

# Notices

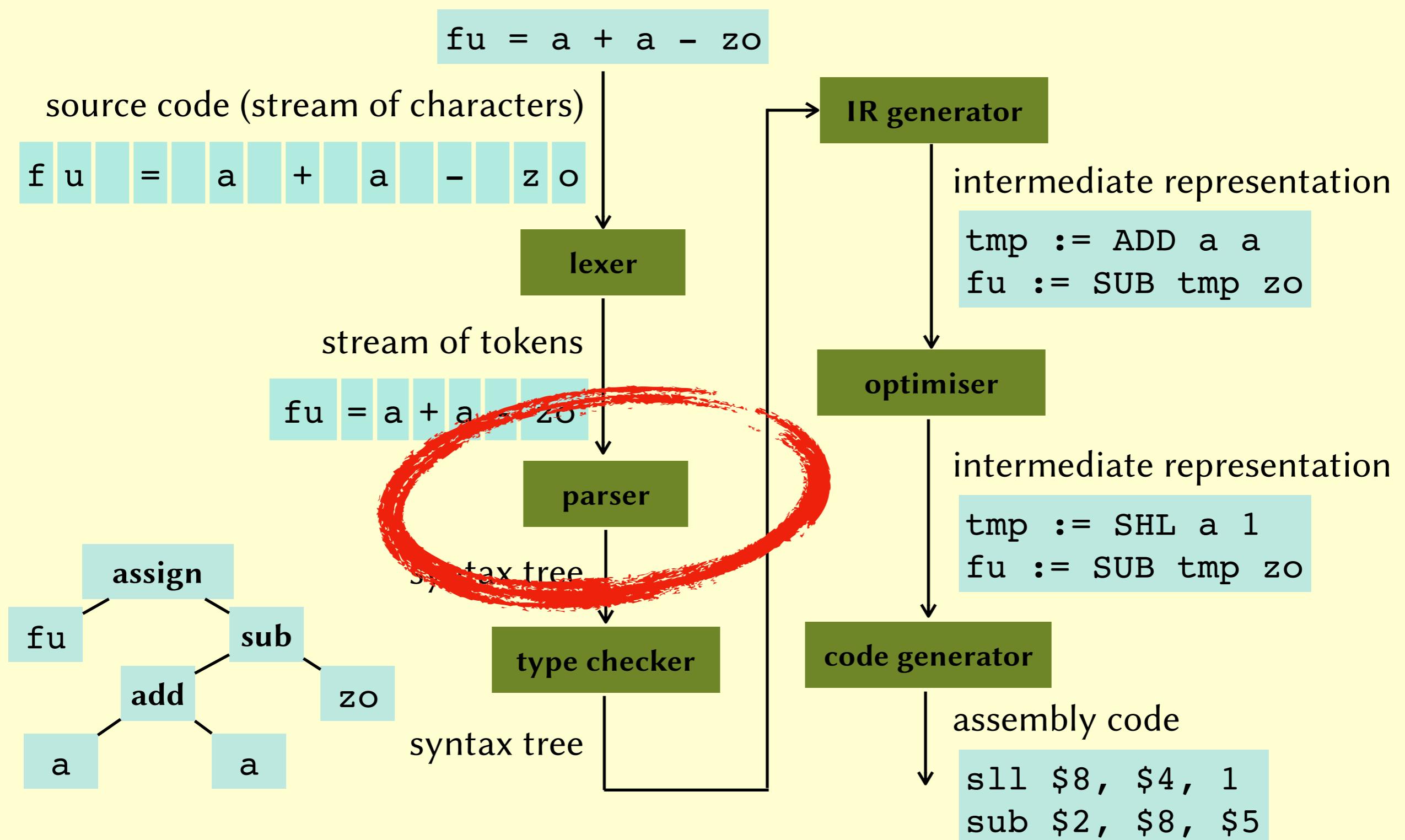
- Coursework spec now available at  
<https://github.com/LangProc/langproc-cw>
- Please start thinking about pairing up.
- Please submit your lab hash to the `compiler_labs.xlsx` spreadsheet in the Files tab of the General channel on Teams (first lab is due in one week).

# Lecture 4: Parsing

John Wickerson

Compilers

# Anatomy of a compiler



# Outline

→ A lesson in **grammar**

- How to build a **recursive descent** parser
- How to build a **shift/reduce** parser
- How to use **Yacc** to generate a parser automatically

# Grammars

sentence → nounphrase verbphrase

nounphrase → det noun

nounphrase → det noun prephrase

verbphrase → verb

verbphrase → verb nounphrase

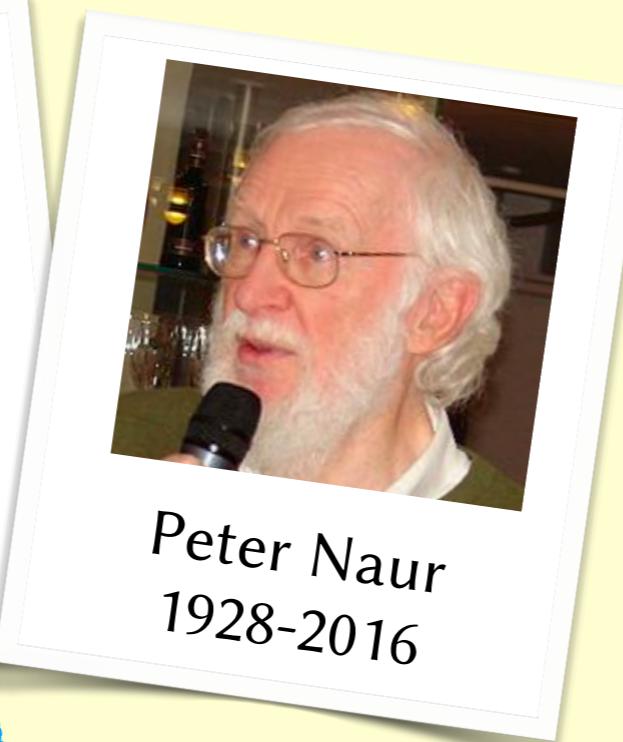
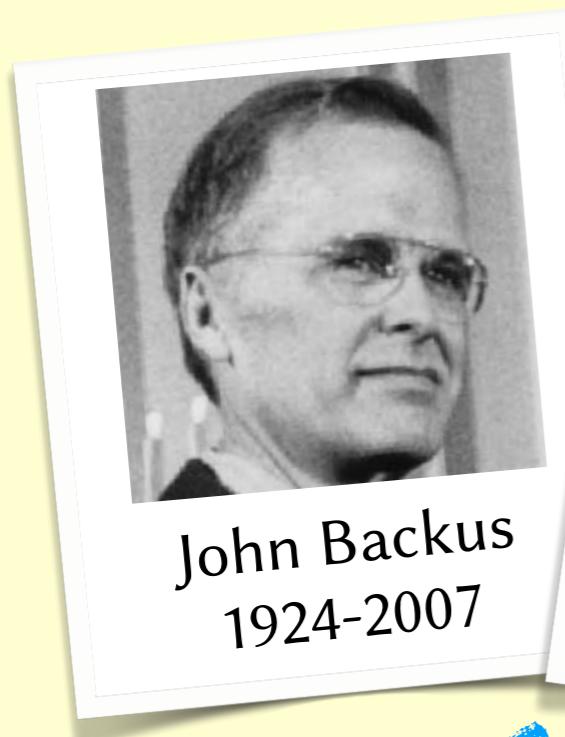
verbphrase → verb prephrase

