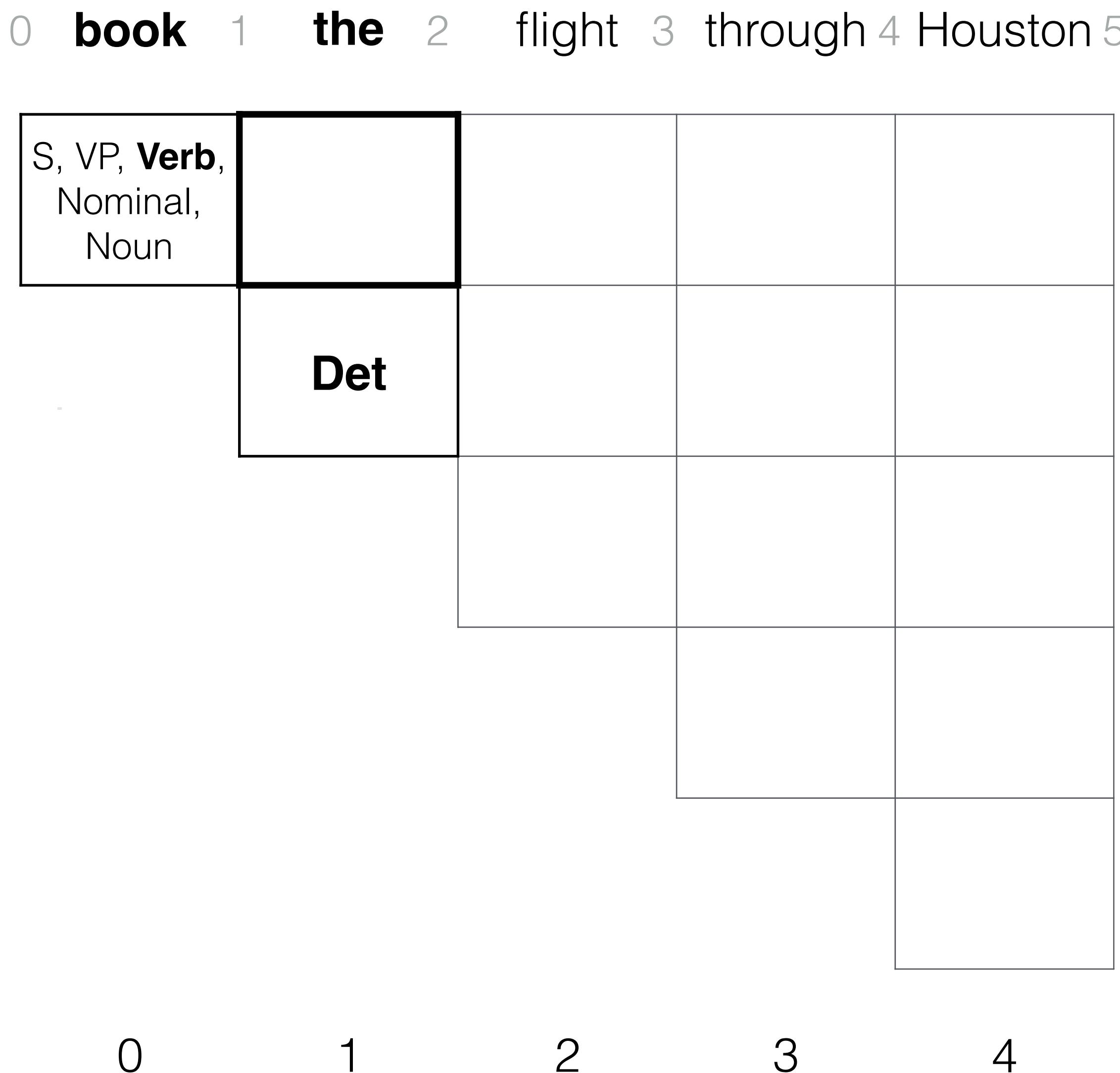


$S \rightarrow NP VP$   
 $S \rightarrow book | prefer$   
 $S \rightarrow Verb NP$   
 $S \rightarrow X2 PP$   
 $S \rightarrow Verb PP$   
 $S \rightarrow VP PP$   
 $NP \rightarrow TWA | Houston$   
 $NP \rightarrow Det Nominal$   
 $Nominal \rightarrow book | flight$   
 $Nominal \rightarrow Nominal PP$   
 $VP \rightarrow book | prefer$   
 $VP \rightarrow Verb NP$   
 $VP \rightarrow X2 PP$   
 $X2 \rightarrow Verb NP$   
 $VP \rightarrow Verb PP$   
 $VP \rightarrow VP PP$   
 $PP \rightarrow Prep. NP$   
 $Det \rightarrow that | the | a$   
 $Noun \rightarrow book | flight$   
 $Verb \rightarrow book | prefer$   
 $Prop-N \rightarrow Houston | NWA$   
 $Aux \rightarrow does$   
 $Prep. \rightarrow from | to | through$

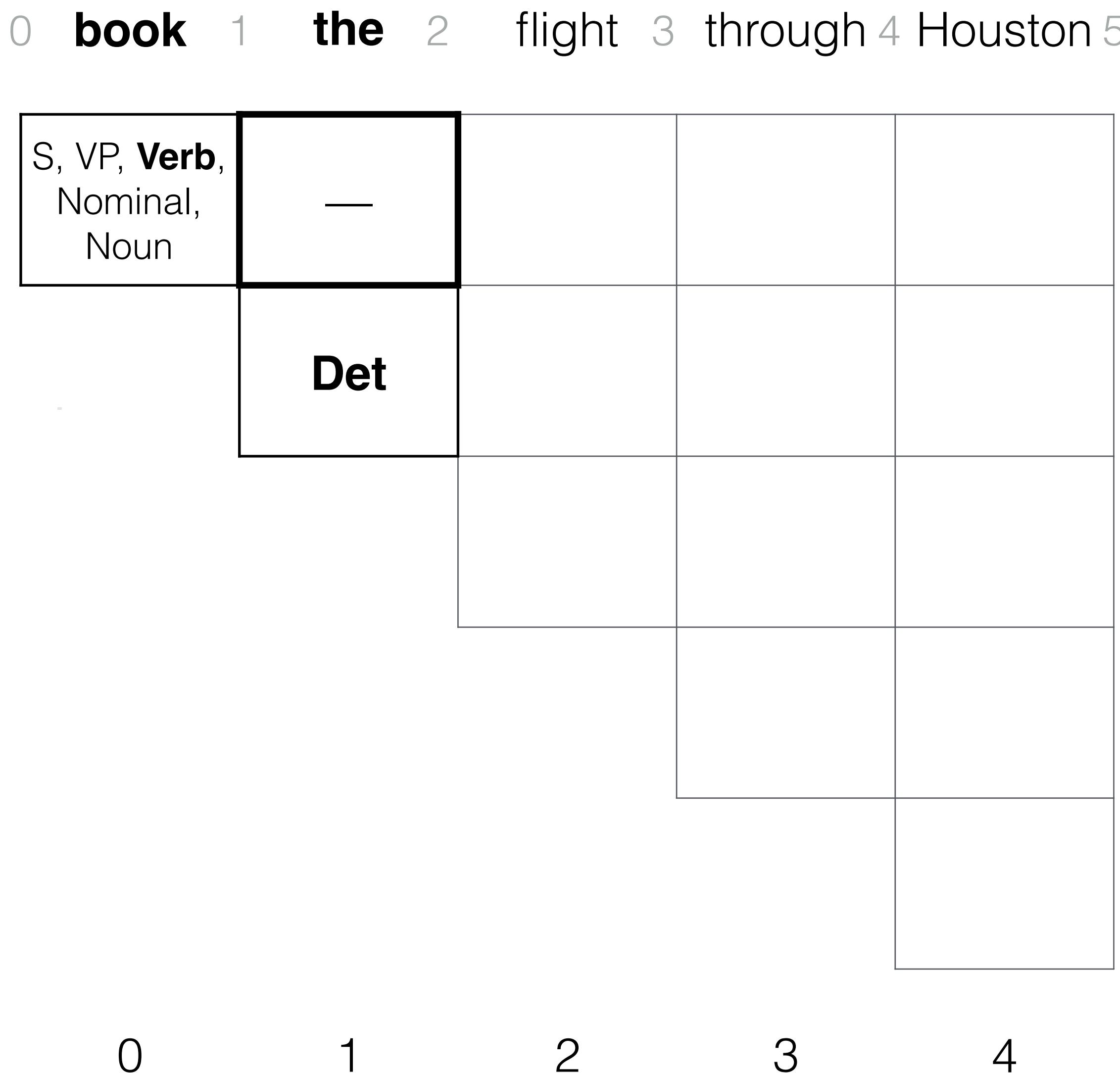
0 **book** 1 **the** 2 flight 3 through 4 Houston 5



S → NP VP	
S → book   prefer	
S → Verb NP	
S → X2 PP	
S → Verb PP	0
S → VP PP	
NP → TWA   Houston	
NP → Det Nominal	
Nominal → book   flight	1
Nominal → Nominal PP	
VP → book   prefer	
VP → Verb NP	
VP → X2 PP	2
X2 → Verb NP	
VP → Verb PP	
VP → VP PP	3
PP → Prep. NP	
Det → that   the   a	
Noun → book   flight	4
Verb → book   prefer	
Prop-N → Houston   NWA	
Aux → does	
Prep. → from   to   through	



S → NP VP	
S → book   prefer	
S → Verb NP	
S → X2 PP	
S → Verb PP	0
S → VP PP	
NP → TWA   Houston	
NP → Det Nominal	
Nominal → book   flight	1
Nominal → Nominal PP	
VP → book   prefer	
VP → Verb NP	
VP → X2 PP	2
X2 → Verb NP	
VP → Verb PP	
VP → VP PP	3
PP → Prep. NP	
Det → that   the   a	
Noun → book   flight	4
Verb → book   prefer	
Prop-N → Houston   NWA	
Aux → does	
Prep. → from   to   through	



$S \rightarrow NP VP$   
 $S \rightarrow book \mid prefer$   
 $S \rightarrow Verb NP$   
 $S \rightarrow X2 PP$   
 $S \rightarrow Verb PP$   
 $S \rightarrow VP PP$   
 $NP \rightarrow TWA \mid Houston$   
 $NP \rightarrow Det Nominal$   
 $Nominal \rightarrow book \mid flight$   
 $Nominal \rightarrow Nominal PP$   
 $VP \rightarrow book \mid prefer$   
 $VP \rightarrow Verb NP$   
 $VP \rightarrow X2 PP$   
 $X2 \rightarrow Verb NP$   
 $VP \rightarrow Verb PP$   
 $VP \rightarrow VP PP$   
 $PP \rightarrow Prep. NP$   
 $Det \rightarrow that \mid the \mid a$   
 $Noun \rightarrow book \mid flight$   
 $Verb \rightarrow book \mid prefer$   
 $Prop-N \rightarrow Houston \mid NWA$   
 $Aux \rightarrow does$   
 $Prep. \rightarrow from \mid to \mid through$

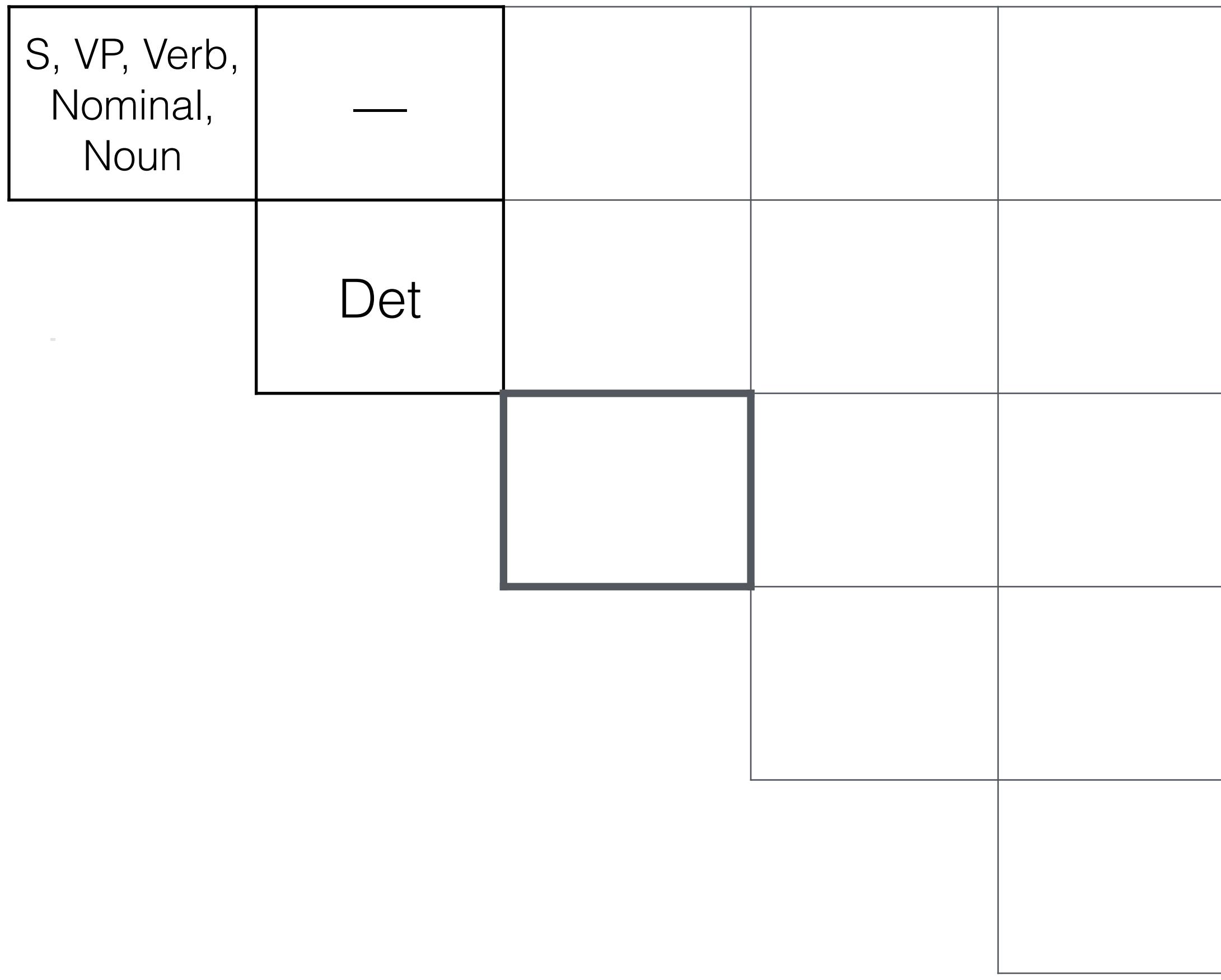
0 **book** 1 **the** 2 flight 3 through 4 Houston 5

0	<b>book</b>	1	<b>the</b>	2 flight 3 through 4 Houston 5
1	S, VP, Verb, Nominal, Noun	—		
2		Det		
3				
4				

0 1 2 3 4

S → NP VP	
S → book   prefer	
S → Verb NP	
S → X2 PP	0
S → Verb PP	
S → VP PP	
NP → TWA   Houston	
NP → Det Nominal	1
Nominal → book   flight	
Nominal → Nominal PP	
VP → book   prefer	
VP → Verb NP	
VP → X2 PP	2
X2 → Verb NP	
VP → Verb PP	
VP → VP PP	3
PP → Prep. NP	
Det → that   the   a	
Noun → book   flight	4
Verb → book   prefer	
Prop-N → Houston   NWA	
Aux → does	
Prep. → from   to   through	