# Grammars

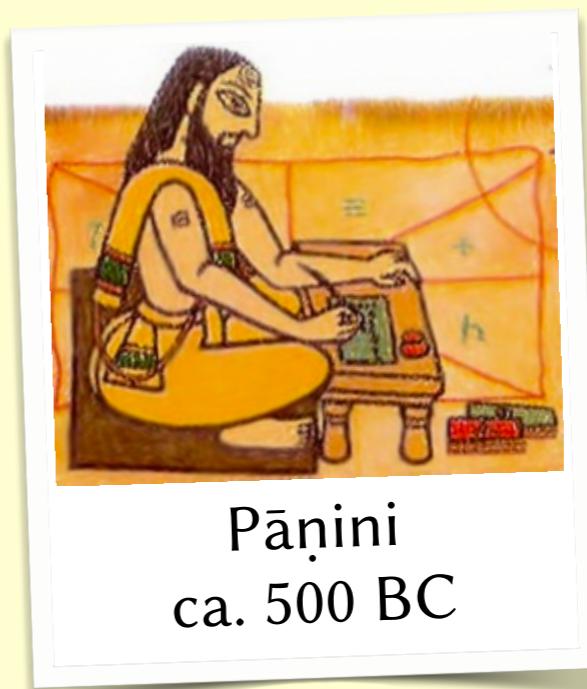
sentence ::= nounphrase verbphrase

nounphrase ::= det noun | det noun prepphrase

verbphrase ::= verb | verb nounphrase | verb prepphrase



↑  
hence, Backus-Naur form



# Grammars

sentence ::= nounphrase verbphrase

nounphrase ::= det noun | det noun prepphrase

verbphrase ::= verb | verb nounphrase | verb prepphrase

prepphrase ::= prep nounphrase

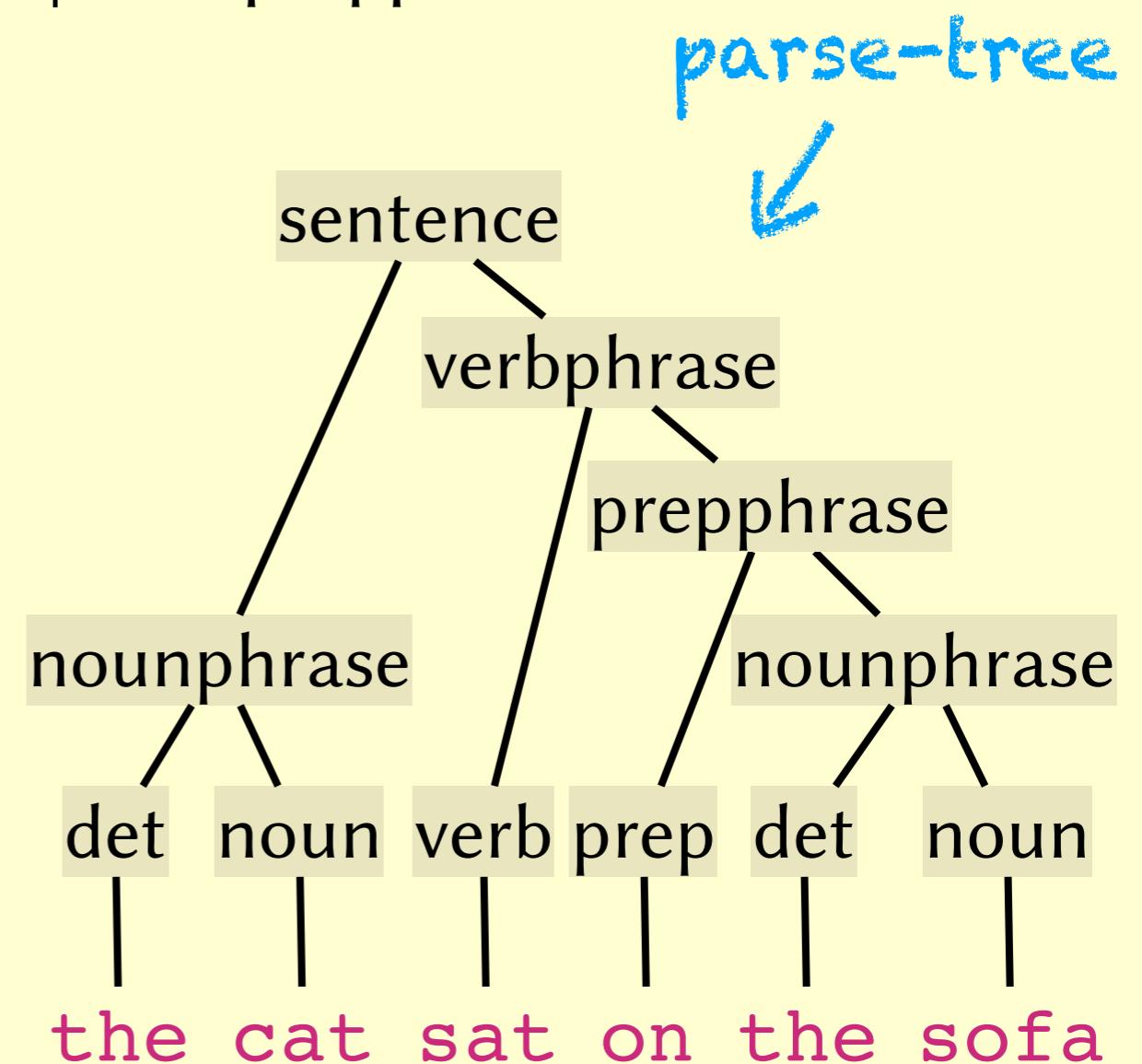
noun ::= cat | dog | sofa

det ::= a | the

verb ::= sat | ate

prep ::= on | with

non-terminal ↑  
terminal ↑  
non-terminal



# Grammars

sentence ::= nounphrase verbphrase

nounphrase

verbphrase

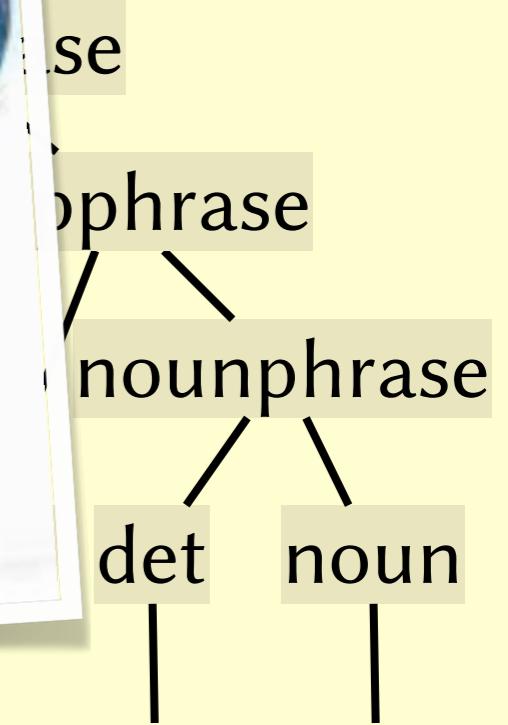
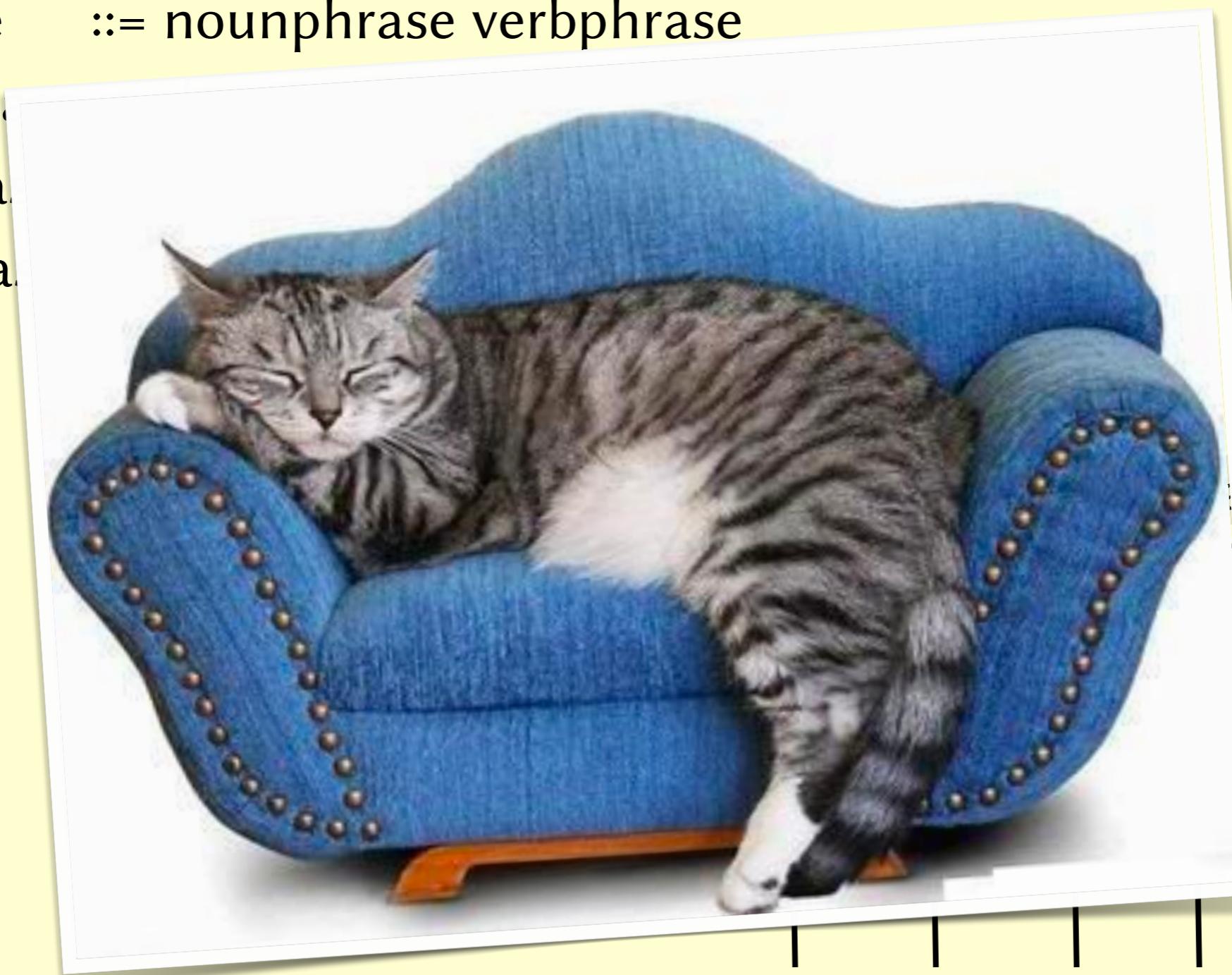
prepphrase

noun

det

verb

prep



the cat sat on the sofa

# Grammars

sentence ::= nounphrase verbphrase

nounphrase ::= det noun | det noun prepphrase

verbphrase ::= verb | verb nounphrase | verb prepphrase

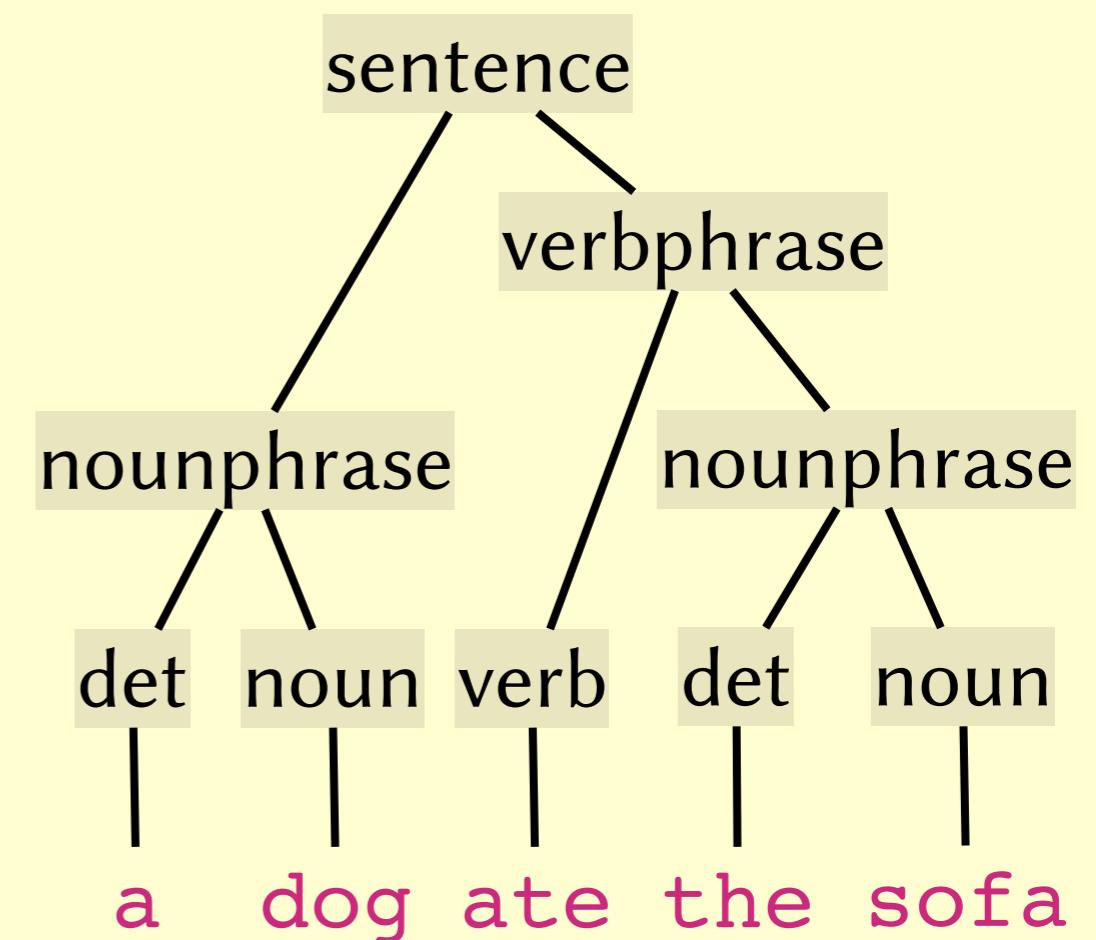
prepphrase ::= prep nounphrase

noun ::= cat | dog | sofa

det ::= a | the

verb ::= sat | ate

prep ::= on | with



# Grammars

sentence

nounphrase

verbphrase

prepphrase

noun

det

verb

prep



# Grammars

sentence ::= nounphrase verbphrase

nounphrase ::= det noun | det noun prepphrase

verbphrase ::= verb | verb nounphrase | verb prepphrase

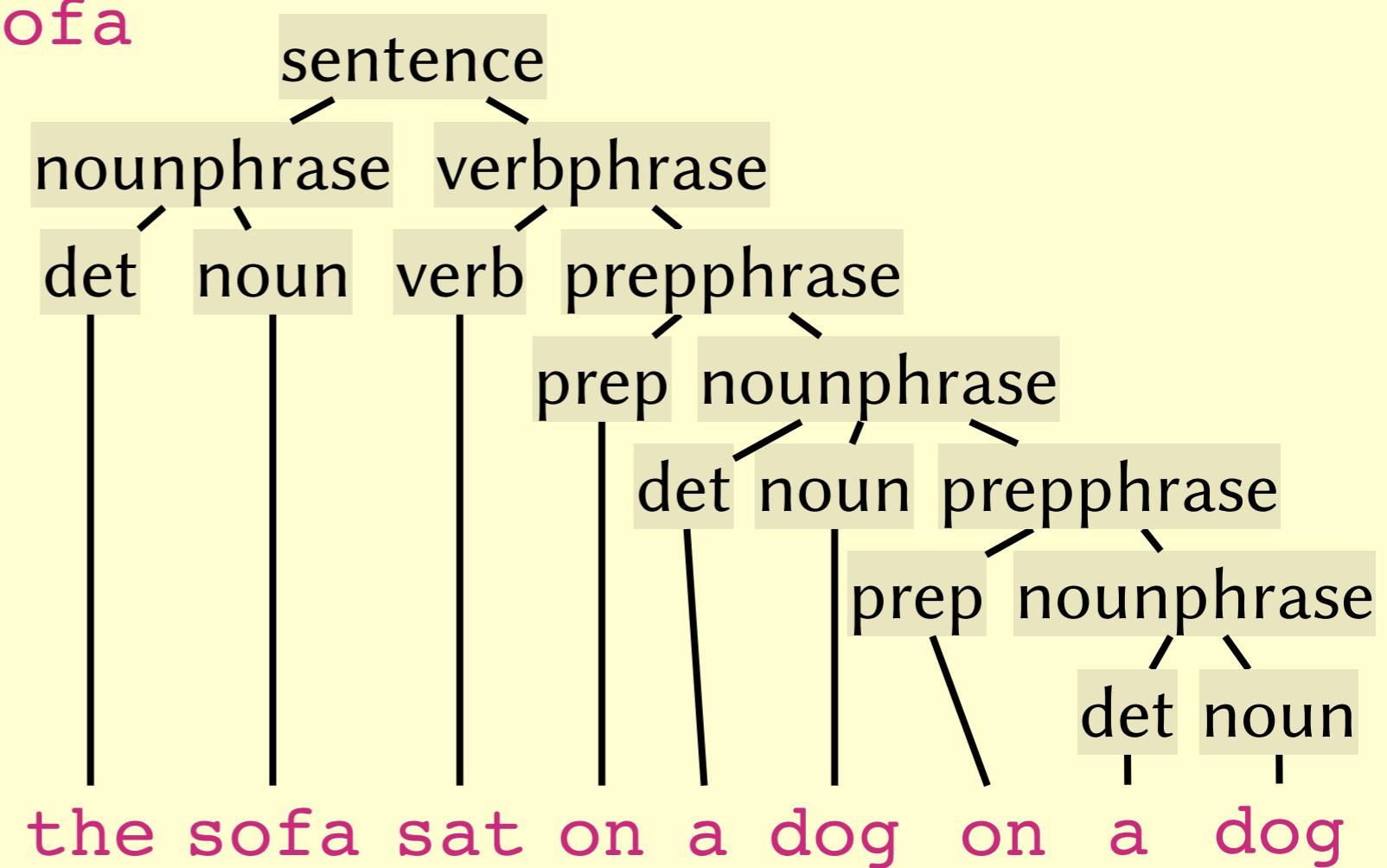
prepphrase ::= prep nounphrase

noun ::= cat | dog | sofa

det ::= a | the

verb ::= sat | ate

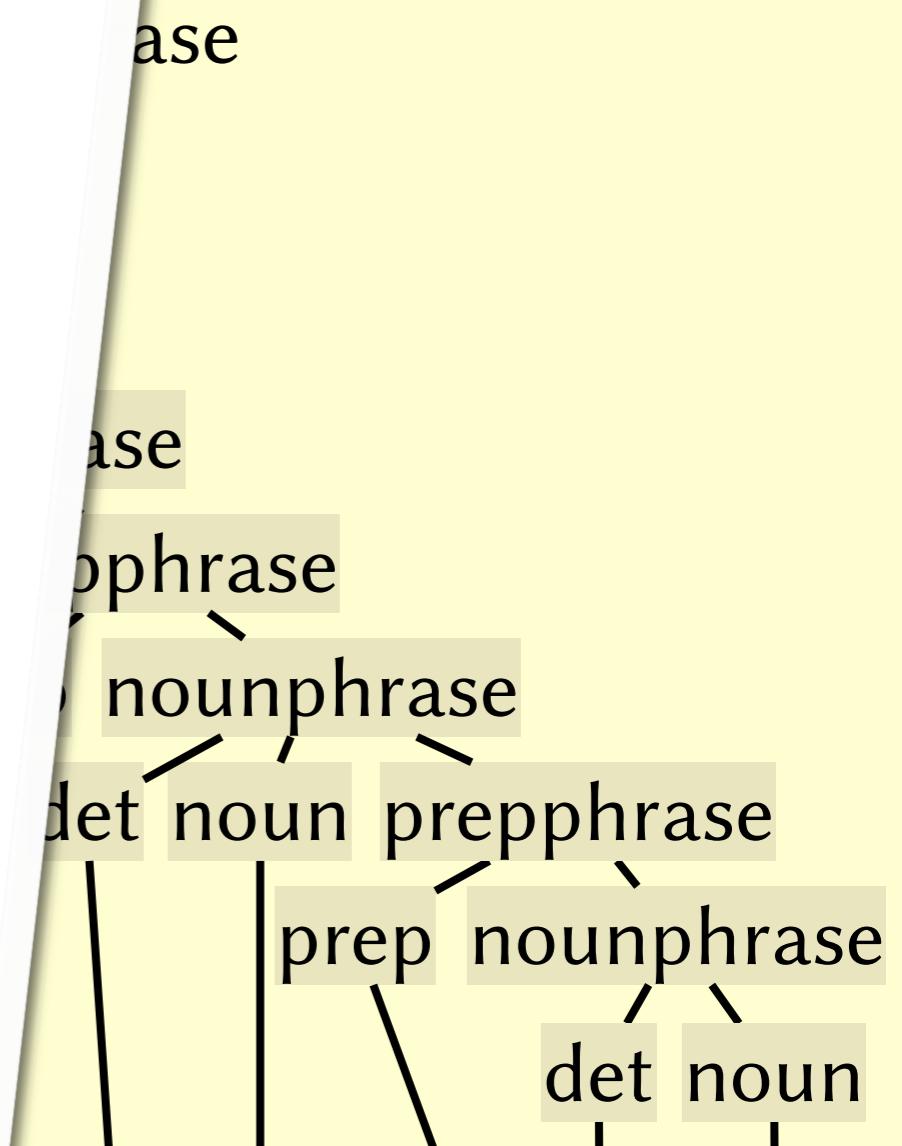
prep ::= on | with



sentence ::  
nounphrase ::  
verbphrase ::  
prepphrase  
noun  
det  
verb  
prep