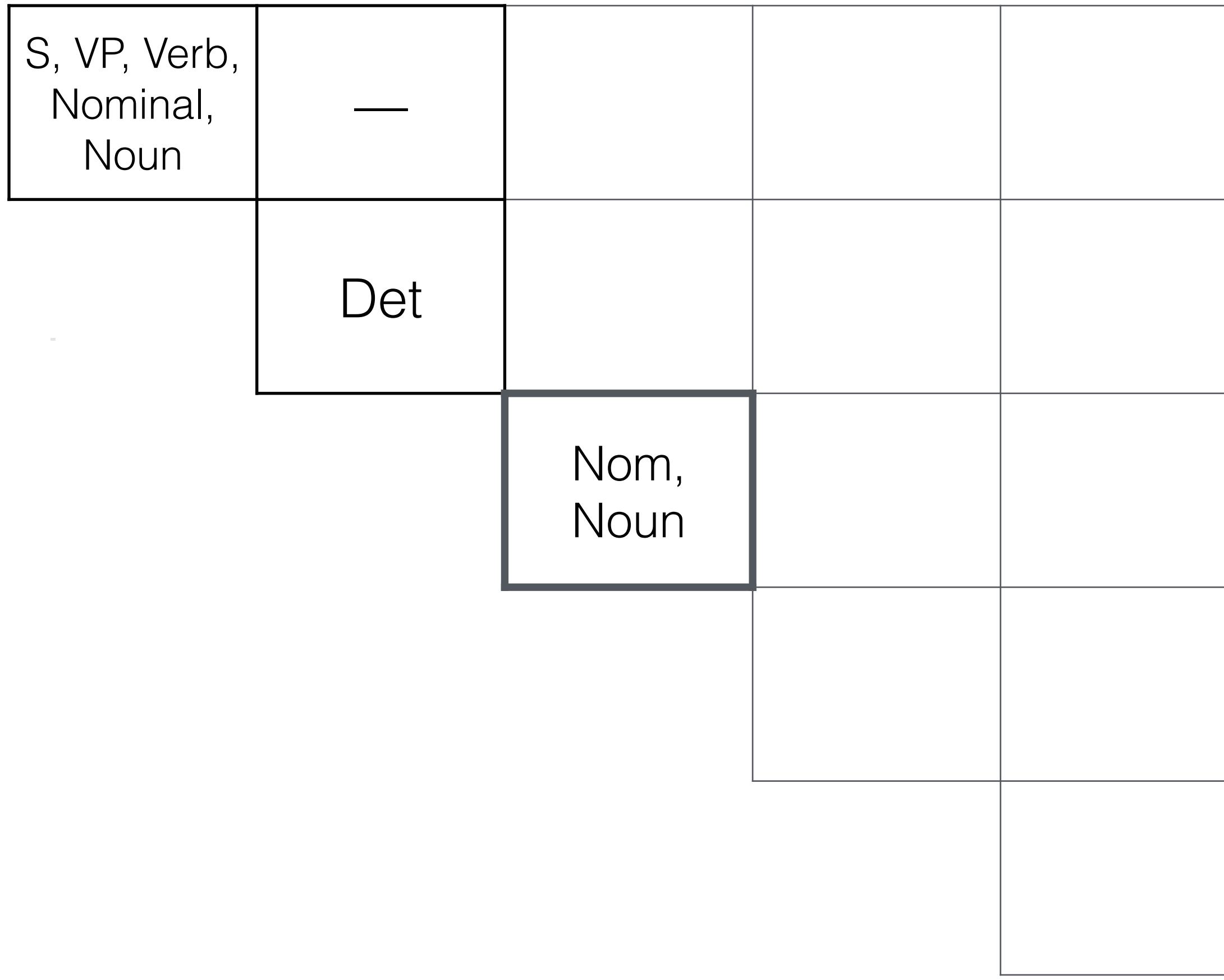
0 book 1 the 2 **flight** 3 through 4 Houston 5



0 1 2 3 4

S → NP VP	
S → book   prefer	
S → Verb NP	
S → X2 PP	0
S → Verb PP	
S → VP PP	
NP → TWA   Houston	
NP → Det Nominal	
<b>Nominal</b> → book   <b>flight</b>	1
Nominal → Nominal PP	
VP → book   prefer	
VP → Verb NP	
VP → X2 PP	2
X2 → Verb NP	
VP → Verb PP	
VP → VP PP	3
PP → Prep. NP	
Det → that   the   a	
<b>Noun</b> → book   <b>flight</b>	4
Verb → book   prefer	
Prop-N → Houston   NWA	
Aux → does	
Prep. → from   to   through	

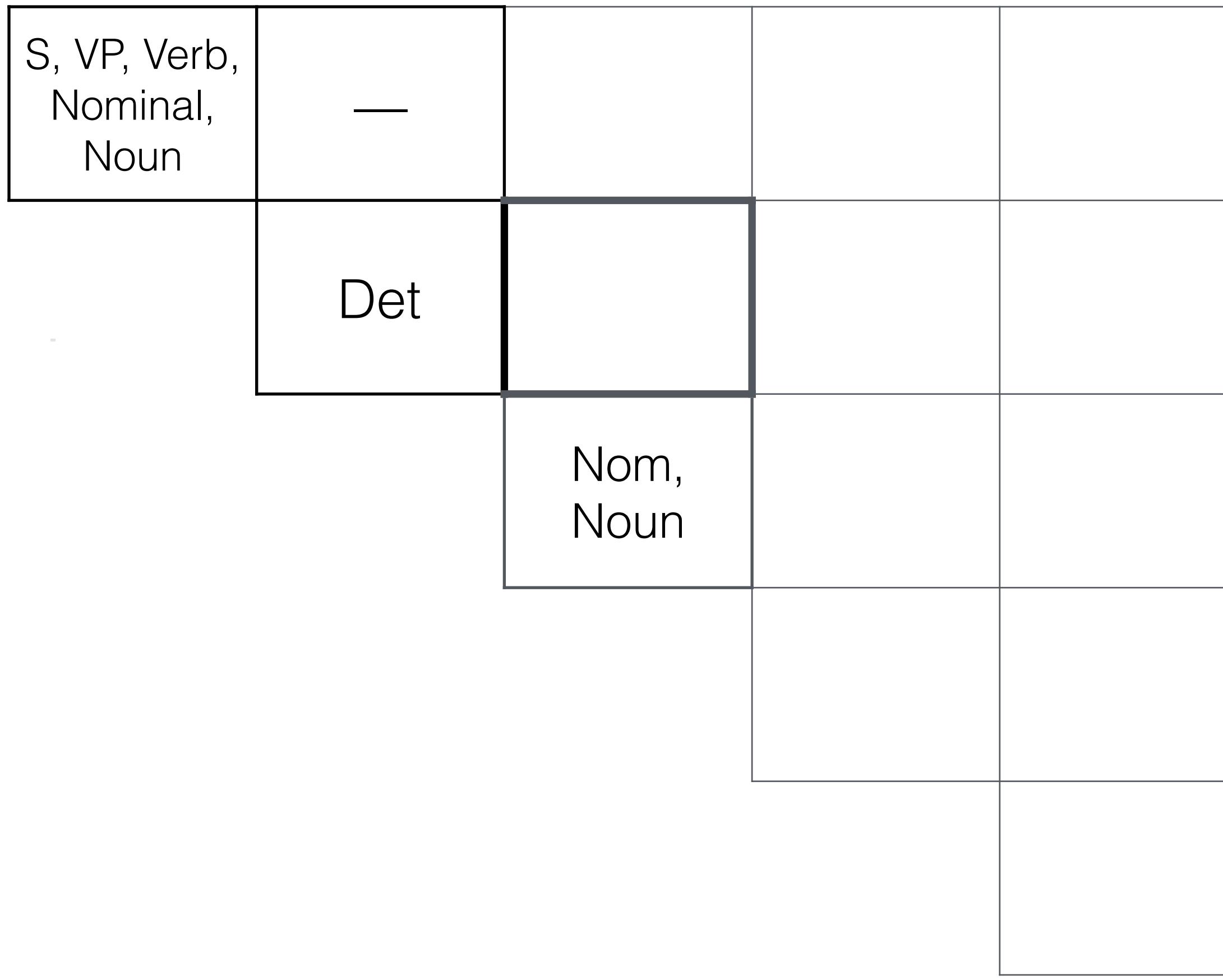
0 book 1 the 2 **flight** 3 through 4 Houston 5



0 1 2 3 4

S → NP VP	
S → book   prefer	
S → Verb NP	
S → X2 PP	0
S → Verb PP	
S → VP PP	
NP → TWA   Houston	
NP → Det Nominal	
Nominal → book   flight	1
Nominal → Nominal PP	
VP → book   prefer	
VP → Verb NP	
VP → X2 PP	2
X2 → Verb NP	
VP → Verb PP	
VP → VP PP	3
PP → Prep. NP	
Det → that   the   a	
Noun → book   flight	4
Verb → book   prefer	
Prop-N → Houston   NWA	
Aux → does	
Prep. → from   to   through	

0 book 1 **the** 2 **flight** 3 through 4 Houston 5



0 1 2 3 4

$S \rightarrow NP\ VP$

$S \rightarrow book \mid prefer$

$S \rightarrow Verb\ NP$

$S \rightarrow X2\ PP$

$S \rightarrow Verb\ PP$

$S \rightarrow VP\ PP$

$NP \rightarrow TWA \mid Houston$

**NP → Det Nominal**

$Nominal \rightarrow book \mid flight$

$Nominal \rightarrow Nominal\ PP$

$VP \rightarrow book \mid prefer$

$VP \rightarrow Verb\ NP$

$VP \rightarrow X2\ PP$

$X2 \rightarrow Verb\ NP$

$VP \rightarrow Verb\ PP$

$VP \rightarrow VP\ PP$

$PP \rightarrow Prep.\ NP$

$Det \rightarrow that \mid the \mid a$

$Noun \rightarrow book \mid flight$

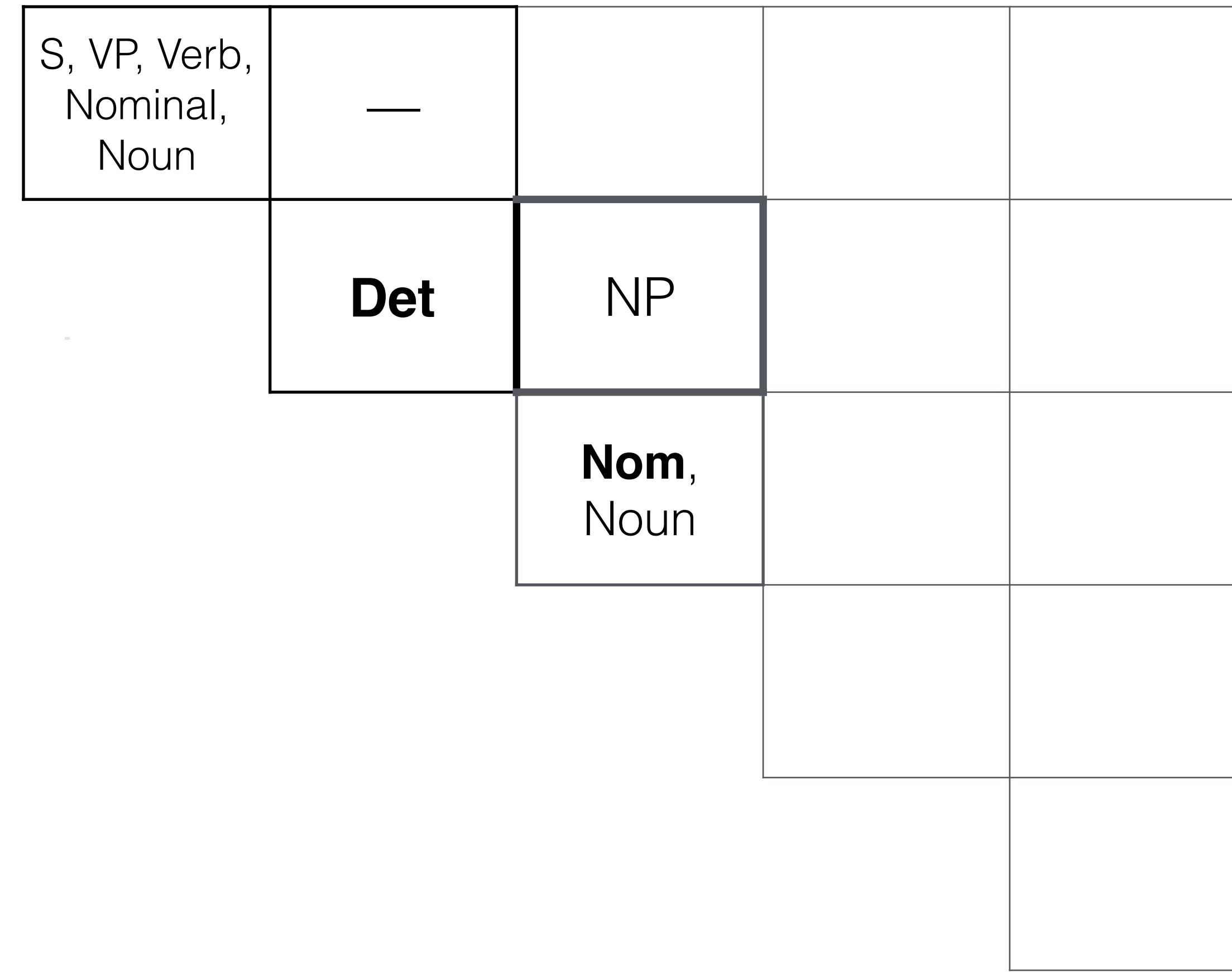
$Verb \rightarrow book \mid prefer$

$Prop-N \rightarrow Houston \mid NWA$

$Aux \rightarrow does$

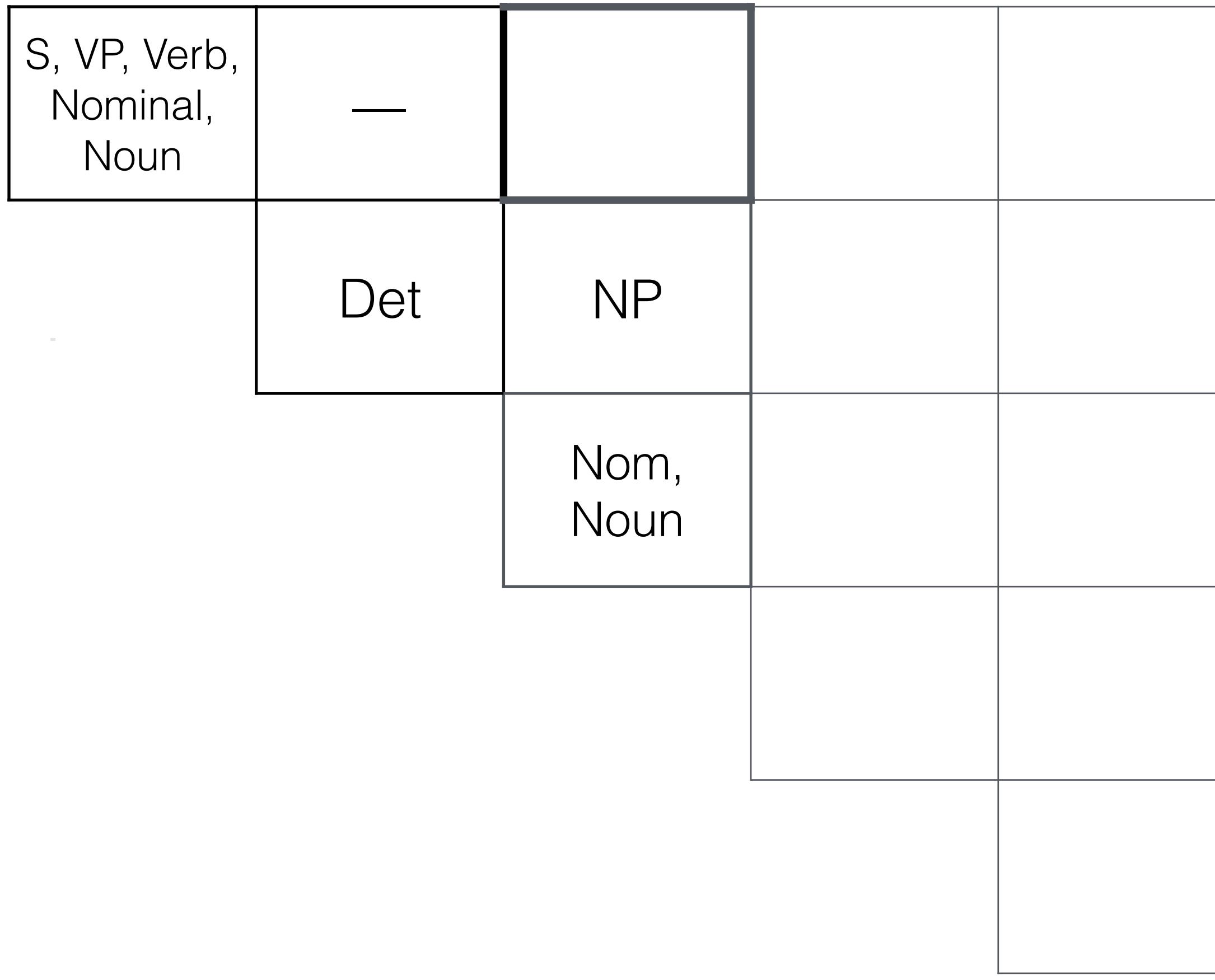
$Prep. \rightarrow from \mid to \mid through$

0 book 1 **the** 2 **flight** 3 through 4 Houston 5



S → NP VP	
S → book   prefer	
S → Verb NP	
S → X2 PP	
S → Verb PP	0
S → VP PP	
NP → TWA   Houston	
NP → Det Nominal	
Nominal → book   flight	1
Nominal → Nominal PP	
VP → book   prefer	
VP → Verb NP	
VP → X2 PP	2
X2 → Verb NP	
VP → Verb PP	
VP → VP PP	3
PP → Prep. NP	
Det → that   the   a	
Noun → book   flight	4
Verb → book   prefer	
Prop-N → Houston   NWA	
Aux → does	
Prep. → from   to   through	

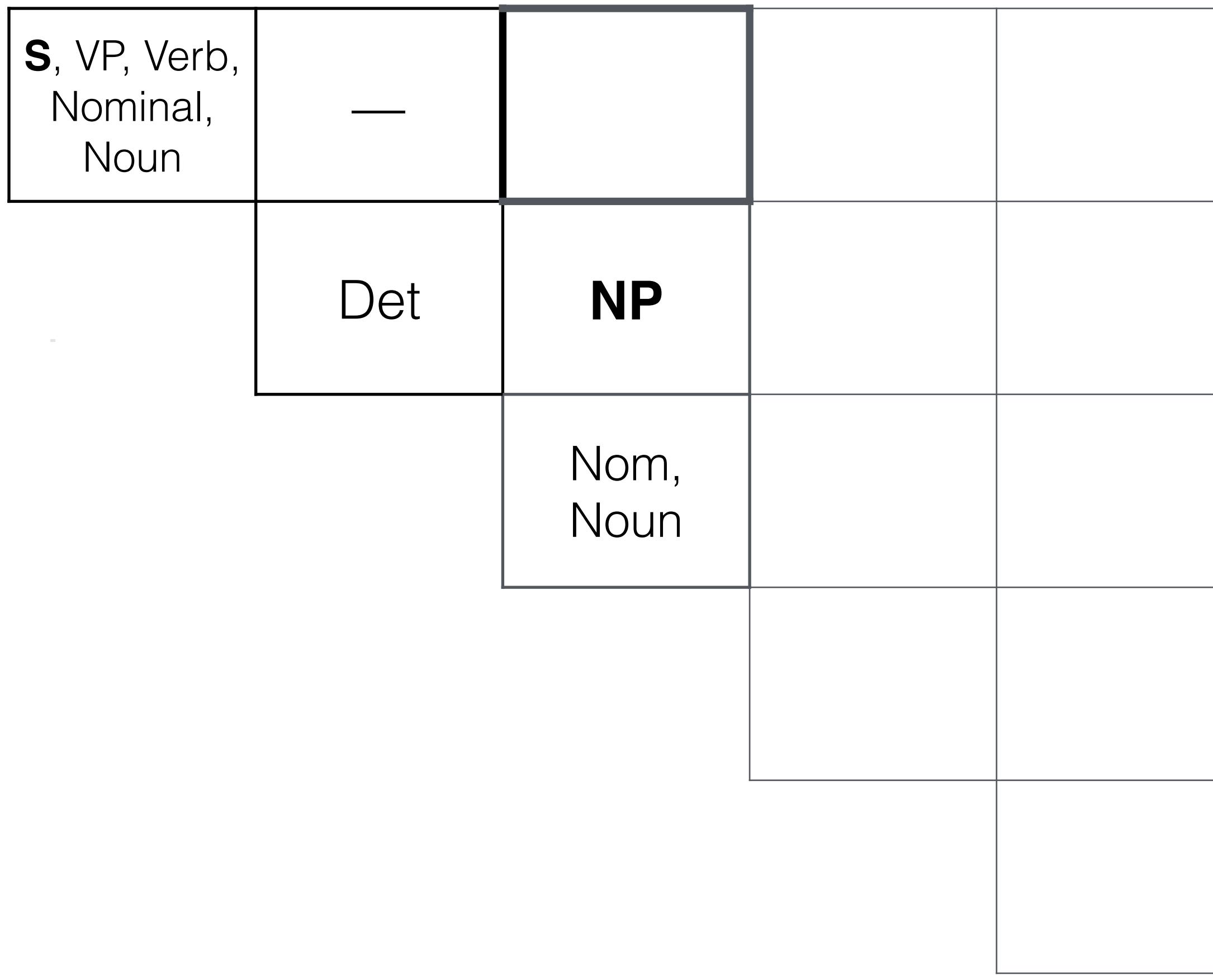
0 **book** 1 **the** 2 **flight** 3 through 4 Houston 5



0 1 2 3 4

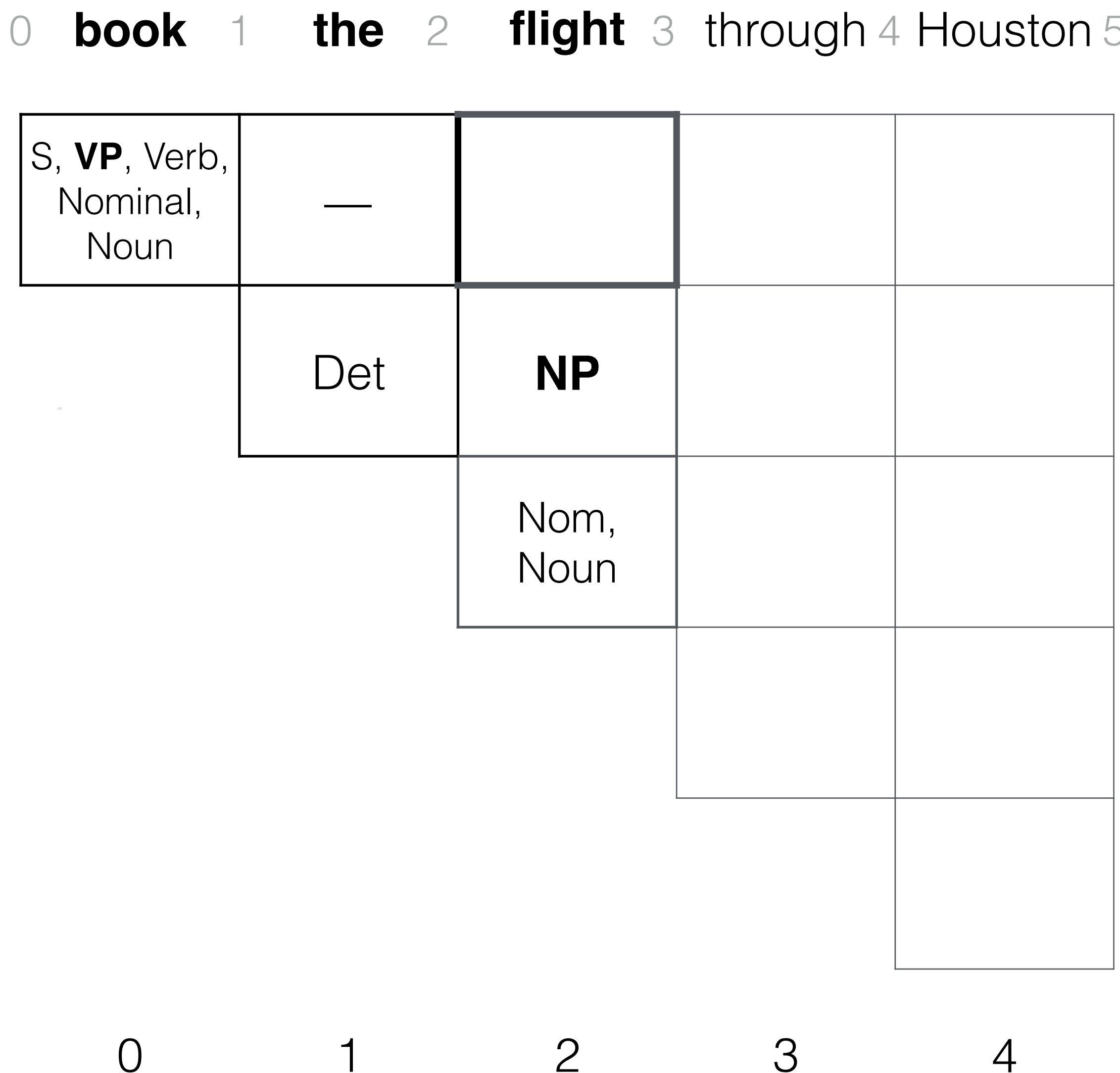
S → NP VP	
S → book   prefer	
S → Verb NP	
S → X2 PP	
S → Verb PP	0
S → VP PP	
NP → TWA   Houston	
NP → Det Nominal	
Nominal → book   flight	1
Nominal → Nominal PP	
VP → book   prefer	
VP → Verb NP	
VP → X2 PP	2
X2 → Verb NP	
VP → Verb PP	
VP → VP PP	3
PP → Prep. NP	
Det → that   the   a	
Noun → book   flight	4
Verb → book   prefer	
Prop-N → Houston   NWA	
Aux → does	
Prep. → from   to   through	

0 **book** 1 **the** 2 **flight** 3 **through** 4 **Houston** 5



0 1 2 3 4

S → NP VP	
S → book   prefer	
S → Verb NP	
S → X2 PP	
S → Verb PP	0
S → VP PP	
NP → TWA   Houston	
NP → Det Nominal	
Nominal → book   flight	1
Nominal → Nominal PP	
VP → book   prefer	
VP → Verb NP	
VP → X2 PP	2
X2 → Verb NP	
VP → Verb PP	
VP → VP PP	3
PP → Prep. NP	
Det → that   the   a	
Noun → book   flight	4
Verb → book   prefer	
Prop-N → Houston   NWA	
Aux → does	
Prep. → from   to   through	