the sofa sat on a dog on a dog



# Ambiguity

**sentence** ::= nounphrase verbphrase

**nounphrase ::= det noun | det noun prepphrase**

`verbphrase ::= verb | verb nounphrase | verb prepphrase | verb nounphrase`

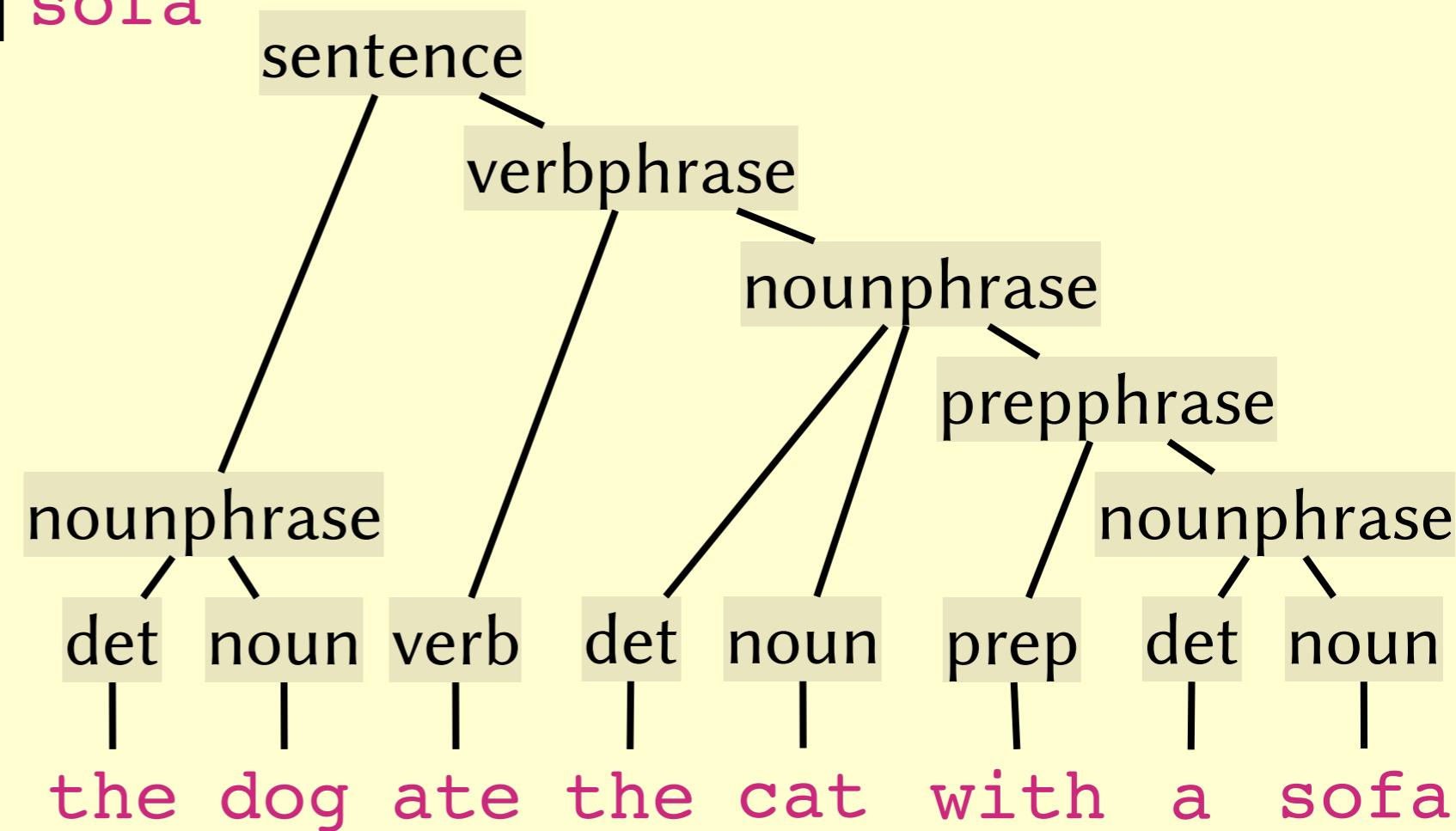
**prepphrase** ::= **prep nounphrase**

**noun** ::= **cat** | **dog** | **sofa**

**det** ::= a | the

**verb** ::= sat | ate

**prep** ::= **on** | **with**



# Ambiguity

**sentence** ::= nounphrase verbphrase

**nounphrase** ::= **det noun** | **det noun prepphrase**

**verbphrase** ::= **verb** | **verb nounphrase** | **verb prepphrase** | **verb nounphrase**

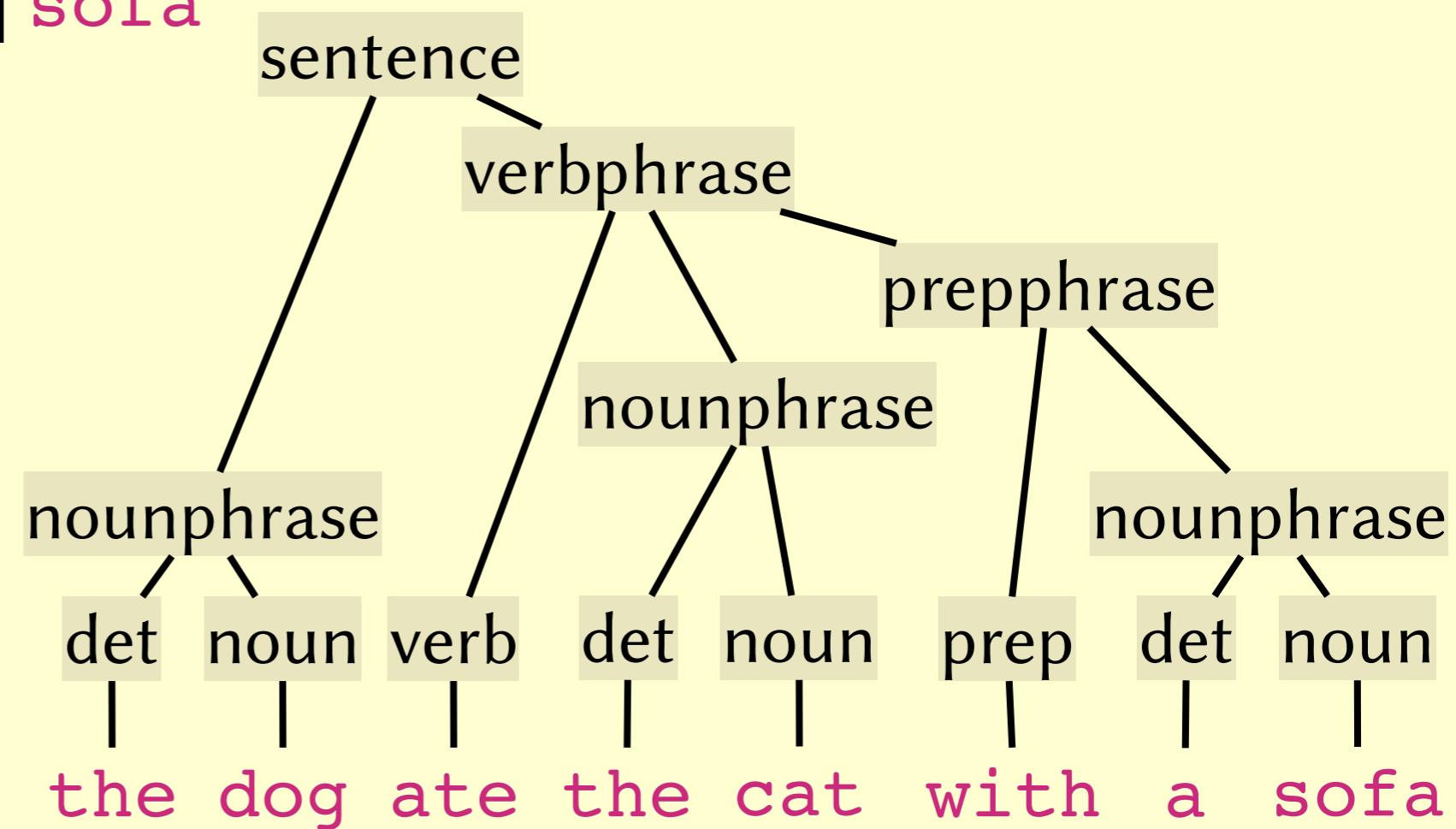
**prepphrase** ::= **prep nounphrase**

**noun** ::= **cat** | **dog** | **sofa**

**det** ::= a | the

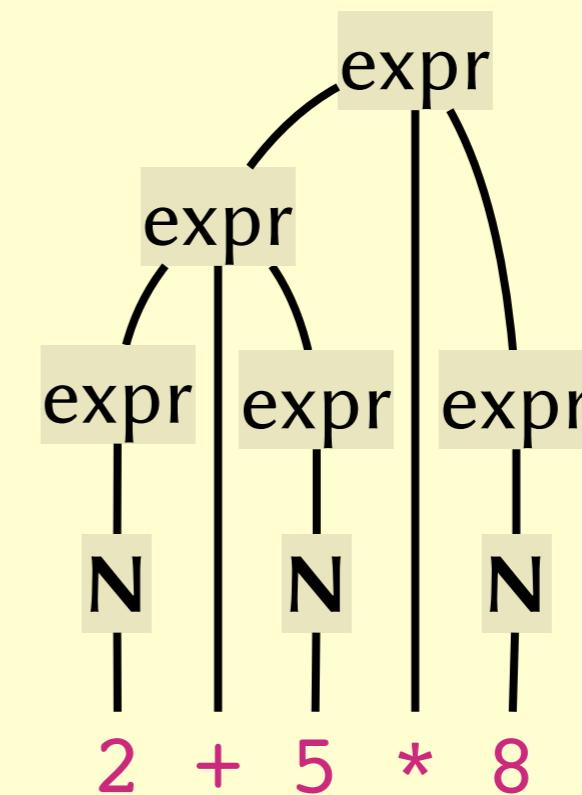
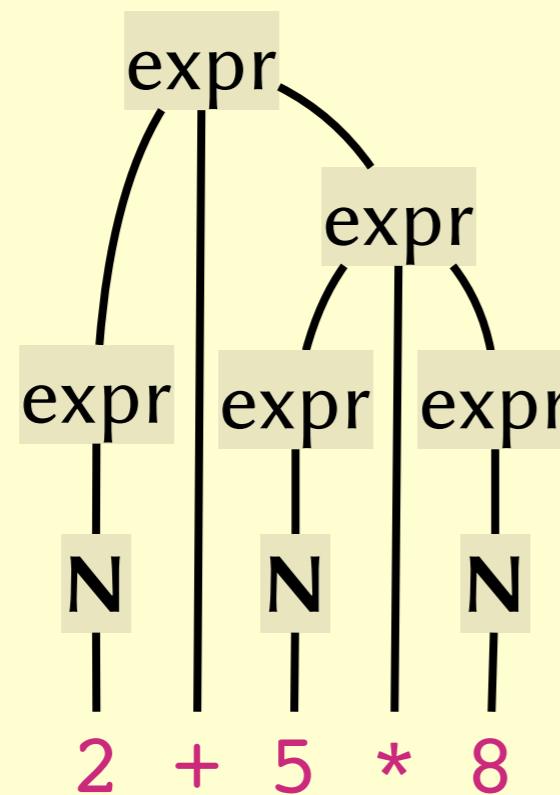
**verb** ::= sat | ate

**prep** ::= **on** | **with**



# Expressions

expr ::= expr + expr | expr \* expr | ( expr ) | N

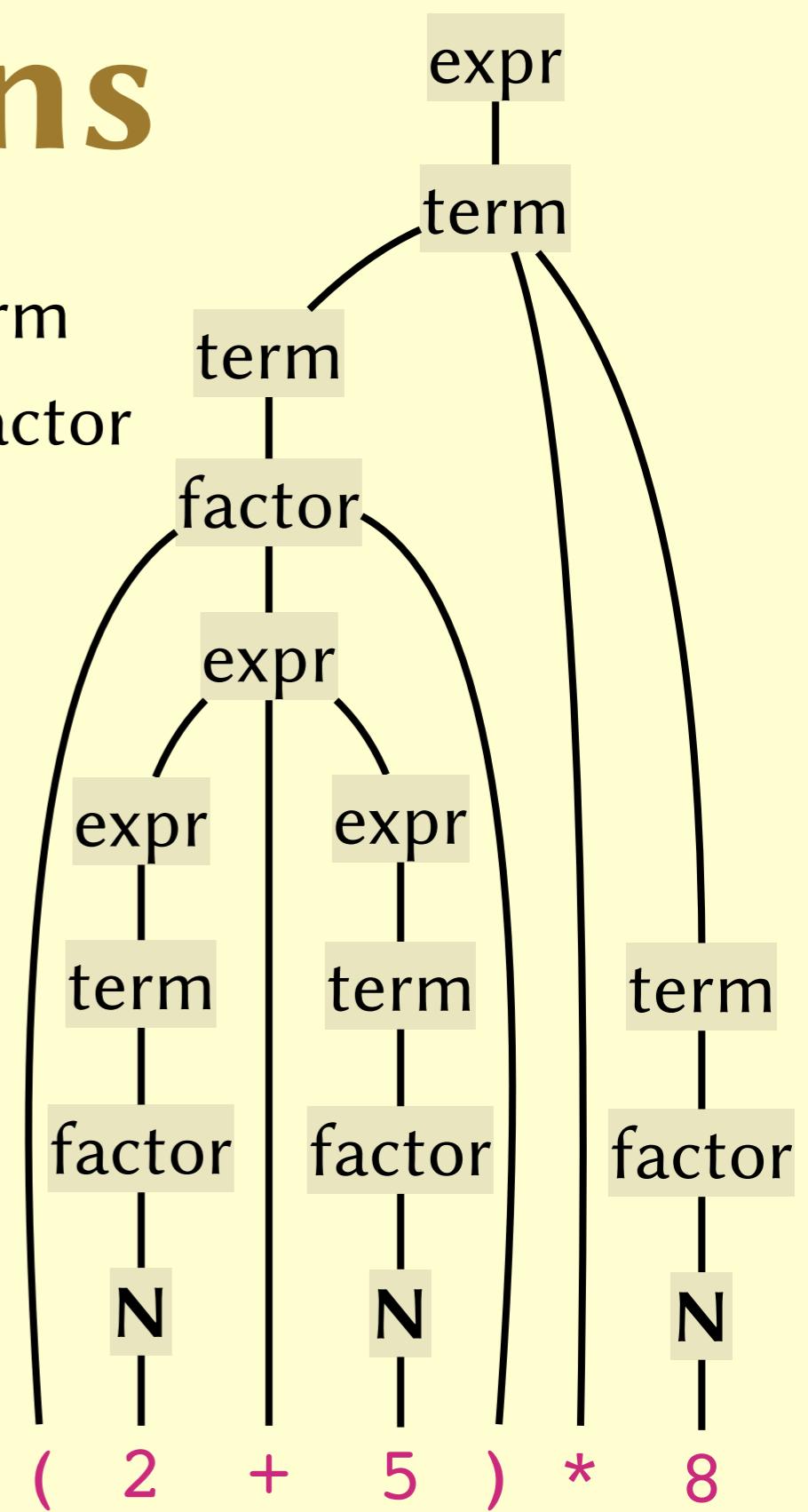
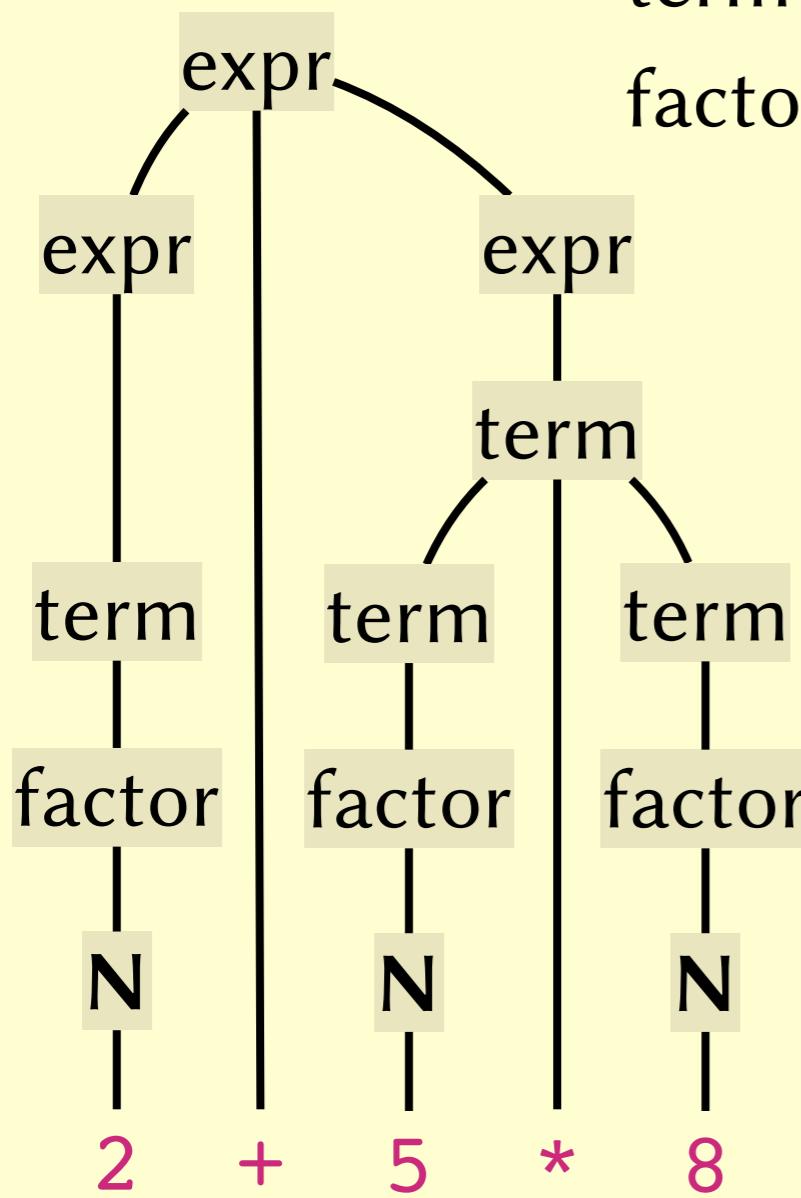


# Expressions

expr ::= expr + expr | term

term ::= term \* term | factor

factor ::= ( expr ) | N

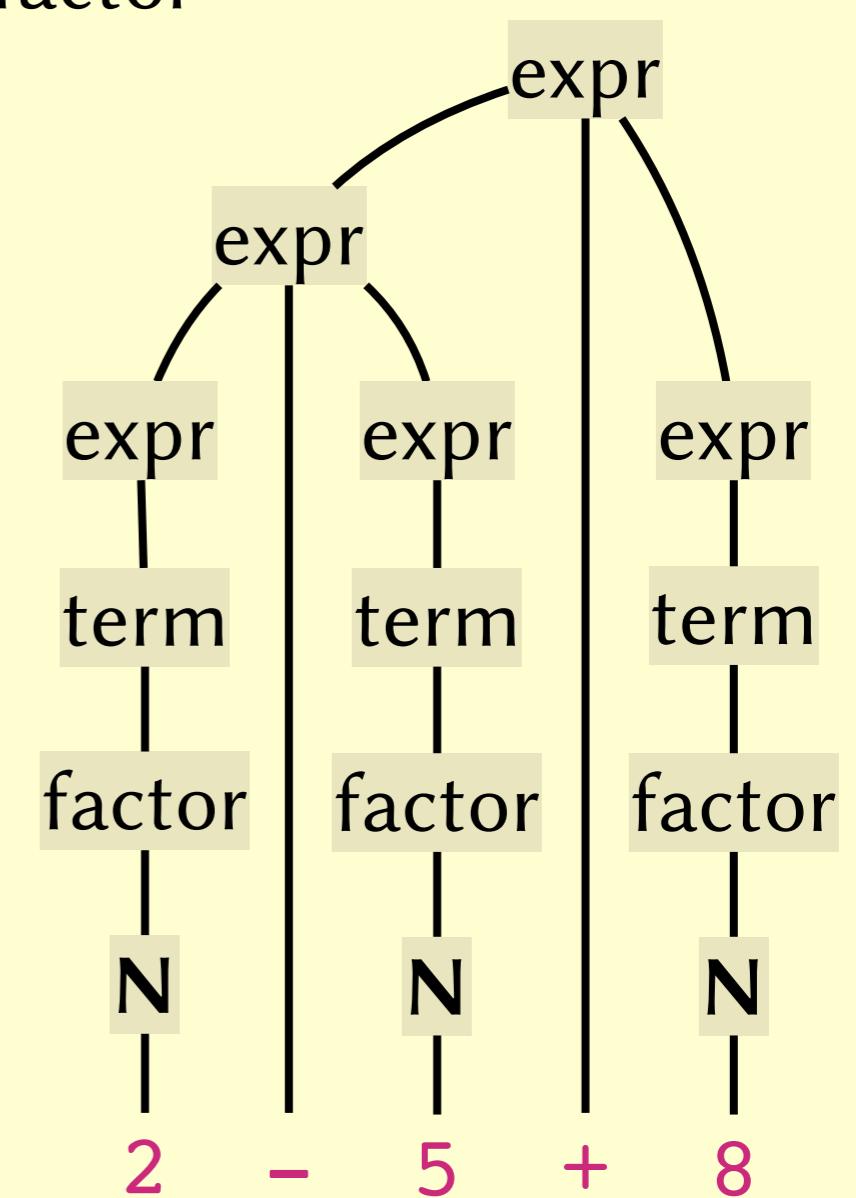
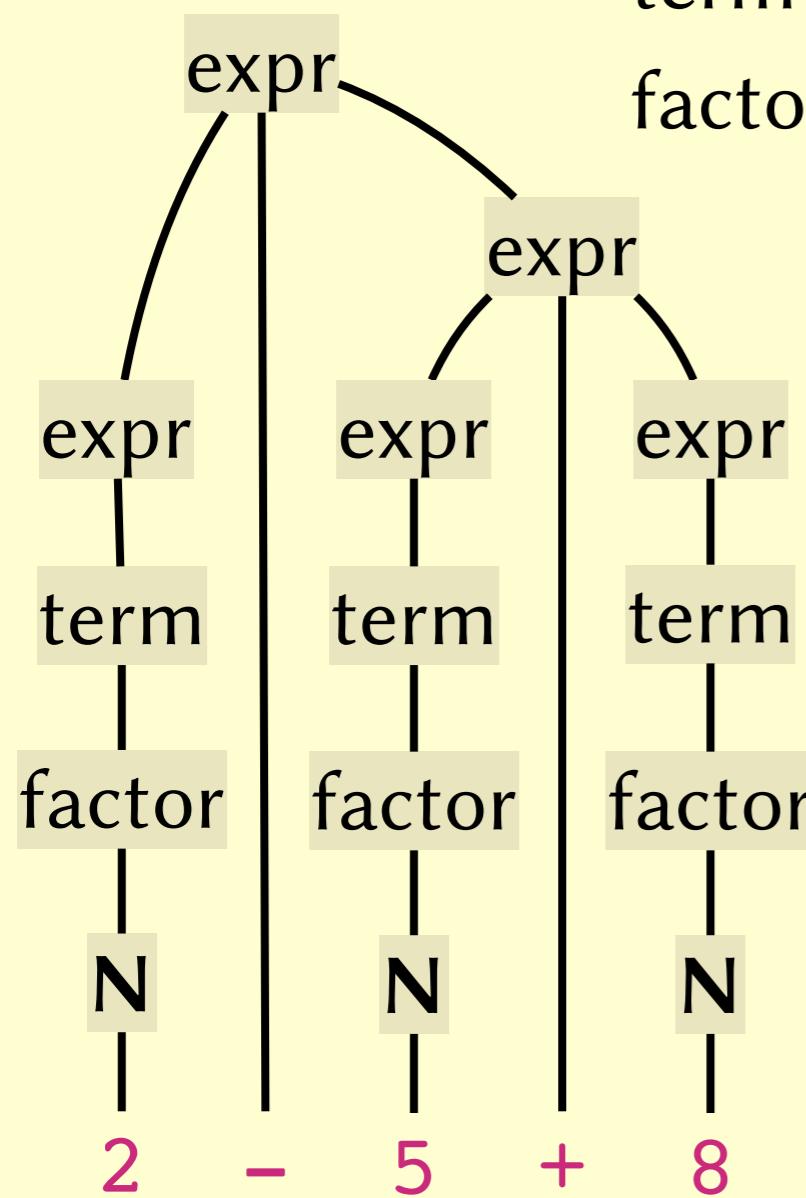


# Expressions

expr ::= expr + expr | expr - expr | term

term ::= term \* term | factor

factor ::= ( expr ) | N

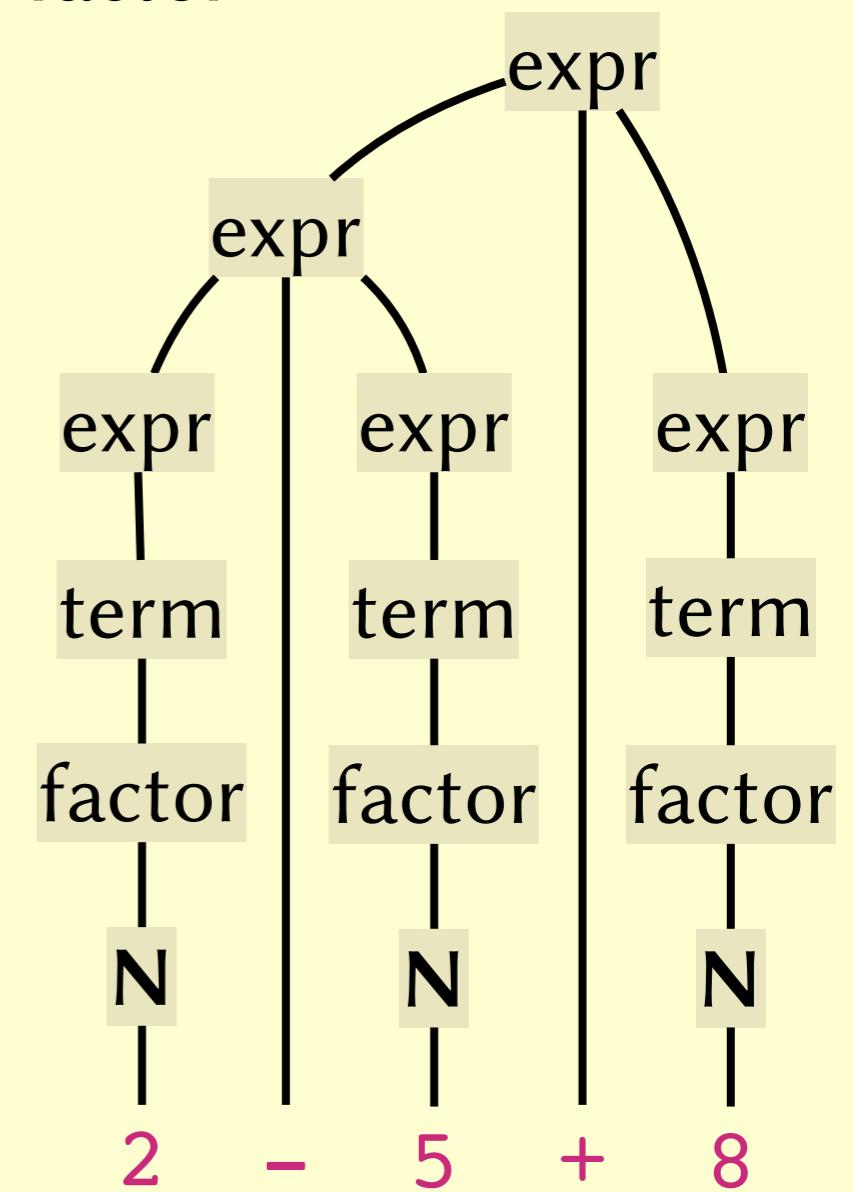
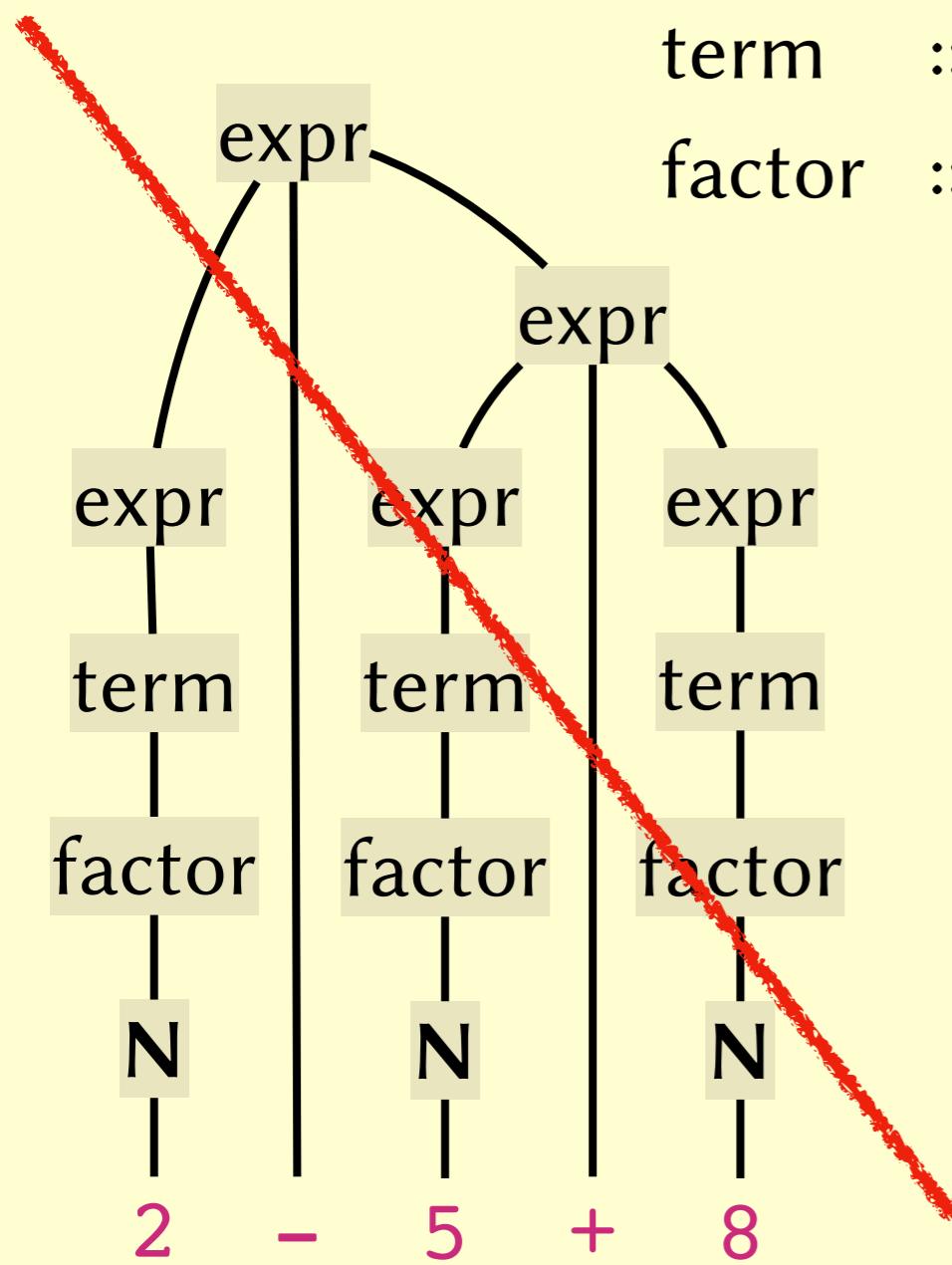