$S \rightarrow NP VP$

$S \rightarrow book | prefer$

**$S \rightarrow Verb NP$**

$S \rightarrow X2 PP$

$S \rightarrow Verb PP$

$S \rightarrow VP PP$

$NP \rightarrow TWA | Houston$

$NP \rightarrow Det Nominal$

$Nominal \rightarrow book | flight$

$Nominal \rightarrow Nominal PP$

$VP \rightarrow book | prefer$

**$VP \rightarrow Verb NP$**

$VP \rightarrow X2 PP$

**$X2 \rightarrow Verb NP$**

$VP \rightarrow Verb PP$

$VP \rightarrow VP PP$

$PP \rightarrow Prep. NP$

$Det \rightarrow that | the | a$

$Noun \rightarrow book | flight$

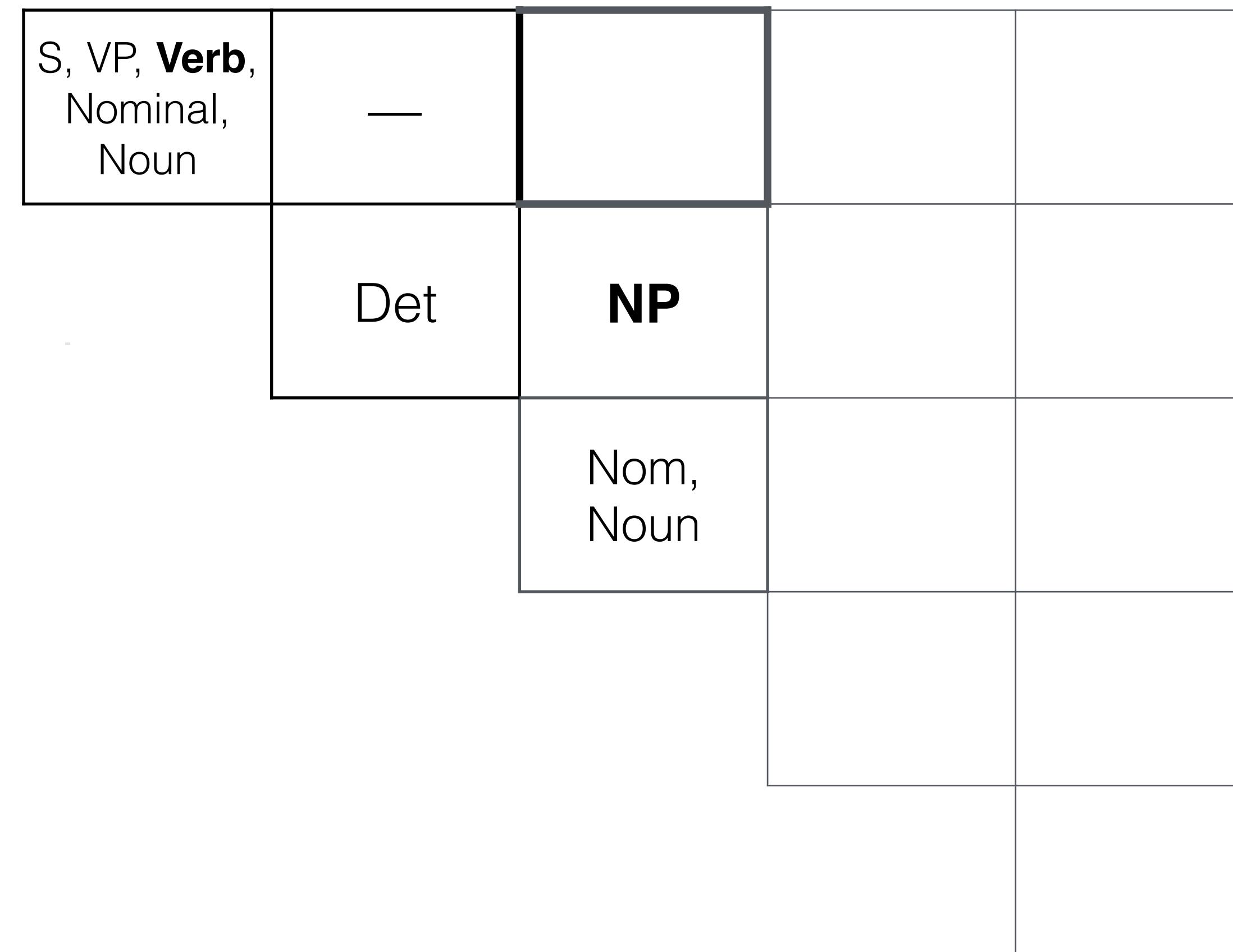
$Verb \rightarrow book | prefer$

$Prop-N \rightarrow Houston | NWA$

$Aux \rightarrow does$

$Prep. \rightarrow from | to | through$

0 **book** 1 **the** 2 **flight** 3 **through** 4 **Houston** 5



0 1 2 3 4

$S \rightarrow NP VP$

$S \rightarrow book \mid prefer$

**$S \rightarrow Verb \ NP$**

$S \rightarrow X2 \ PP$

$S \rightarrow Verb \ PP$

$S \rightarrow VP \ PP$

$NP \rightarrow TWA \mid Houston$

$NP \rightarrow Det \ Nominal$

$Nominal \rightarrow book \mid flight$

$Nominal \rightarrow Nominal \ PP$

$VP \rightarrow book \mid prefer$

**$VP \rightarrow Verb \ NP$**

$VP \rightarrow X2 \ PP$

**$X2 \rightarrow Verb \ NP$**

$VP \rightarrow Verb \ PP$

$VP \rightarrow VP \ PP$

$PP \rightarrow Prep. \ NP$

$Det \rightarrow that \mid the \mid a$

$Noun \rightarrow book \mid flight$

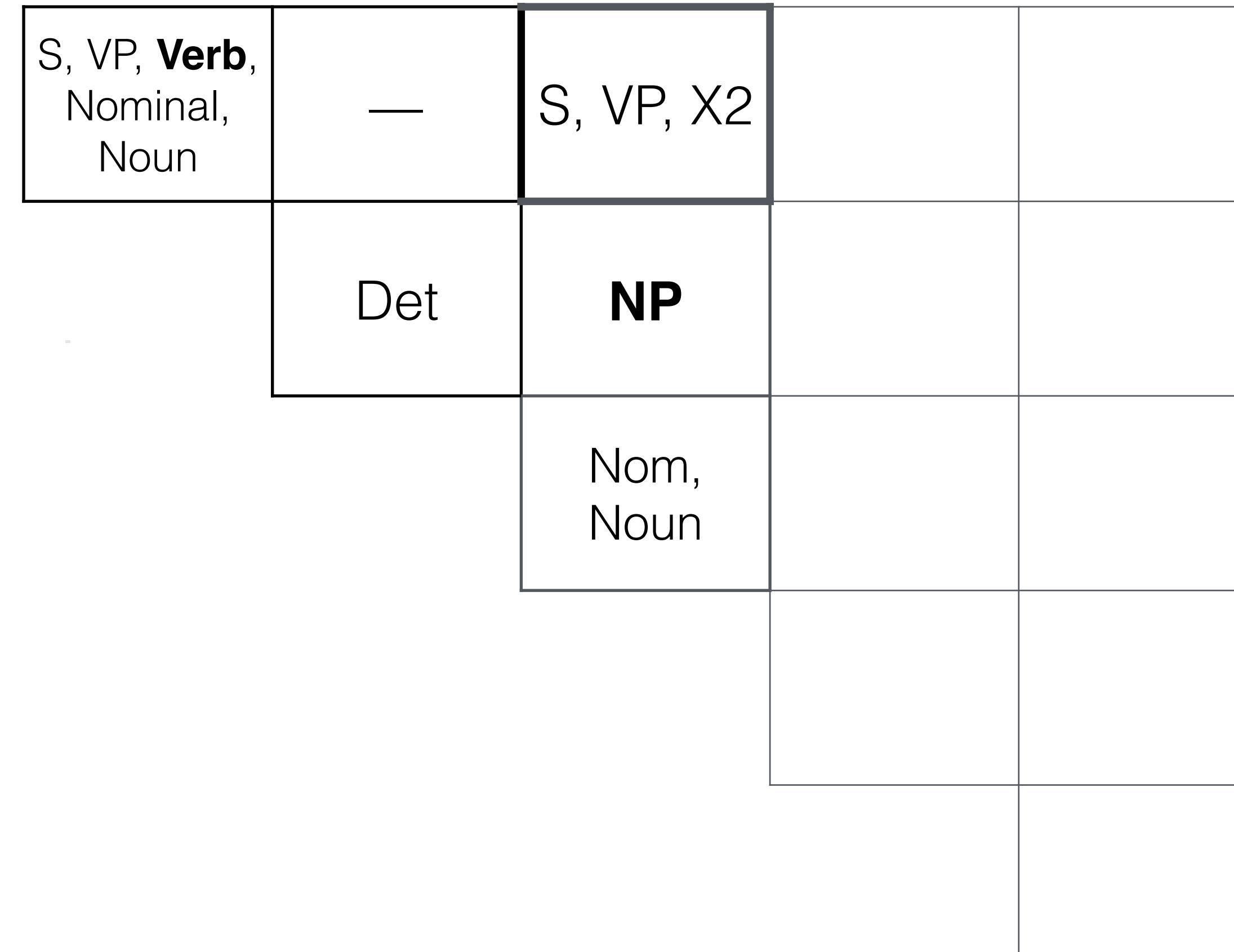
$Verb \rightarrow book \mid prefer$

$Prop-N \rightarrow Houston \mid NWA$

$Aux \rightarrow does$

$Prep. \rightarrow from \mid to \mid through$

0 **book** 1 **the** 2 **flight** 3 **through** 4 **Houston** 5



$S \rightarrow NP VP$   
 $S \rightarrow book | prefer$   
 $S \rightarrow Verb NP$   
 $S \rightarrow X2 PP$   
 $S \rightarrow Verb PP$   
 $S \rightarrow VP PP$   
 $NP \rightarrow TWA | Houston$   
 $NP \rightarrow Det Nominal$   
 $Nominal \rightarrow book | flight$   
 $Nominal \rightarrow Nominal PP$   
 $VP \rightarrow book | prefer$   
 $VP \rightarrow Verb NP$   
 $VP \rightarrow X2 PP$   
 $X2 \rightarrow Verb NP$   
 $VP \rightarrow Verb PP$   
 $VP \rightarrow VP PP$   
 $PP \rightarrow Prep. NP$   
 $Det \rightarrow that | the | a$   
 $Noun \rightarrow book | flight$   
 $Verb \rightarrow book | prefer$   
 $Prop-N \rightarrow Houston | NWA$   
 $Aux \rightarrow does$   
 $Prep. \rightarrow from | to | through$

0 book 1 the 2 flight 3 through 4 Houston 5

	S, VP, Verb, Nominal, Noun	—	S, VP, X2	S, VP
0		Det	NP	NP
1			Nom, Noun	Nom
2				Prep
3				PP
4				NP, Prop- N

0 1 2 3 4

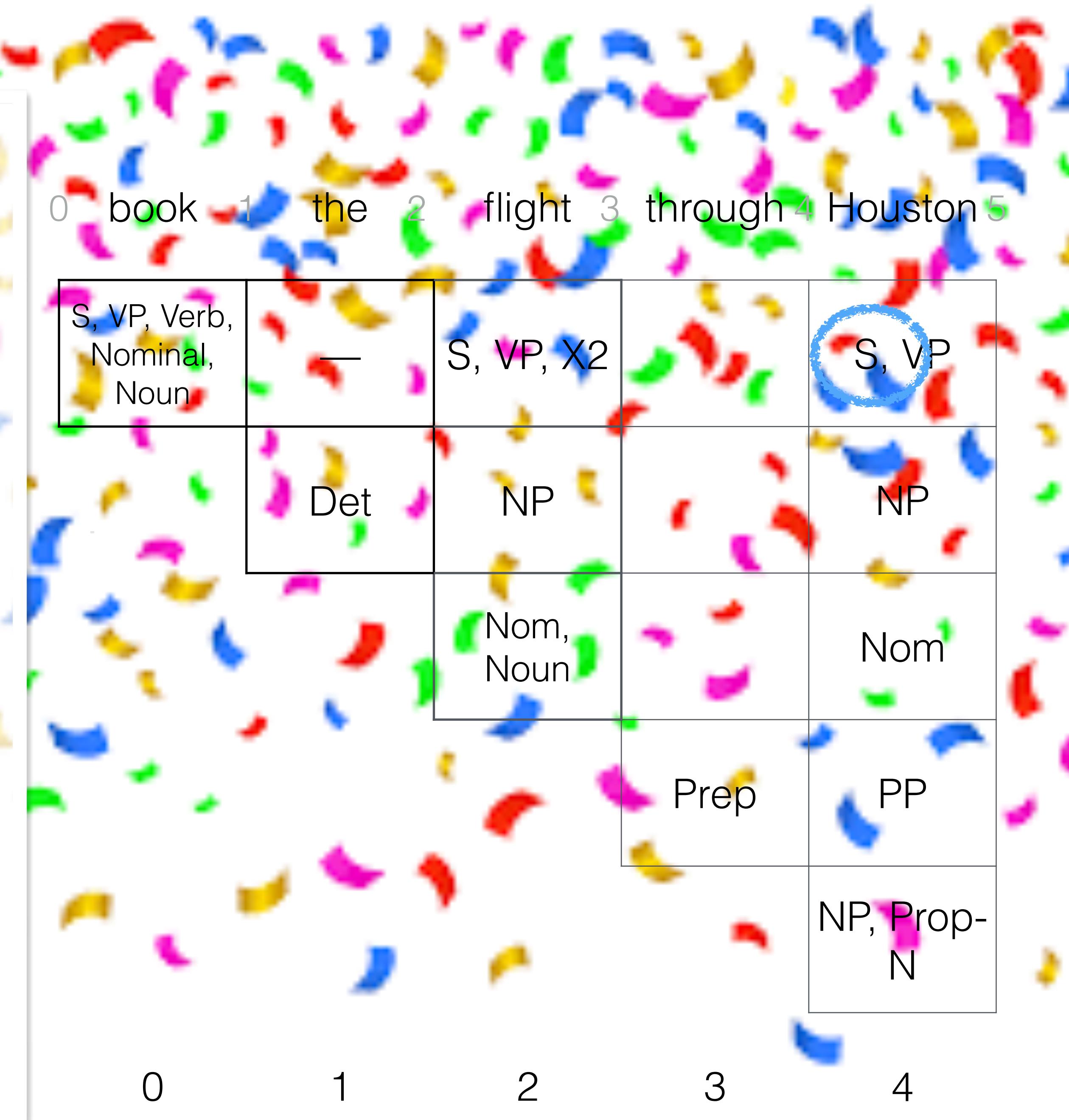
S → NP VP	
S → book   prefer	
S → Verb NP	
S → X2 PP	
S → Verb PP	0
S → VP PP	
NP → TWA   Houston	
NP → Det Nominal	
Nominal → book   flight	1
Nominal → Nominal PP	
VP → book   prefer	
VP → Verb NP	
VP → X2 PP	2
X2 → Verb NP	
VP → Verb PP	
VP → VP PP	3
PP → Prep. NP	
Det → that   the   a	
Noun → book   flight	4
Verb → book   prefer	
Prop-N → Houston   NWA	
Aux → does	
Prep. → from   to   through	

0 book 1 the 2 flight 3 through 4 Houston 5

S, VP, Verb, Nominal, Noun	—	S, VP, X2		S, VP
	Det	NP		NP
		Nom, Noun		Nom
			Prep	PP
				NP, Prop- N

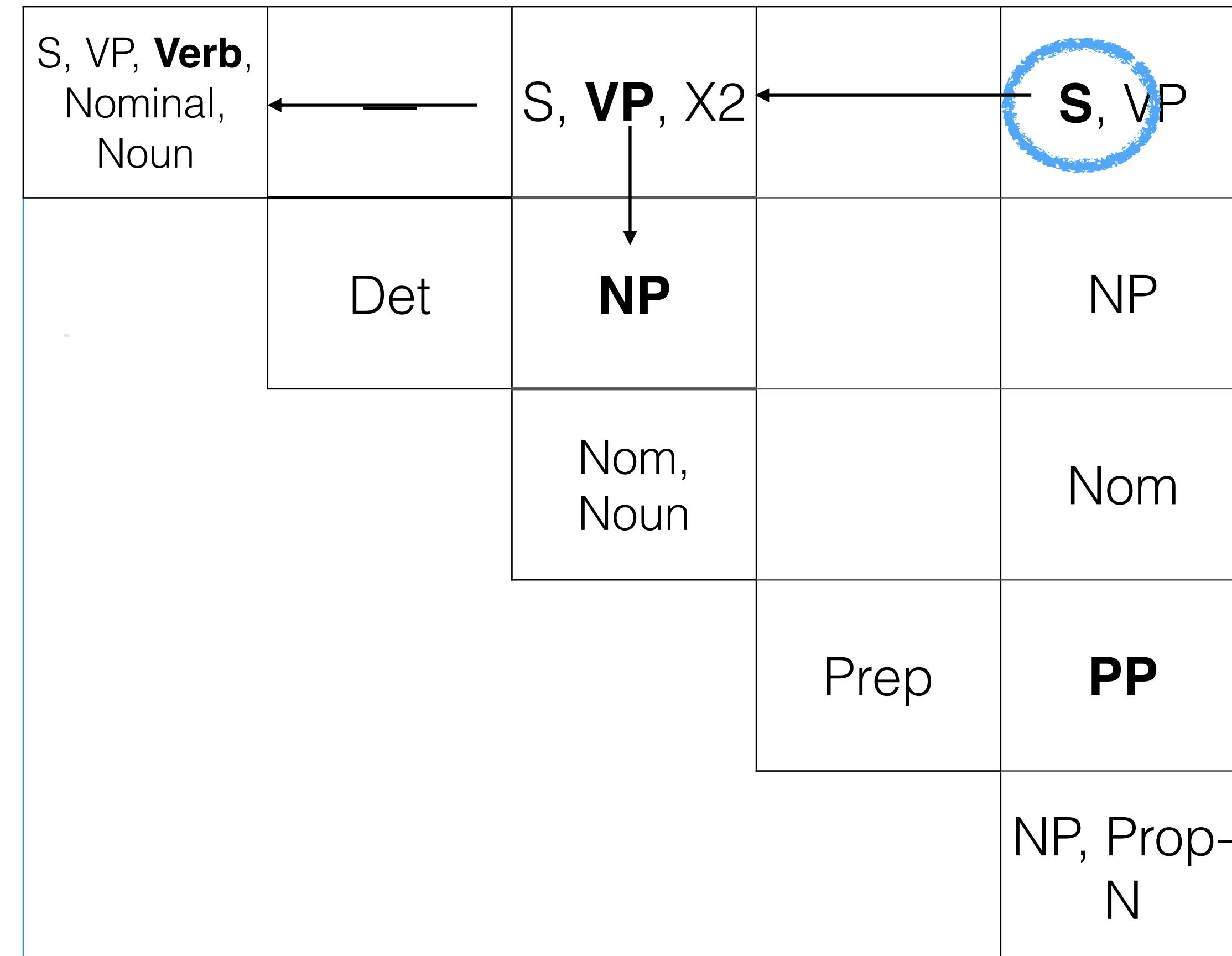
0 1 2 3 4

$S \rightarrow NP VP$   
 $S \rightarrow book | prefer$   
 $S \rightarrow Verb NP$   
 $S \rightarrow X2 PP$   
 $S \rightarrow Verb PP$   
 $S \rightarrow VP PP$   
 $NP \rightarrow TWA | Houston$   
 $NP \rightarrow Det Nominal$   
 $Nominal \rightarrow book | flight$   
 $Nominal \rightarrow Nominal PP$   
 $VP \rightarrow book | prefer$   
 $VP \rightarrow Verb NP$   
 $VP \rightarrow X2 PP$   
 $X2 \rightarrow Verb NP$   
 $VP \rightarrow Verb PP$   
 $VP \rightarrow VP PP$   
 $PP \rightarrow Prep. NP$   
 $Det \rightarrow that | the | a$   
 $Noun \rightarrow book | flight$   
 $Verb \rightarrow book | prefer$   
 $Prop-N \rightarrow Houston | NWA$   
 $Aux \rightarrow does$   
 $Prep. \rightarrow from | to | through$



S → NP VP	
S → book   prefer	
S → Verb NP	
S → X2 PP	0
S → Verb PP	
S → VP PP	
NP → TWA   Houston	
NP → Det Nominal	
Nominal → book   flight	1
Nominal → Nominal PP	
VP → book   prefer	
VP → Verb NP	
VP → X2 PP	2
X2 → Verb NP	
VP → Verb PP	
VP → VP PP	3
PP → Prep. NP	
Det → that   the   a	
Noun → book   flight	4
Verb → book   prefer	
Prop-N → Houston   NWA	
Aux → does	
Prep. → from   to   through	