# Expressions

expr ::= expr + term | expr - term | term

term ::= term \* factor | factor

factor ::= ( expr ) | N

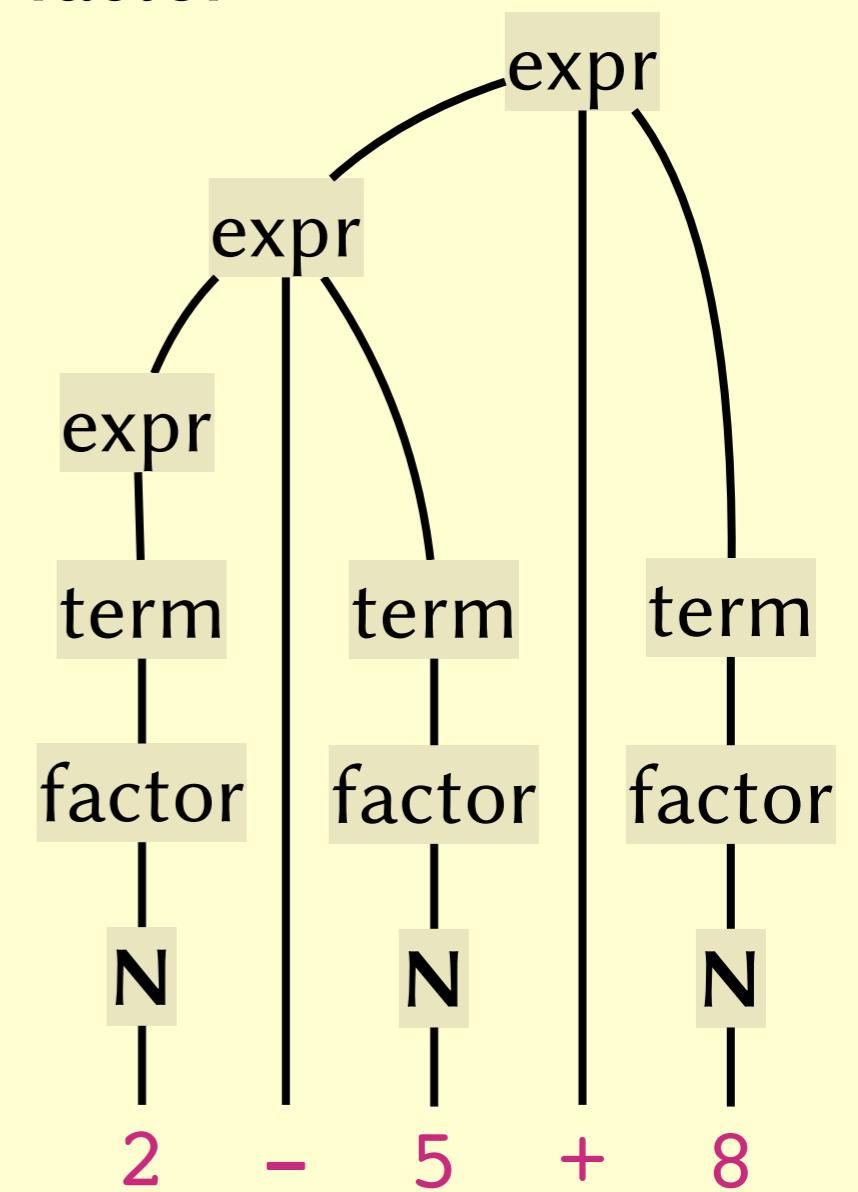
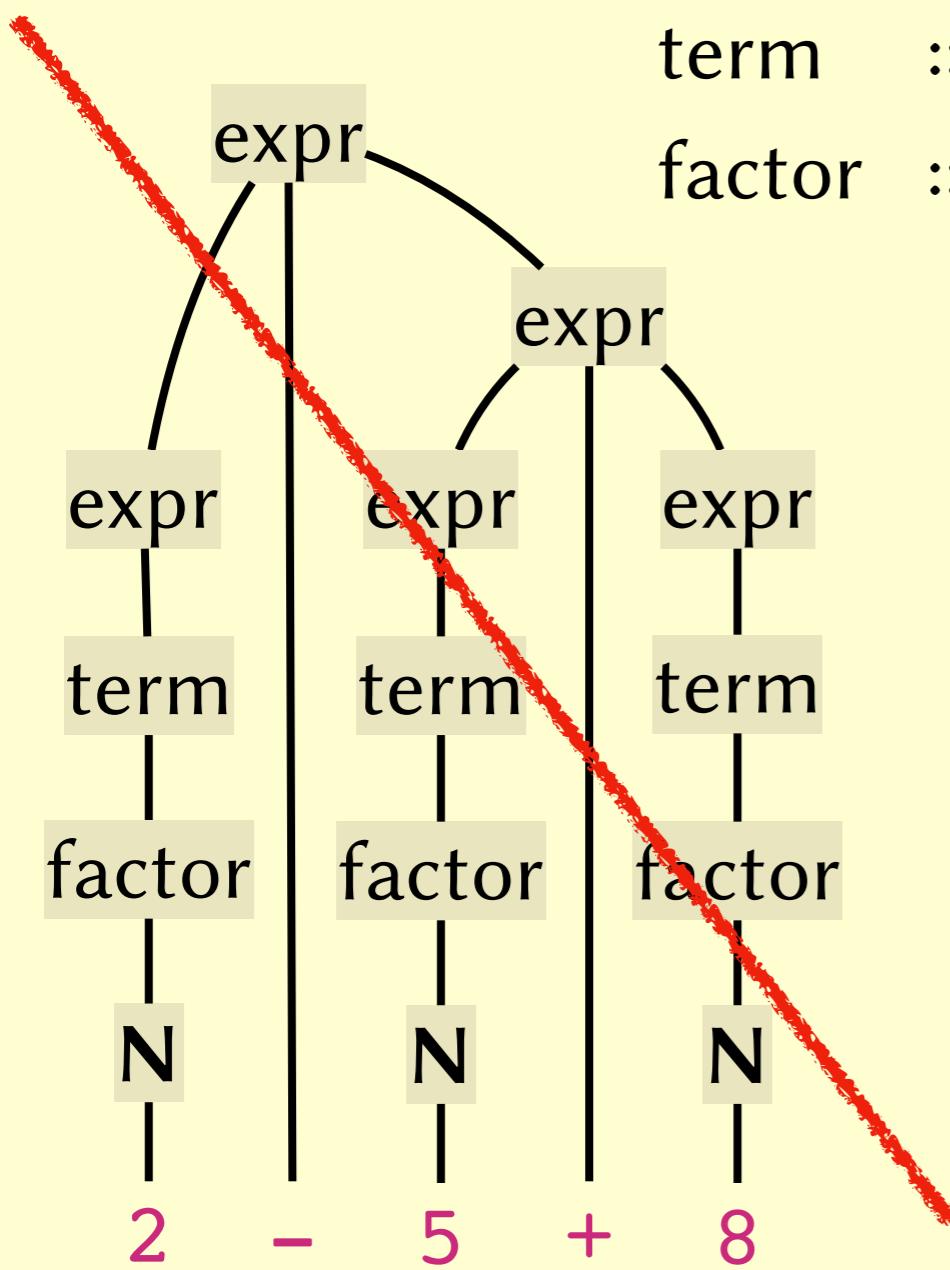


# Expressions

expr ::= expr + term | expr - term | term

term ::= term \* factor | factor

factor ::= ( expr ) | N



# Grammars vs. regexes

$$L ::= aaL \mid a$$

# Grammars vs. regexes

{  $a^n$  | n is odd }

$L ::= aaL \mid a$

# Grammars vs. regexes

{  $a^n$  | n is odd }

$L ::= aaL \mid a \quad a(aa)^*$

# Grammars vs. regexes

**Set of words in the language**

{  $a^n$  | n is odd }

**Grammar**

$L ::= aaL \mid a$

**Regex**

$a(aa)^*$

# Grammars vs. regexes

Set of words in the language	Grammar	Regex
$\{ a^n \mid n \text{ is odd} \}$	$L ::= aaL \mid a$	$a(aa)^*$
	$L ::= aL \mid abL \mid \epsilon$	

# Grammars vs. regexes

Set of words in the language	Grammar	Regex
$\{ a^n \mid n \text{ is odd} \}$	$L ::= aaL \mid a$	$a(aa)^*$
no consecutive <b>b</b> s	$L ::= aL \mid abL \mid \epsilon$	

# Grammars vs. regexes

Set of words in the language	Grammar	Regex
$\{ a^n \mid n \text{ is odd} \}$	$L ::= aaL \mid a$	$a(aa)^*$
no consecutive <b>b</b> s	$L ::= aL \mid abL \mid \epsilon$	$(ab?)^*$

# Grammars vs. regexes

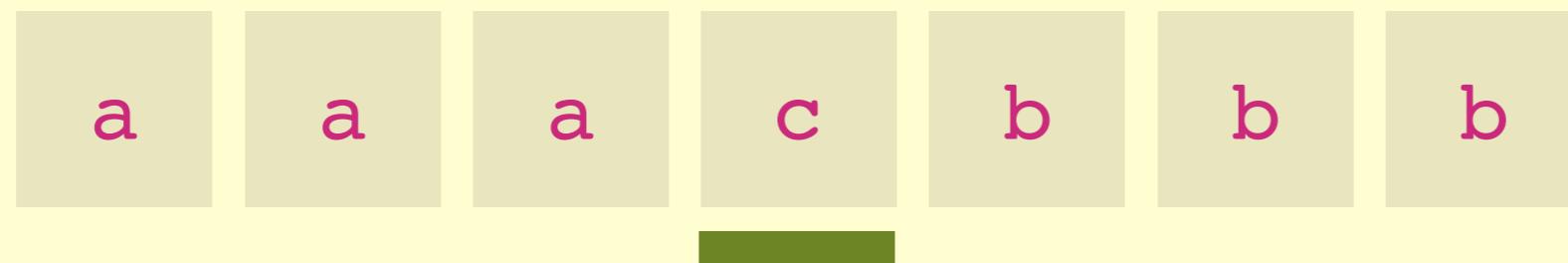
Set of words in the language	Grammar	Regex
$\{ a^n \mid n \text{ is odd} \}$	$L ::= aaL \mid a$	$a(aa)^*$
no consecutive <b>b</b> s	$L ::= aL \mid abL \mid \epsilon$	$(ab?)^*$
$\{ a^n b^n \mid n \geq 0 \}$		

# Grammars vs. regexes

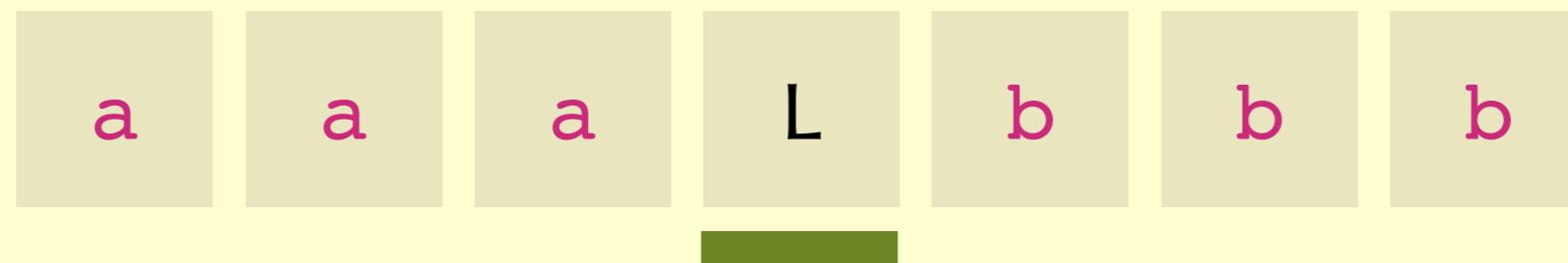
Set of words in the language	Grammar	Regex
$\{ a^n \mid n \text{ is odd} \}$	$L ::= aaL \mid a$	$a(aa)^*$
no consecutive <b>b</b> s	$L ::= aL \mid abL \mid \epsilon$	$(ab?)^*$
$\{ a^n c b^n \mid n \geq 0 \}$	$L ::= aLb \mid c$	

$$\{ a^n c b^n \mid n \geq 0 \}$$

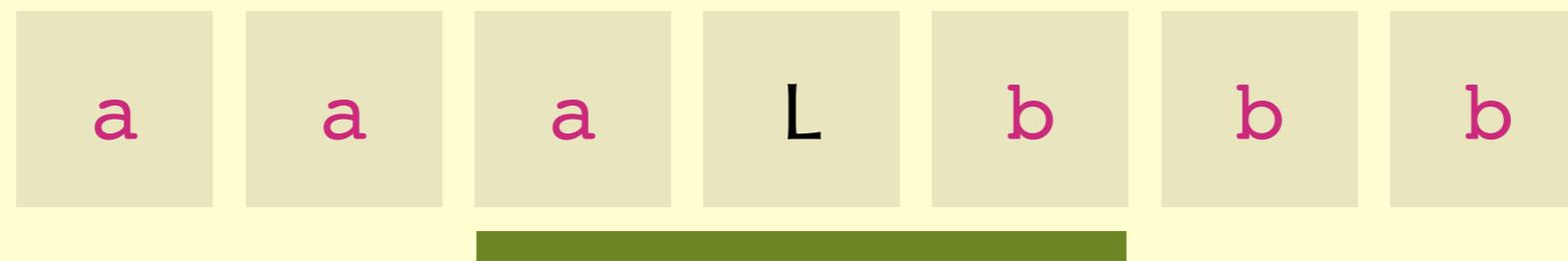
$L \rightarrow aLb$   
→  $L \rightarrow c$



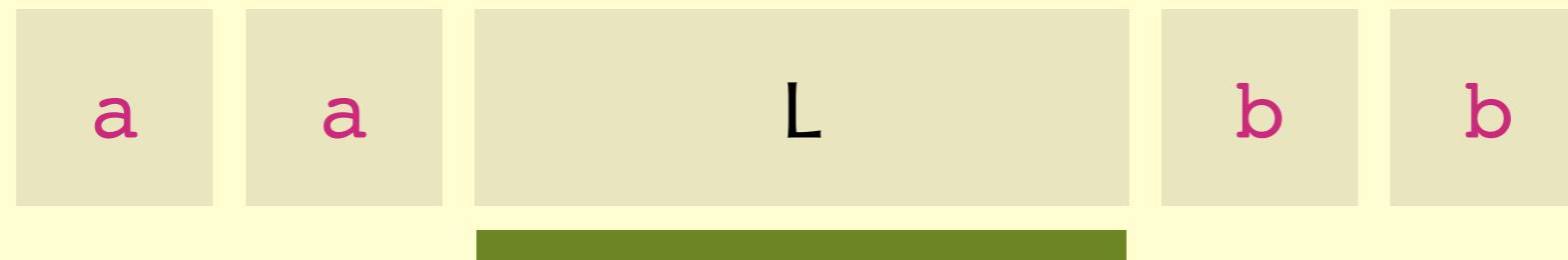
$L \rightarrow aLb$   
→  $L \rightarrow c$



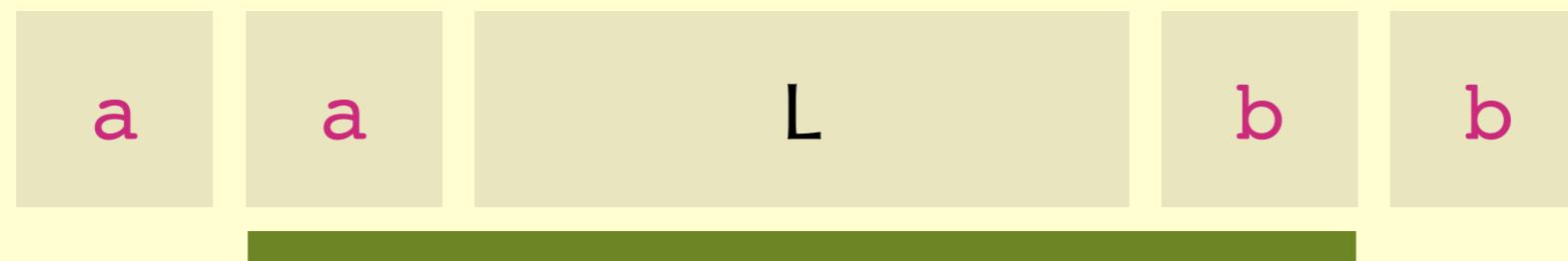
→ L → aLb  
L → c



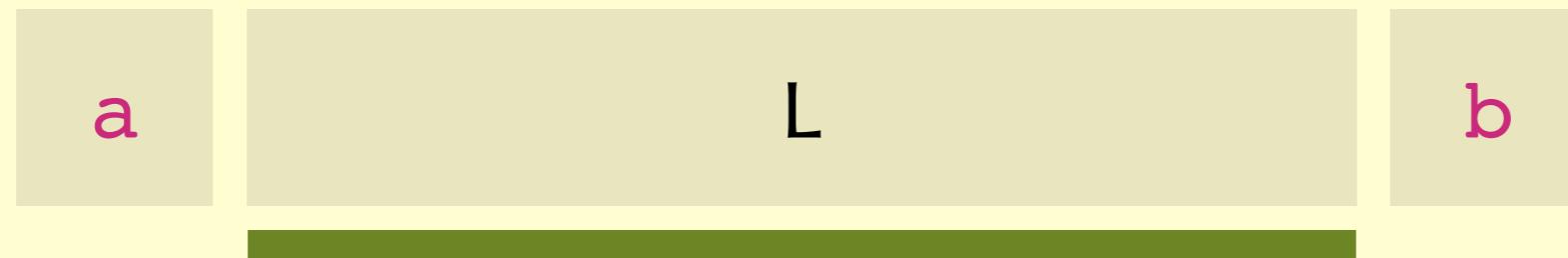
→  $L \rightarrow aLb$   
 $L \rightarrow c$



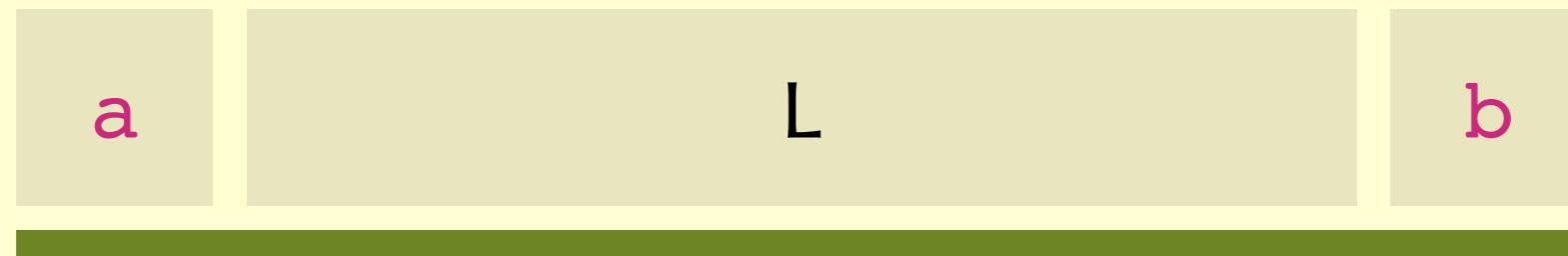
→  $L \rightarrow aLb$   
 $L \rightarrow c$



→  $L \rightarrow aLb$   
 $L \rightarrow c$



→  $L \rightarrow aLb$   
 $L \rightarrow c$

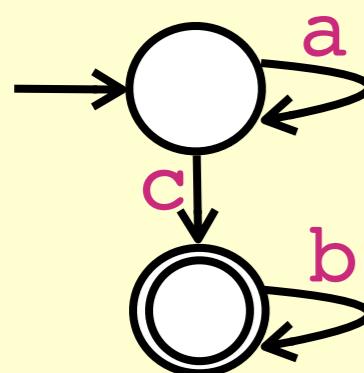


→  $L \rightarrow aLb$   
 $L \rightarrow c$

$L$

# Grammars vs. regexes

Set of words in the language	Grammar	Regex
$\{ a^n \mid n \text{ is odd} \}$	$L ::= aaL \mid a$	$a(aa)^*$
no consecutive <b>b</b> s	$L ::= aL \mid abL \mid \epsilon$	$(ab?)^*$
$\{ a^n c b^n \mid n \geq 0 \}$	$L ::= aLb \mid c$	



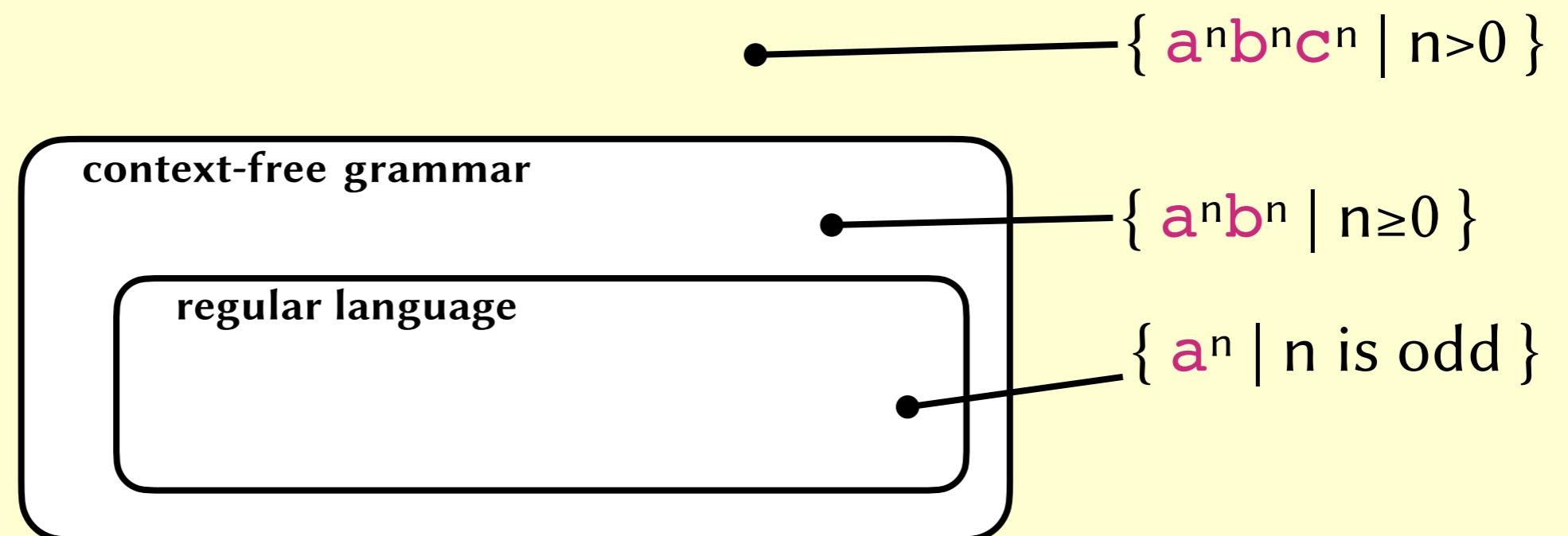
# Grammars vs. regexes

Set of words in the language	Grammar	Regex
$\{ a^n \mid n \text{ is odd} \}$	$L ::= aaL \mid a$	$a(aa)^*$
no consecutive <b>b</b> s	$L ::= aL \mid abL \mid \epsilon$	$(ab?)^*$
$\{ a^n c b^n \mid n \geq 0 \}$	$L ::= aLb \mid c$	<b>X</b>
	<p>not a "regular" language</p> <p>recall <math>\text{expr} ::= \dots \mid (\text{expr})</math></p> <p>not a "regular" grammar</p>	

# Grammars vs. regexes

Set of words in the language	Grammar	Regex
$\{ a^n \mid n \text{ is odd} \}$	$L ::= aaL \mid a$	$a(aa)^*$
no consecutive <b>b</b> s	$L ::= aL \mid abL \mid \epsilon$	$(ab?)^*$
$\{ a^n b^n \mid n \geq 0 \}$	$L ::= aLb \mid c$	✗
$\{ a^n b^n c^n \mid n > 0 \}$	✗	✗