0 book 1 the 2 flight 3 through 4 Houston 5

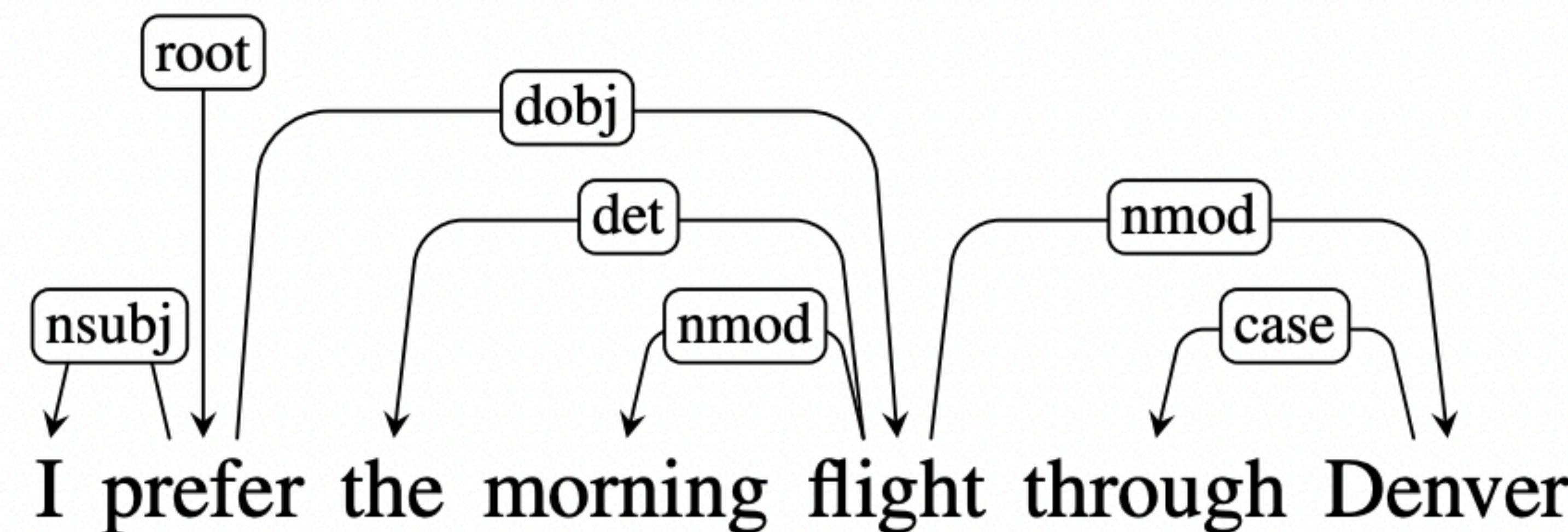


# Topics

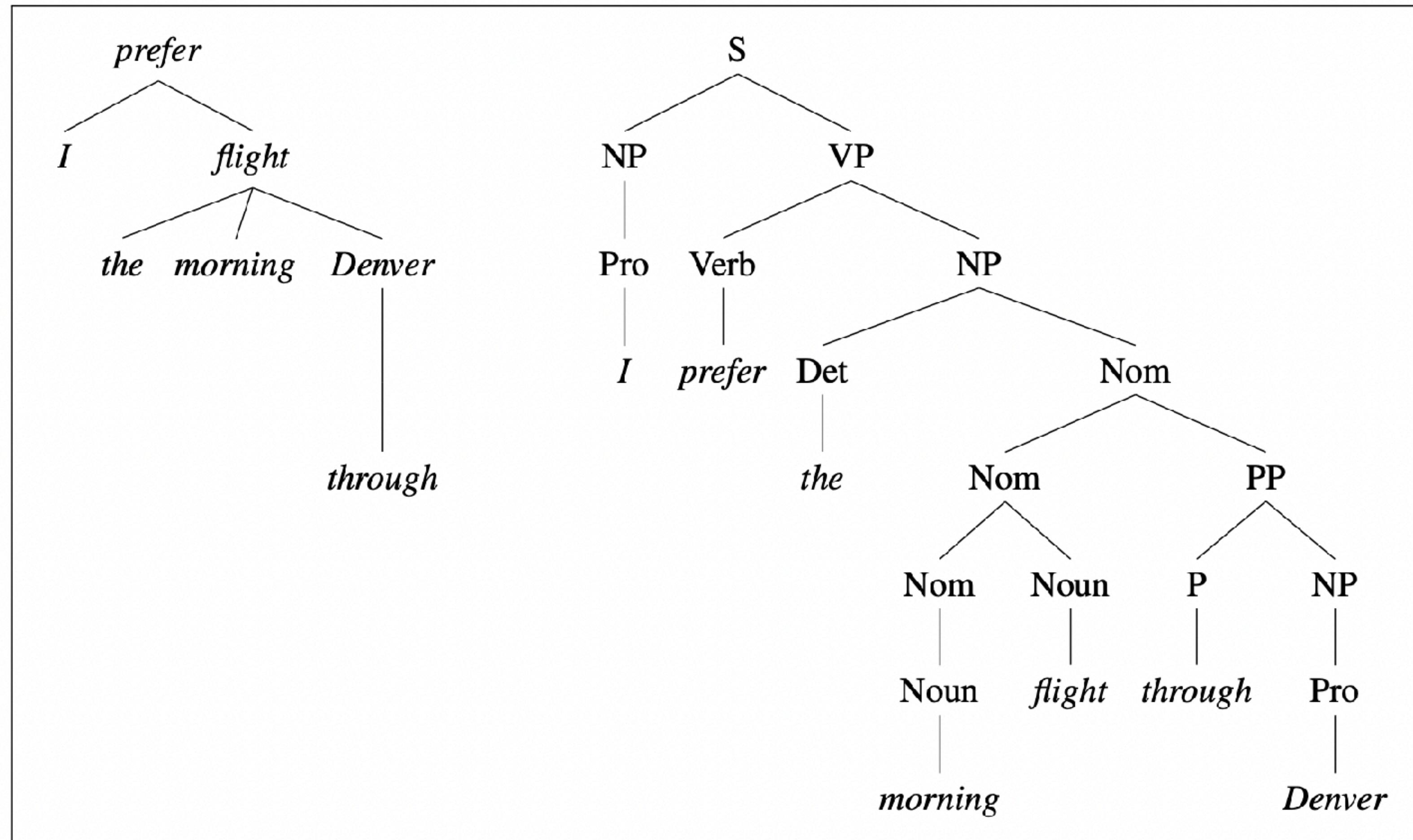
- Formal Grammars
- Constituency Parsing with the CKY Algorithm
- **Dependency Parsing with the Shift-Reduce Algorithm**

# Dependency Grammar

- Goal is to make explicit the relationships between words (*not* their substitutability)
- Typically, a verb is at the ROOT
- The core of the sentence is subject-verb-object
- Additional clauses/modifiers hang off of these



# Dependency Grammar



**Figure 14.1** Dependency and constituent analyses for *I prefer the morning flight through Denver*.

# Dependency Parsing

- Dependency parses are represented as a directed graph
  - Vertices ( $V$ ) and Arcs ( $A$ )
  - $V$  is roughly the set of words+punctuation
    - In morphologically rich languages, includes stems and affixes too
  - Typically constrained to be single-rooted trees
    - This is a simplifying assumption, and means we get some things wrong (like the CFG assumption)

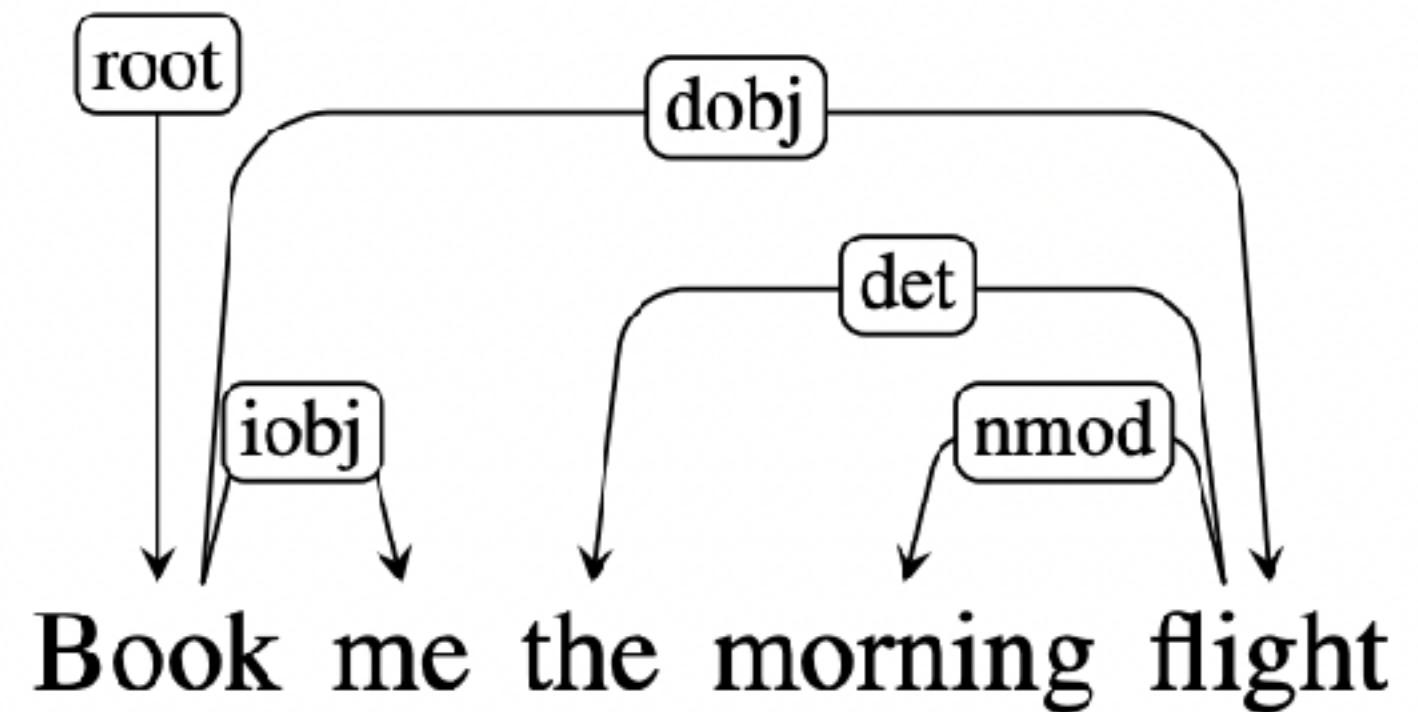
# Dependency Parsing

- Transition-Based Parser — specifically “Shift-Reduce” Algorithm
- Basic idea:
  - Progress left to right in the sentence, maintain a stack
  - At each step, either: 1) assign this word to be the head of previous word, 2) assign a previously seen word to be the head of this word, or 3) defer decision to a later step
  - Design about 1, 2, or 3 is made by a supervised classifier

# Dependency Parsing

- Three operations, each deals with top 2 elements on the stack
  - LeftArc: set stack[0] as head of stack[1]; remove stack[1] from stack
  - RightArc: set stack[1] as head of stack[0]; remove stack[0] from stack
  - Shift: Push new word onto stack
- I.e., once a word is assigned a head, it is removed
- Preconditions: ROOT can't have a parent (so can't apply LeftArc if stack = [\* , ROOT, ...])

# Dependency Parsing



# Dependency Parsing

Book me the morning flight

```
stack = [root]
```

```
buffer = [book, me, the, morning, flight]
```

```
relations = []
```



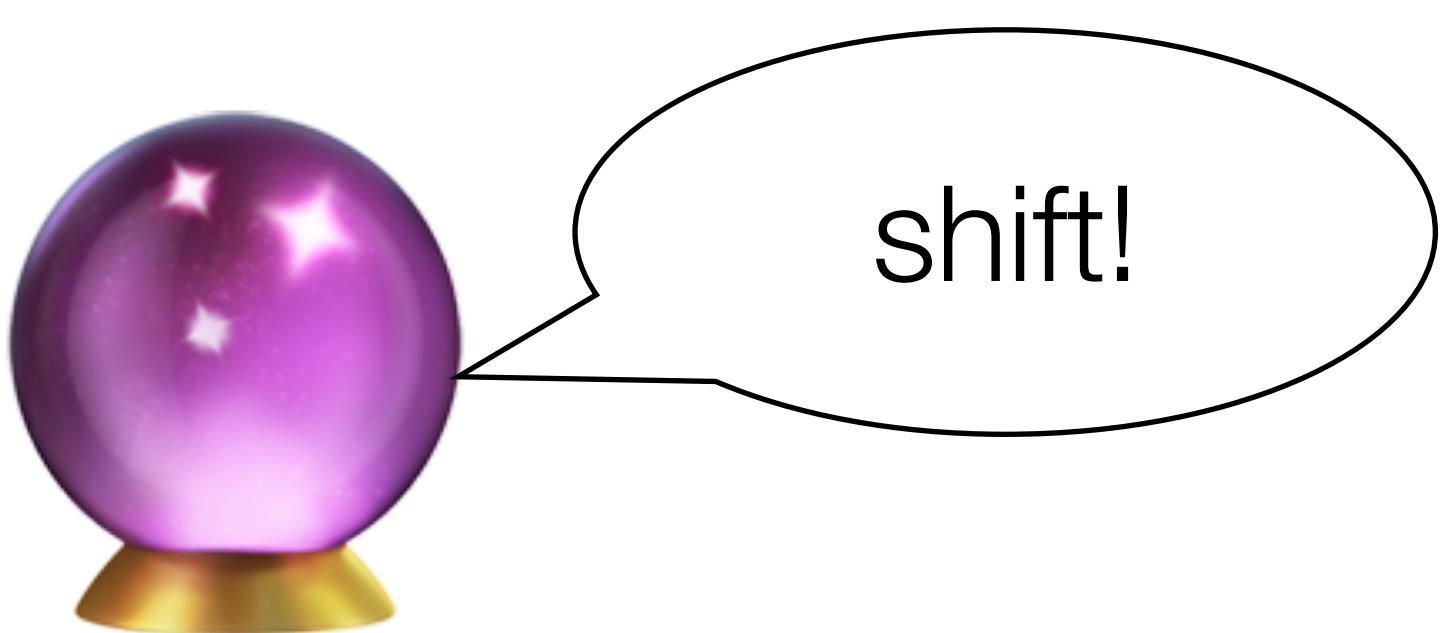
# Dependency Parsing

Book me the morning flight

stack = [root]

buffer = [book, me, the, morning, flight]

relations = [ ]



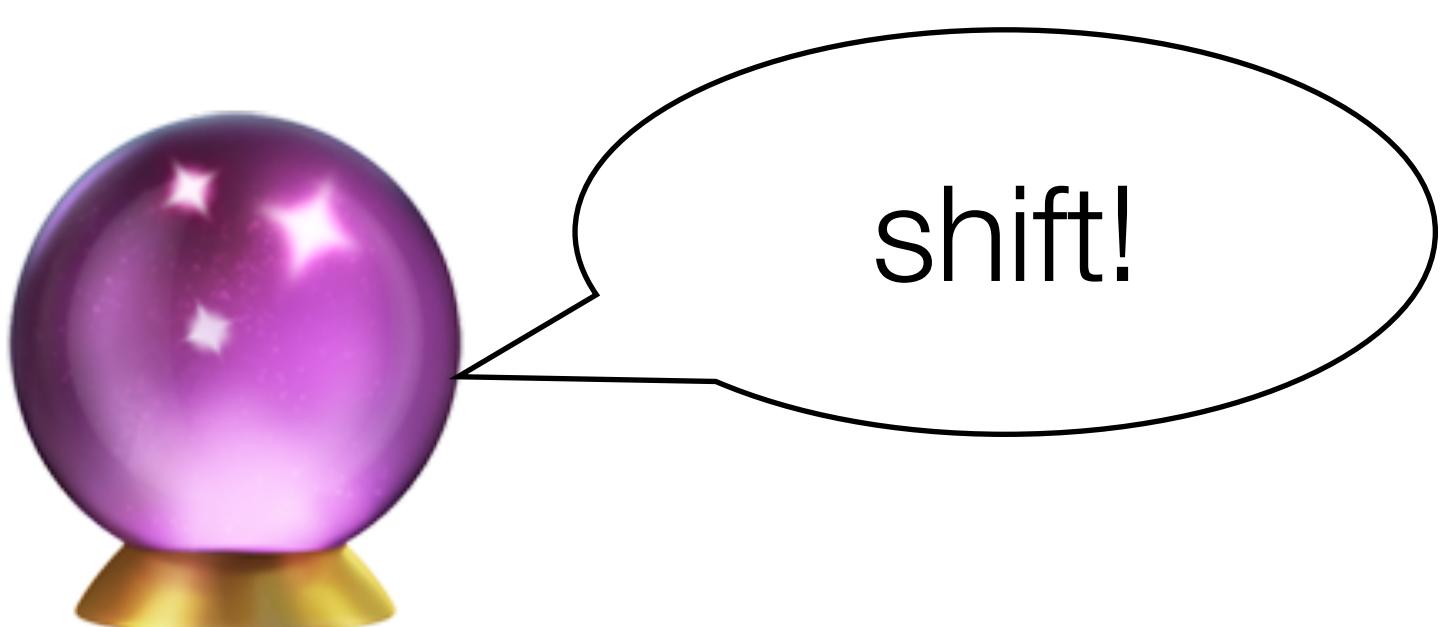
# Dependency Parsing

Book me the morning flight

stack = [root]

buffer = [book, me, the, morning, flight]

relations = []



Step 0: when there are less than 2 things on the stack, always shift.

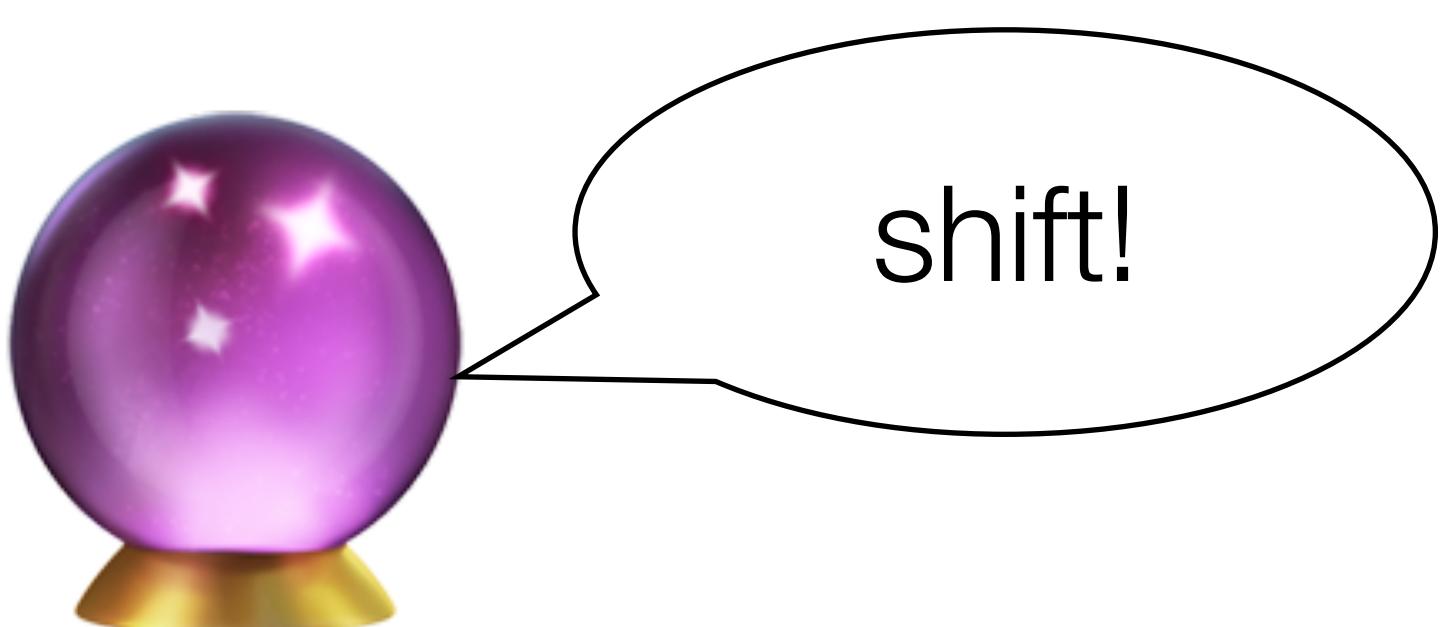
# Dependency Parsing

Book me the morning flight

stack = [root]

buffer = [book, me, the, morning, flight]

relations = []



Step 0: when there are less than 2 things on the stack, always shift.