# Hierarchy of languages



# Context-sensitive grammars

Note:  $|LHS| \leq |RHS|$



$L \rightarrow aMLc$

$L \rightarrow aMc$

$Ma \rightarrow aM$

$Mc \rightarrow bc$

$Mb \rightarrow bb$

$\{ a^n b^n c^n \mid n > 0 \}$



# Context-sensitive grammars

$L \rightarrow aMLc$

$L \rightarrow aMc$

$Ma \rightarrow aM$

$Mc \rightarrow bc$

→  $Mb \rightarrow bb$

$\{ a^n b^n c^n \mid n > 0 \}$



# Context-sensitive grammars

$L \rightarrow aMLc$

$L \rightarrow aMc$

$Ma \rightarrow aM$

$Mc \rightarrow bc$

→  $Mb \rightarrow bb$

$\{ a^n b^n c^n \mid n > 0 \}$



# Context-sensitive grammars

$L \rightarrow aMLc$

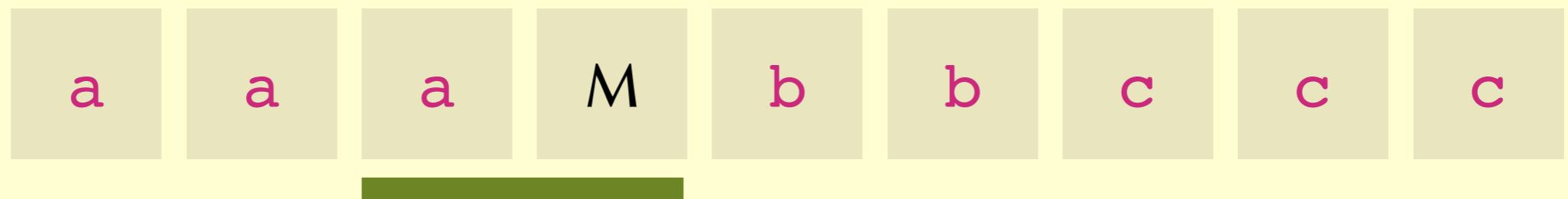
$L \rightarrow aMc$

$\rightarrow Ma \rightarrow aM$

$Mc \rightarrow bc$

$Mb \rightarrow bb$

$\{ a^n b^n c^n \mid n > 0 \}$



# Context-sensitive grammars

$L \rightarrow aMLc$

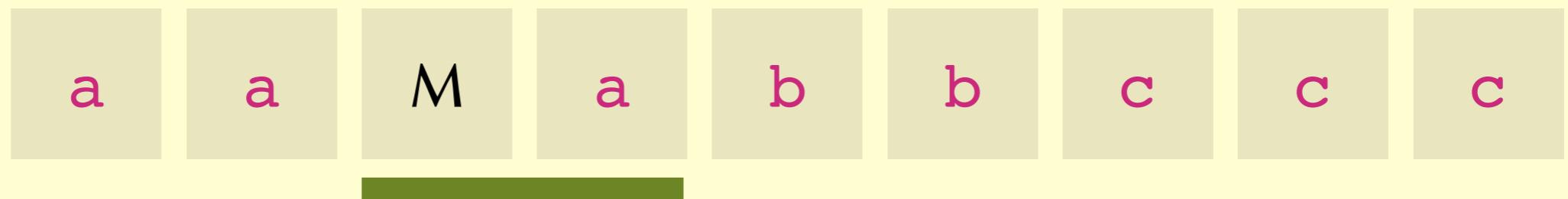
$L \rightarrow aMc$

$\rightarrow Ma \rightarrow aM$

$Mc \rightarrow bc$

$Mb \rightarrow bb$

$\{ a^n b^n c^n \mid n > 0 \}$



# Context-sensitive grammars

$L \rightarrow aMLc$

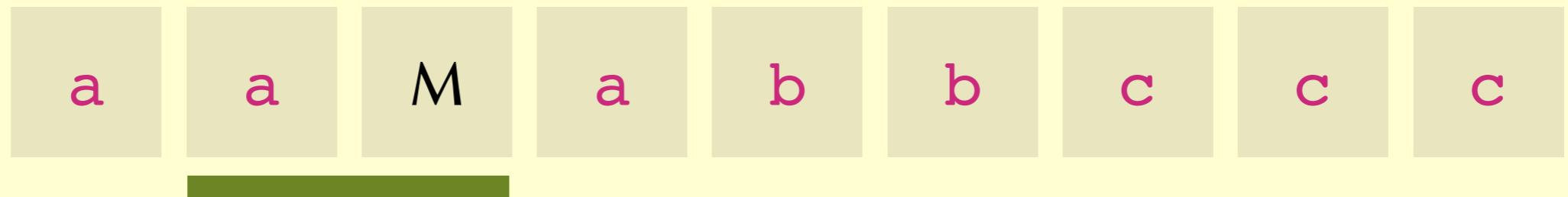
$L \rightarrow aMc$

$\rightarrow Ma \rightarrow aM$

$Mc \rightarrow bc$

$Mb \rightarrow bb$

$\{ a^n b^n c^n \mid n > 0 \}$



# Context-sensitive grammars

$L \rightarrow aMLc$

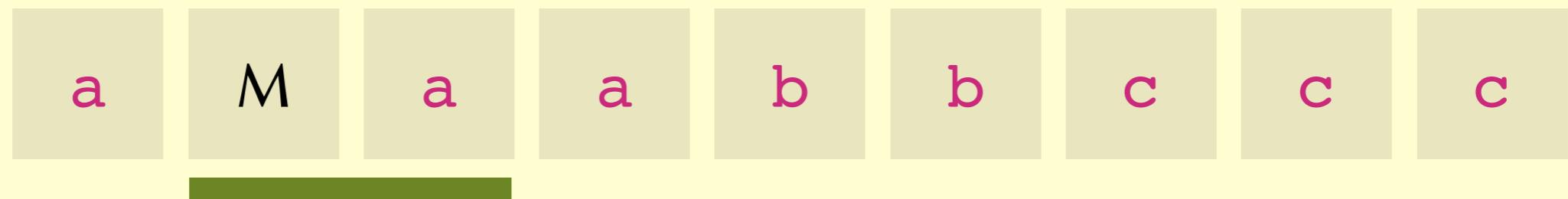
$L \rightarrow aMc$

$\rightarrow M a \rightarrow aM$

$M c \rightarrow bc$

$M b \rightarrow bb$

$\{ a^n b^n c^n \mid n > 0 \}$



# Context-sensitive grammars

$L \rightarrow aMLc$

$L \rightarrow aMc$

$Ma \rightarrow aM$

$Mc \rightarrow bc$

→  $Mb \rightarrow bb$

$\{ a^n b^n c^n \mid n > 0 \}$



# Context-sensitive grammars

$L \rightarrow aMLc$

$L \rightarrow aMc$

$Ma \rightarrow aM$

$Mc \rightarrow bc$

→  $Mb \rightarrow bb$

$\{ a^n b^n c^n \mid n > 0 \}$



# Context-sensitive grammars

$L \rightarrow aMLc$

$L \rightarrow aMc$

$\rightarrow Ma \rightarrow aM$

$Mc \rightarrow bc$

$Mb \rightarrow bb$

$\{ a^n b^n c^n \mid n > 0 \}$



# Context-sensitive grammars

$\{ a^n b^n c^n \mid n > 0 \}$

$L \rightarrow a M L c$   
 $L \rightarrow a M c$   
 $\rightarrow M a \rightarrow a M$   
 $M c \rightarrow b c$   
 $M b \rightarrow b b$



# Context-sensitive grammars

$L \rightarrow aMLc$

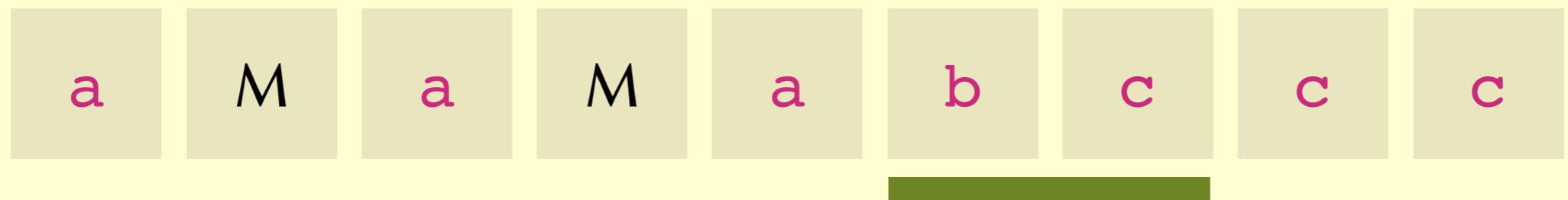
$L \rightarrow aMc$

$Ma \rightarrow aM$

→  $Mc \rightarrow bc$

$Mb \rightarrow bb$

$\{ a^n b^n c^n \mid n > 0 \}$



# Context-sensitive grammars

$L \rightarrow aMLc$

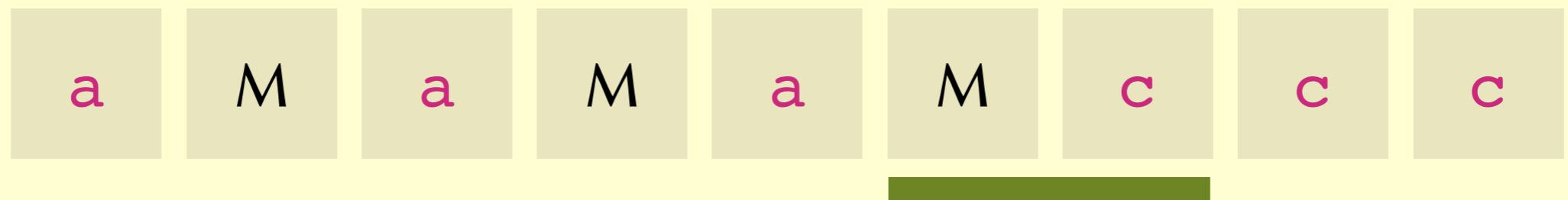
$L \rightarrow aMc$

$Ma \rightarrow aM$

→  $Mc \rightarrow bc$

$Mb \rightarrow bb$

$\{ a^n b^n c^n \mid n > 0 \}$



# Context-sensitive grammars

$\{ a^n b^n c^n \mid n > 0 \}$

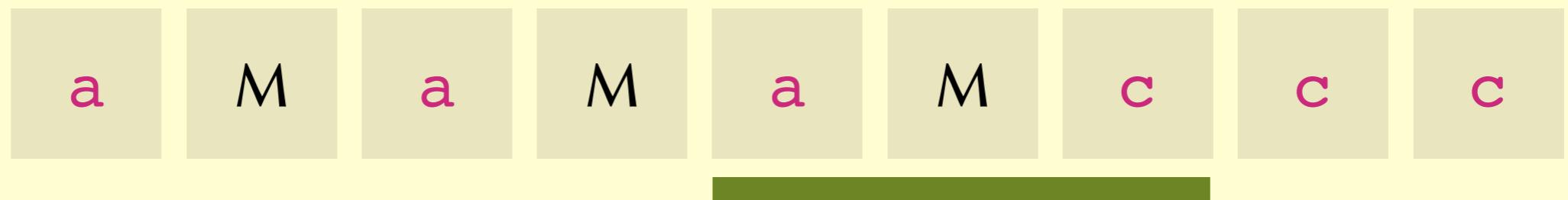
$L \rightarrow aMLc$

$\rightarrow L \rightarrow aMc$

$Ma \rightarrow aM$

$Mc \rightarrow bc$

$Mb \rightarrow bb$



# Context-sensitive grammars

$\{ a^n b^n c^n \mid n > 0 \}$

$L \rightarrow aMLc$

$\rightarrow L \rightarrow aMc$

$Ma \rightarrow aM$

$Mc \rightarrow bc$

$Mb \rightarrow bb$



# Context-sensitive grammars

  $L \rightarrow aM Lc$        $\{ a^n b^n c^n \mid n > 0 \}$   
 $L \rightarrow aMc$   
 $Ma \rightarrow aM$   
 $Mc \rightarrow bc$   
 $Mb \rightarrow bb$



# Context-sensitive grammars

→ L → aM<sub>L</sub>c

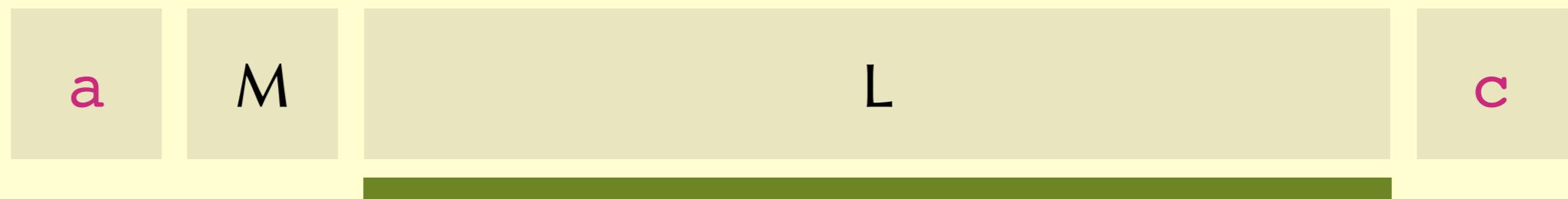
L → aMc

Ma → aM

Mc → bc

Mb → bb

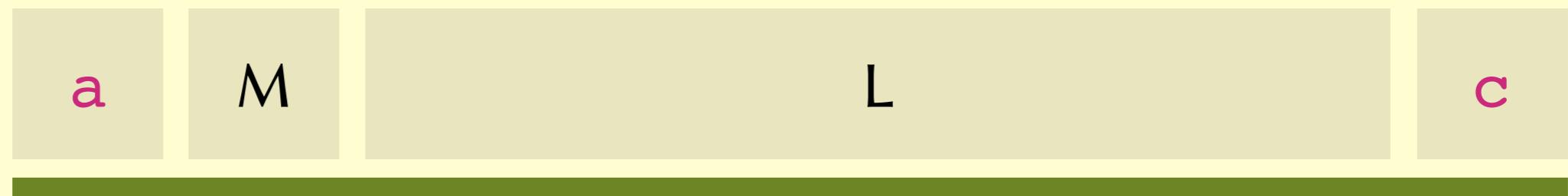
$\{ a^n b^n c^n \mid n > 0 \}$



# Context-sensitive grammars

→ L → aMLc  
L → aMc  
Ma → aM  
Mc → bc  
Mb → bb

$\{ a^n b^n c^n \mid n > 0 \}$



# Context-sensitive grammars

→ L → aM<sub>L</sub>c  
L → aMc  
Ma → aM  
Mc → bc  
Mb → bb

$\{ a^n b^n c^n \mid n > 0 \}$

L

# Hierarchy of languages

unrestricted grammar (Chomsky Type-0)

e.g.,  $Ma \rightarrow a$

...

context-sensitive grammar (Chomsky Type-1)

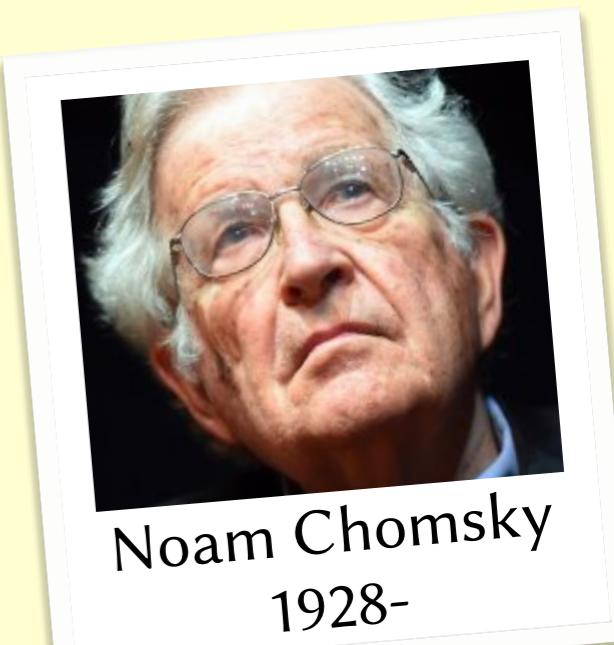
$\{ a^n b^n c^n \mid n > 0 \}$

context-free grammar (Chomsky Type-2)

$\{ a^n b^n \mid n \geq 0 \}$

regular language (Chomsky Type-3)

$\{ a^n \mid n \text{ is odd} \}$



# What we know so far

- Languages can be defined using a **grammar** made up of **production rules** featuring **non-terminals** and **terminals**.
- Grammars should be written to avoid **ambiguity**.
- **Regular grammars** are equivalent to regexes.
- **Context-free grammars** are more expressive than regular grammars.
- And **context-sensitive grammars** are more expressive still.
- Most programming languages can be defined using a grammar that is (more or less) context-free.