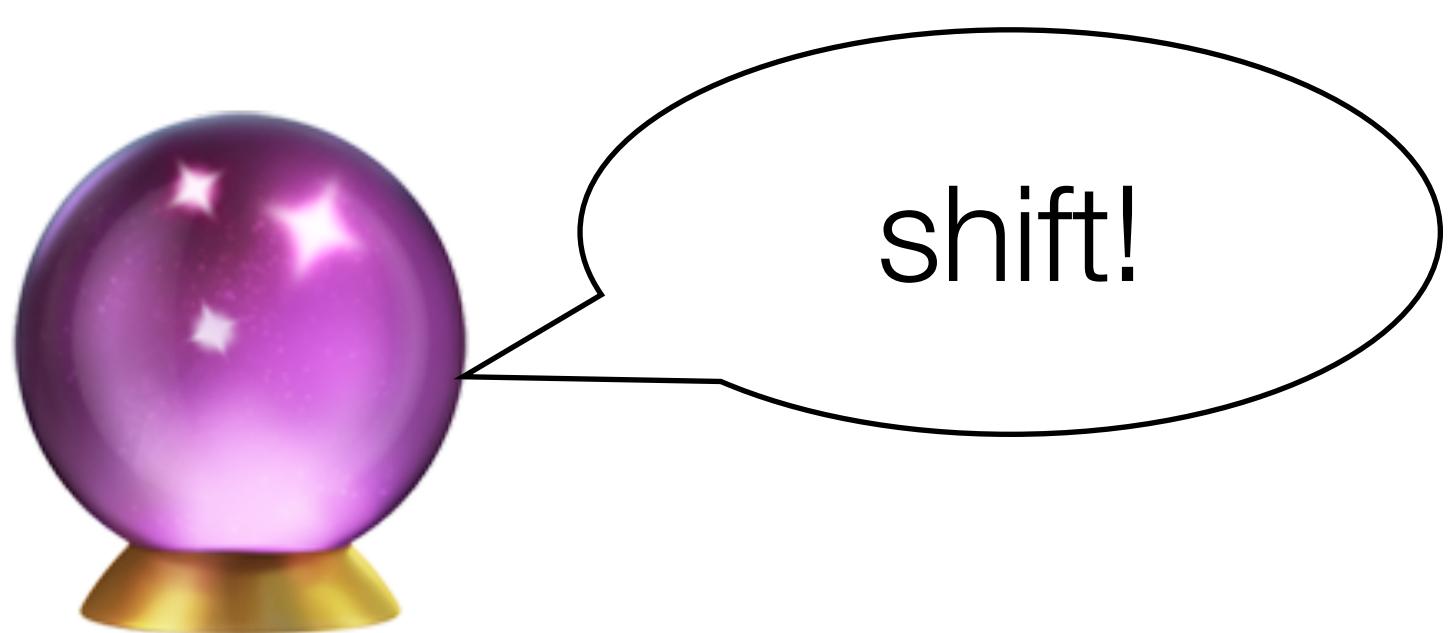
# Dependency Parsing

Book me the morning flight

stack = [book, root]

buffer = [me, the, morning, flight]

relations = [ ]



Step 0: when there are less than 2 things on the stack, always shift.

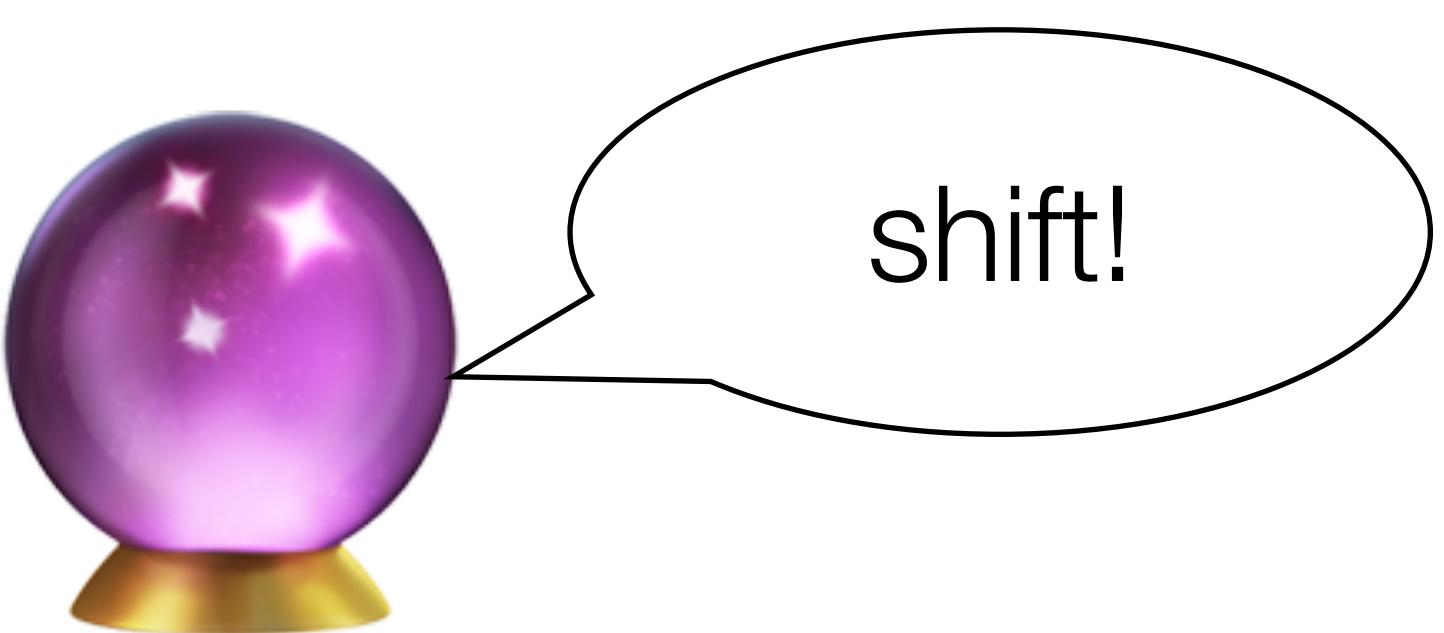
# Dependency Parsing

Book me the morning flight

```
stack = [book, root]
```

```
buffer = [me, the, morning, flight]
```

```
relations = [ ]
```



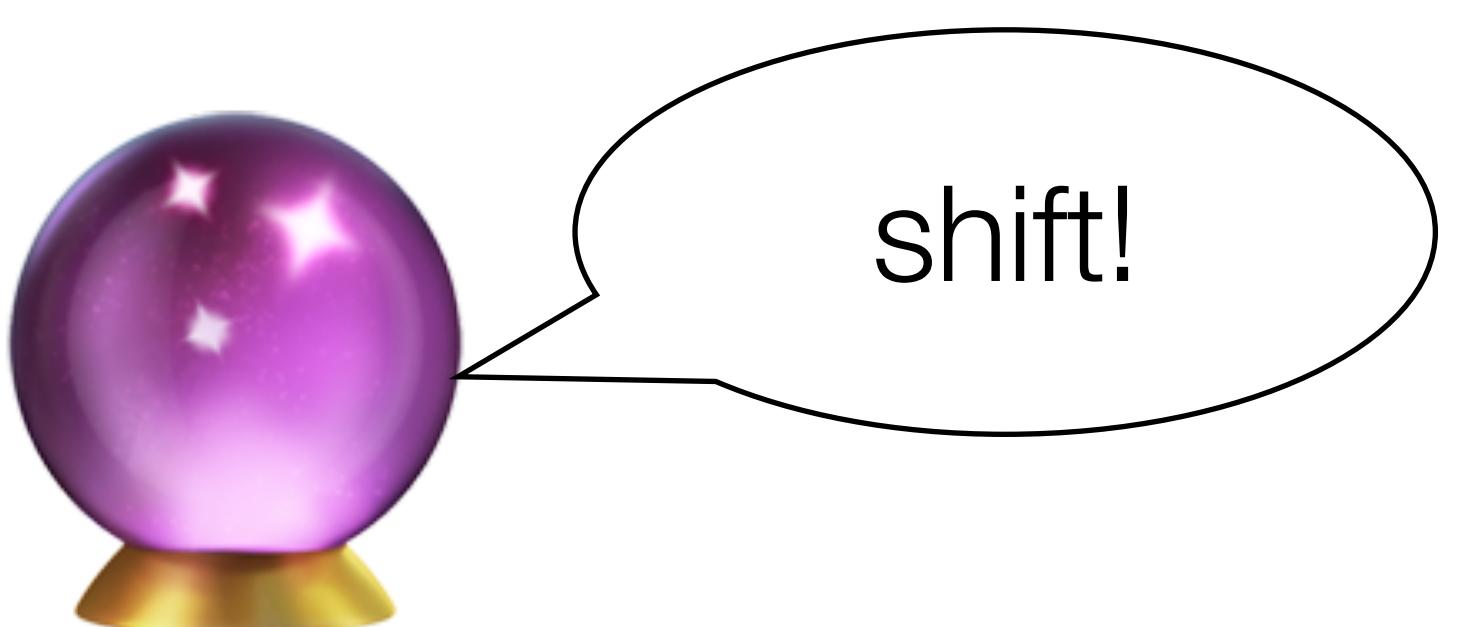
# Dependency Parsing

Book me the morning flight

```
stack = [me, book, root]
```

```
buffer = [the, morning, flight]
```

```
relations = []
```



Step 1

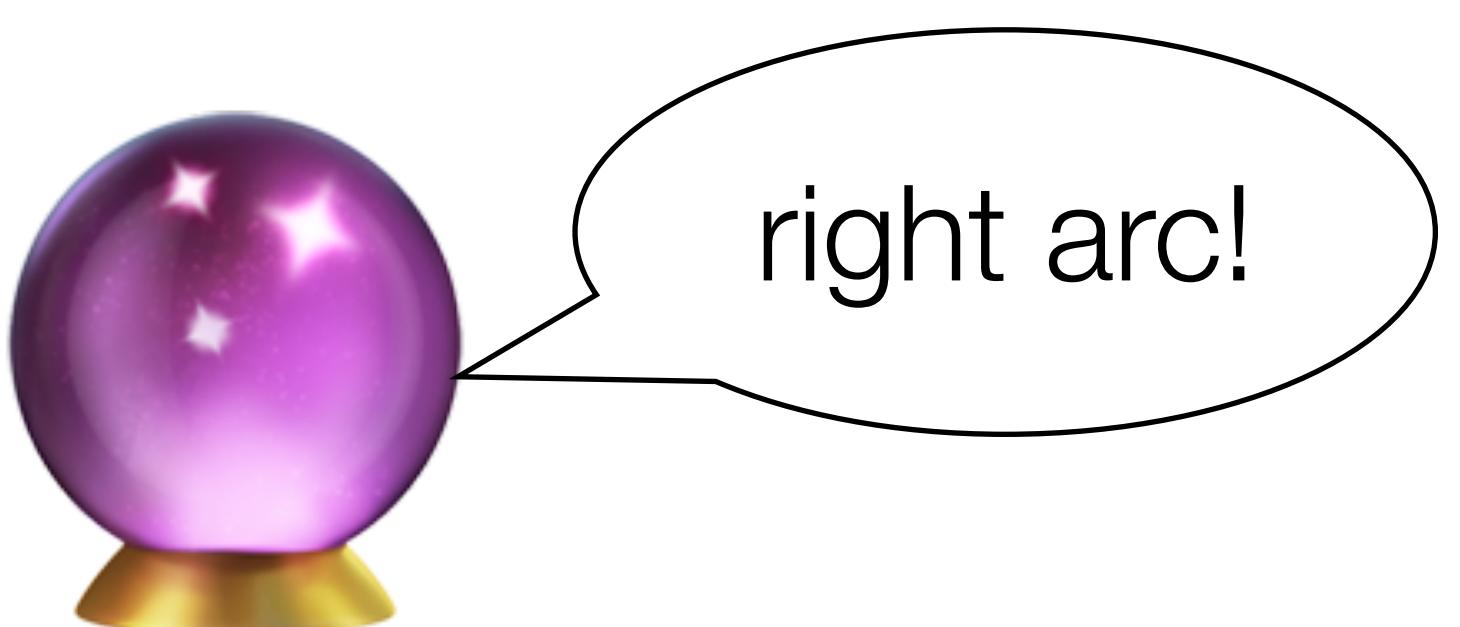
# Dependency Parsing

Book me the morning flight

```
stack = [me, book, root]
```

```
buffer = [the, morning, flight]
```

```
relations = [ ]
```

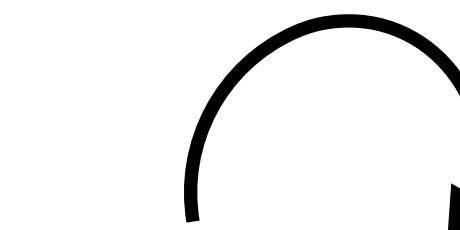
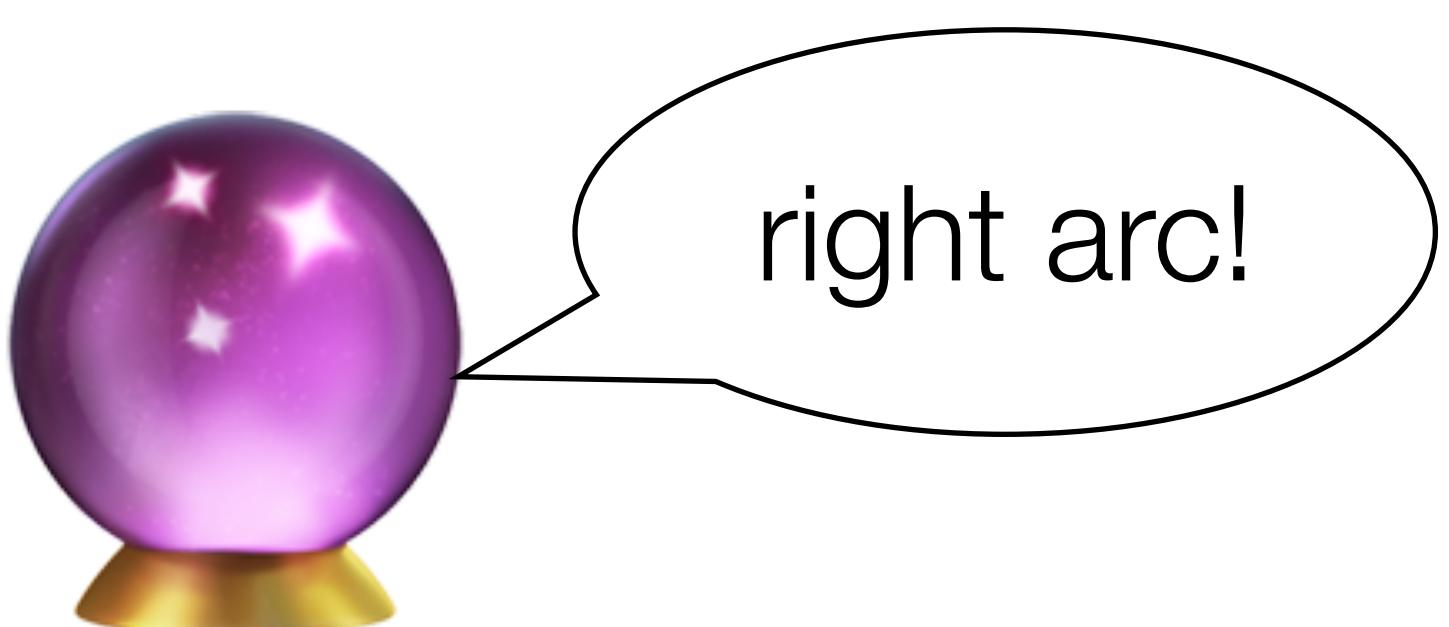


# Dependency Parsing

stack = [book, root]

buffer = [the, morning, flight]

relations = [book->me]



Book me the morning flight

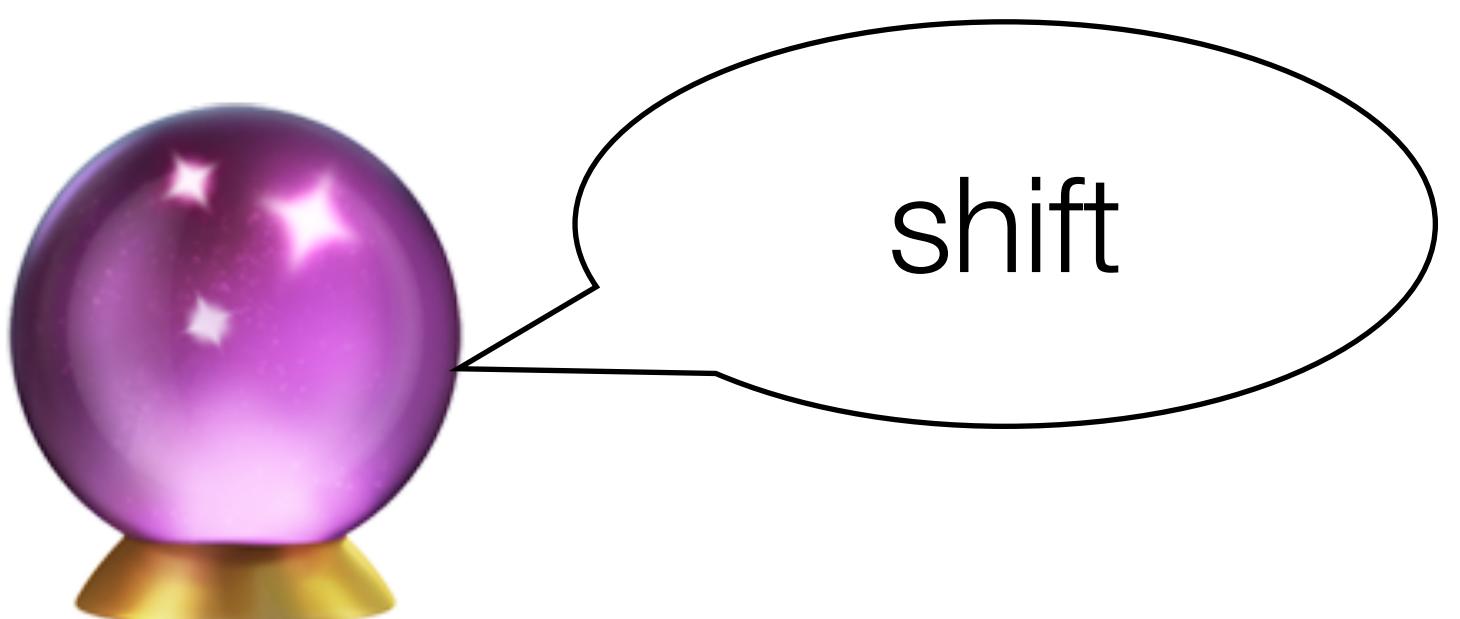
Step 2: assign book as  
head of me and remove  
me from stack

# Dependency Parsing

stack = [the, book, root]

buffer = [morning, flight]

relations = [book->me]



Book me the morning flight

Step 3

# Dependency Parsing

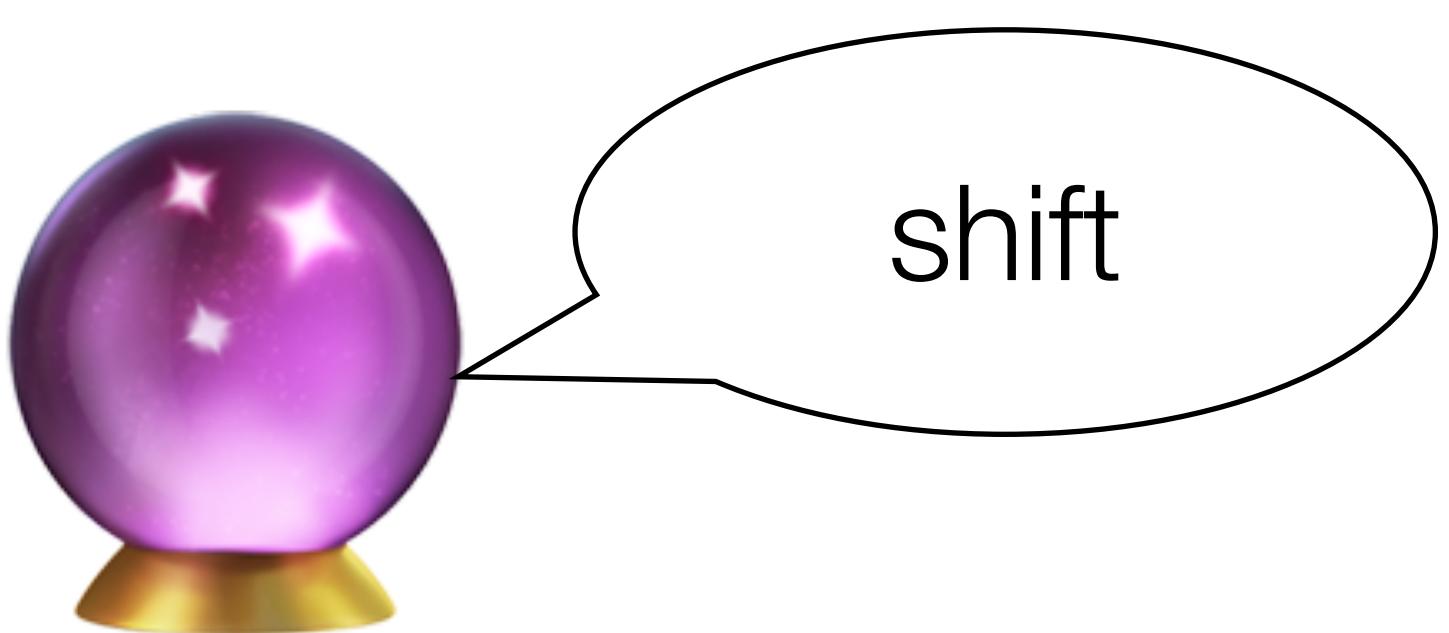
stack = [morning, the, book, root]

buffer = [flight]

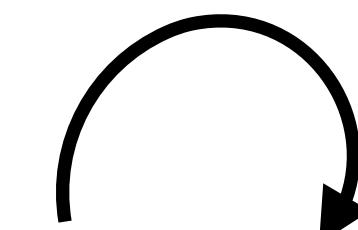
relations = [book->me]

Book me the morning flight  
root

Step 4



# Dependency Parsing

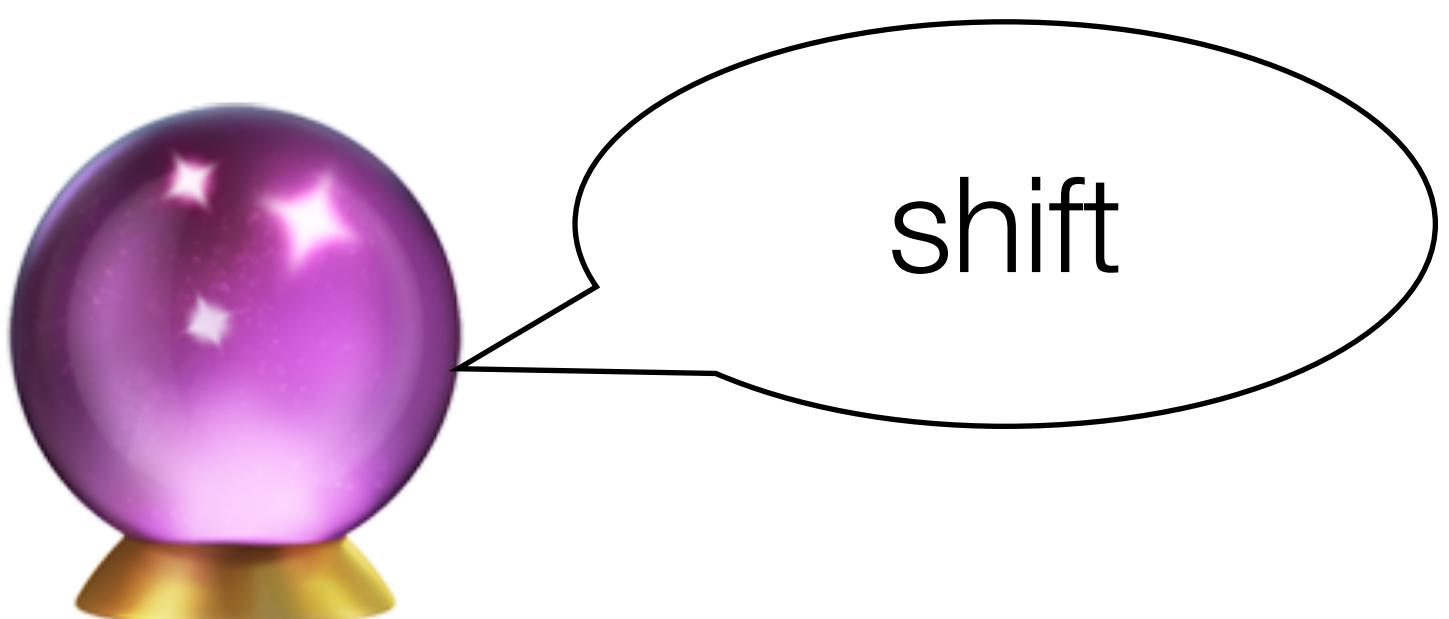


Book me the morning flight

stack = [flight, morning, the, book, root]

buffer = []

relations = [book->me]



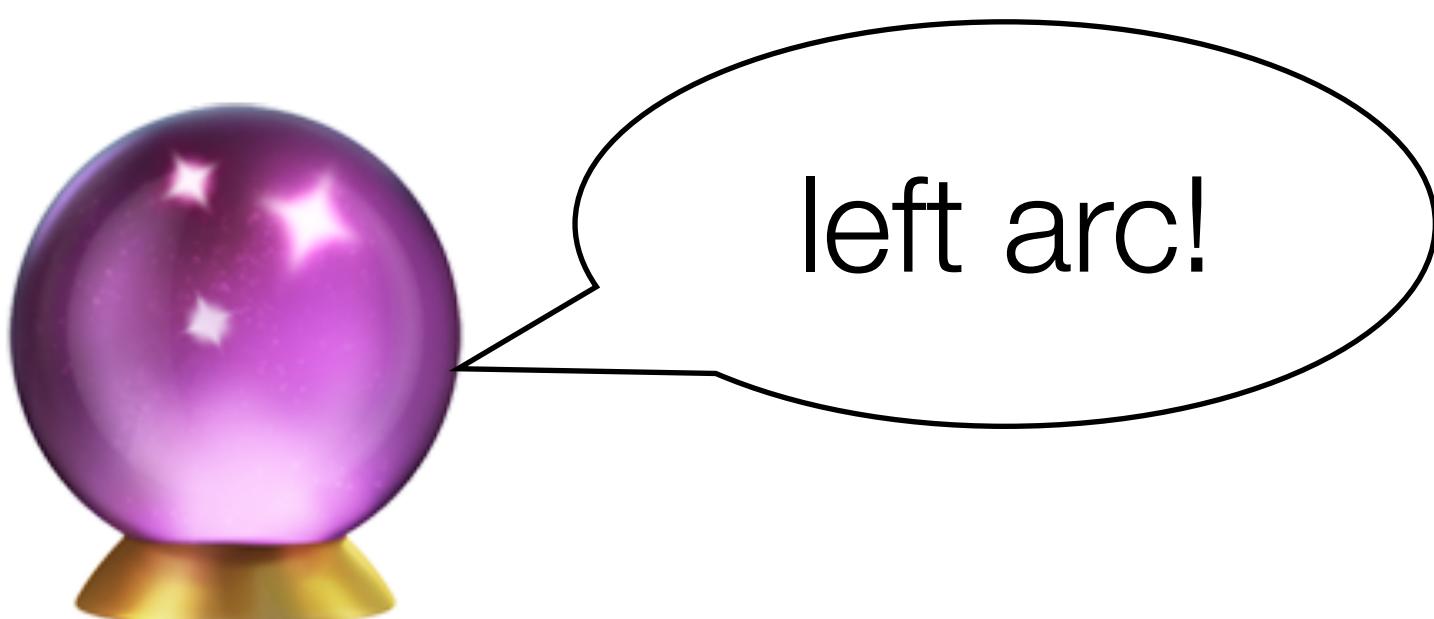
Step 5

# Dependency Parsing

```
stack = [flight, the, book, root]
```

```
buffer = []
```

```
relations = [book->me, flight->morning]
```



Book me the morning flight

```
root
```

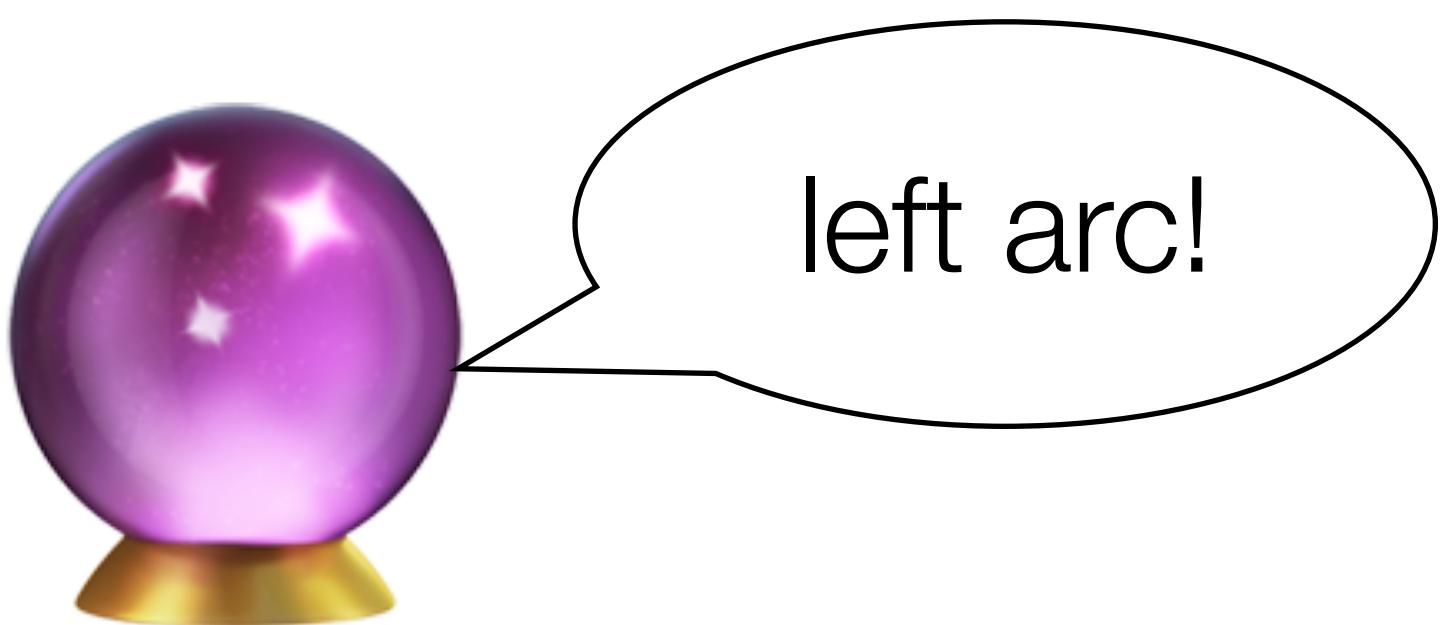
Step 6: assign flight as  
head of morning, remove  
morning

# Dependency Parsing

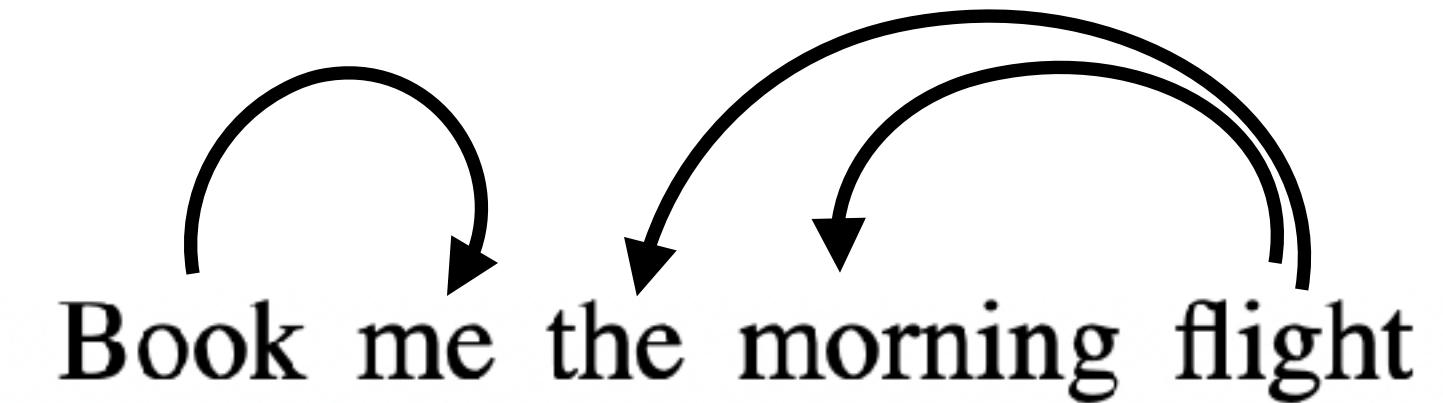
```
stack = [flight, book, root]
```

```
buffer = []
```

```
relations = [book->me, flight->morning,  
             flight->the]
```



Step 7: assign flight as head of the, remove the

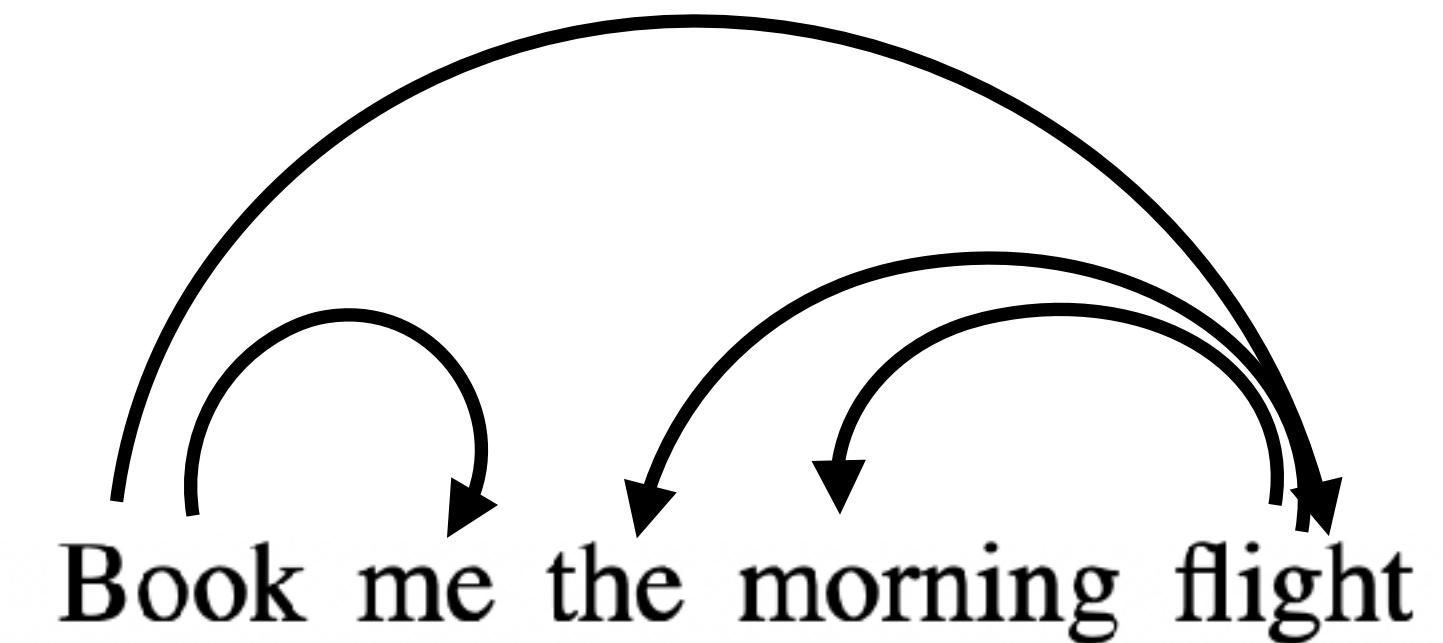
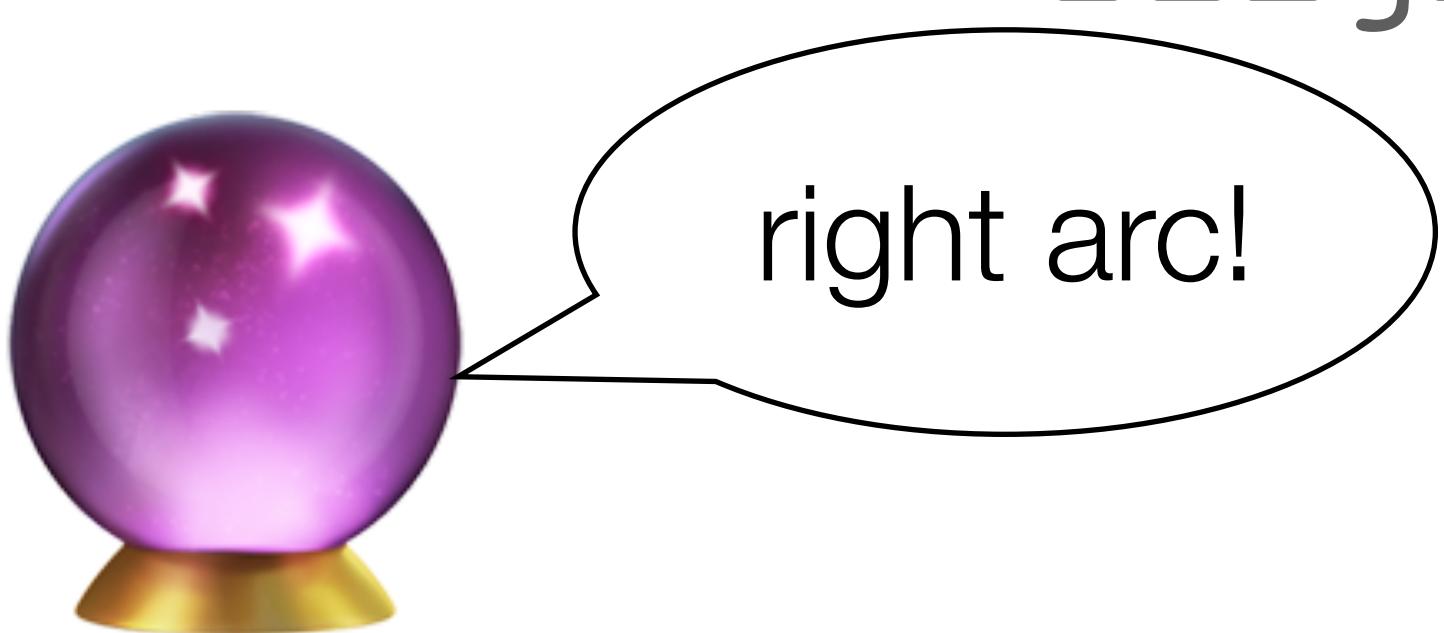


# Dependency Parsing

```
stack = [book, root]
```

```
buffer = []
```

```
relations = [book->me, flight->morning,  
             flight->the, book->flight]
```



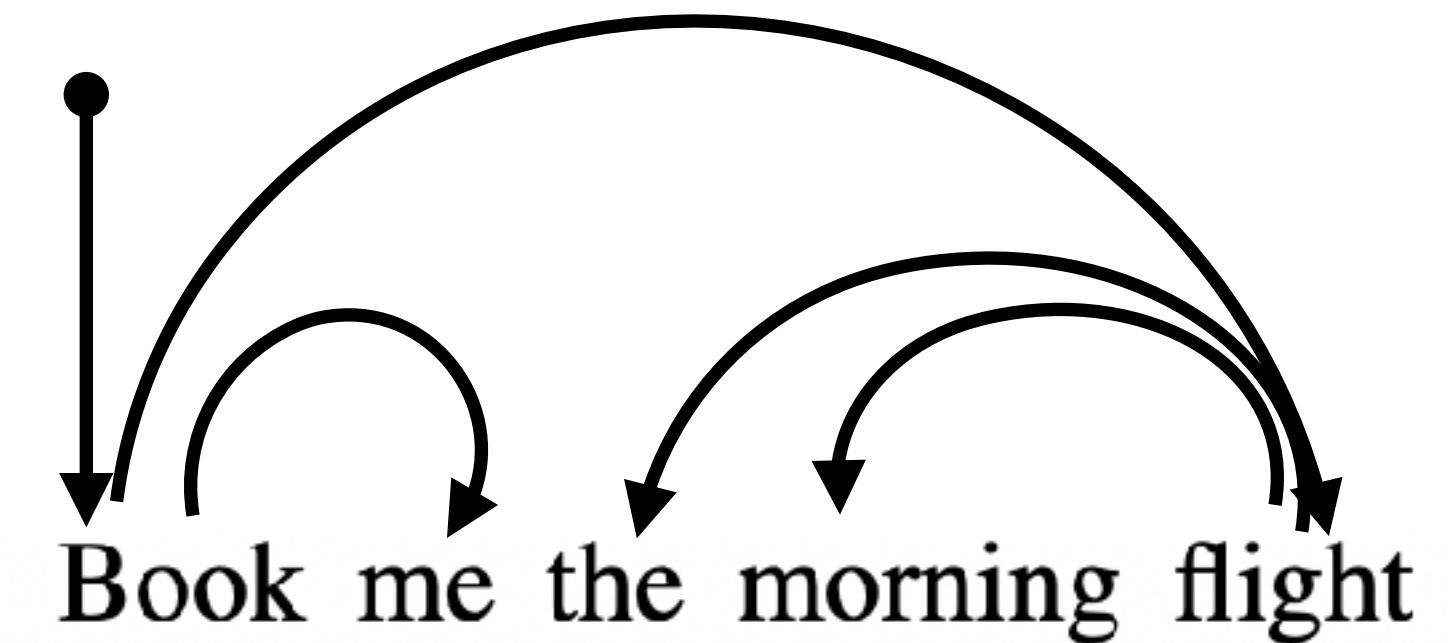
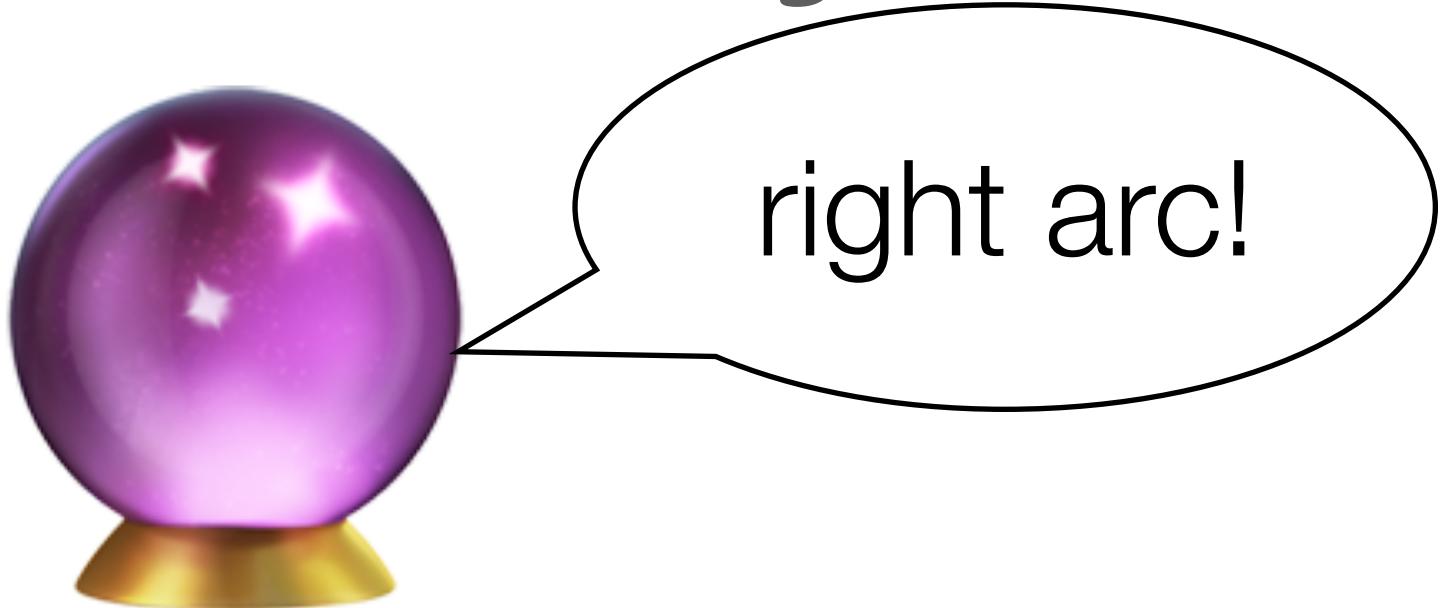
Step 8: assign book as head  
of flight, remove flight

# Dependency Parsing

```
stack = [root]
```

```
buffer = []
```

```
relations = [book->me, flight->morning,  
flight->the, book->flight, root->book]
```



Step 9: assign root as head  
of book, remove book

# Dependency Parsing

```
function DEPENDENCYPARSE(words) returns dependency tree  
  
state  $\leftarrow \{[\text{root}], [\text{words}], []\}$  ; initial configuration  
while state not final  
    t  $\leftarrow \text{ORACLE}(\text{state})$       ; choose a transition operator to apply  
    state  $\leftarrow \text{APPLY}(t, \text{state})$  ; apply it, creating a new state  
return state
```

**Figure 14.6** A generic transition-based dependency parser

# Dependency Parsing

```
function DEPENDENCYPARSE(words) returns dependency tree  
state  $\leftarrow \{[\text{root}], [\text{words}], []\}$  ; initial configuration  
while state not final  
    t  $\leftarrow \text{ORACLE}(\textit{state})$  ; choose a transition operator to apply  
    state  $\leftarrow \text{APPLY}(t, \textit{state})$  ; apply it, creating a new state  
return state  
ML Classifier
```

**Figure 14.6** A generic transition-based dependency parser

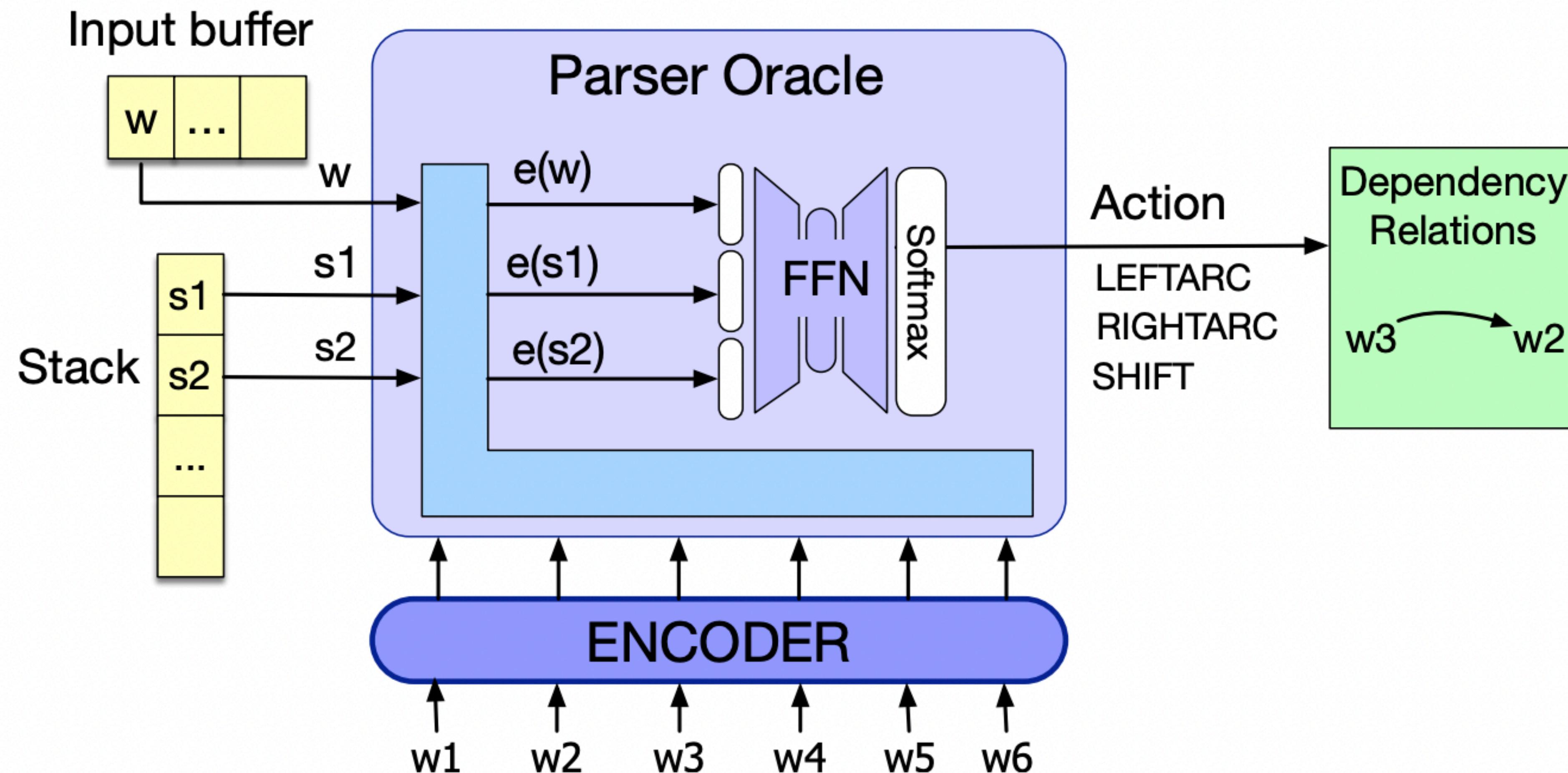
# Dependency Parsing

## Training the Oracle

- Generate training data deterministically from a treebank (see book)
- Then, the usual ML options are available:
  - Feature-based classifiers (Logistic Regression, Naive Bayes, SVM...)
  - Word-Embedding Features
  - Neural Network Classifiers....

# Dependency Parsing

## Training the Oracle



**All done!  
Questions?**