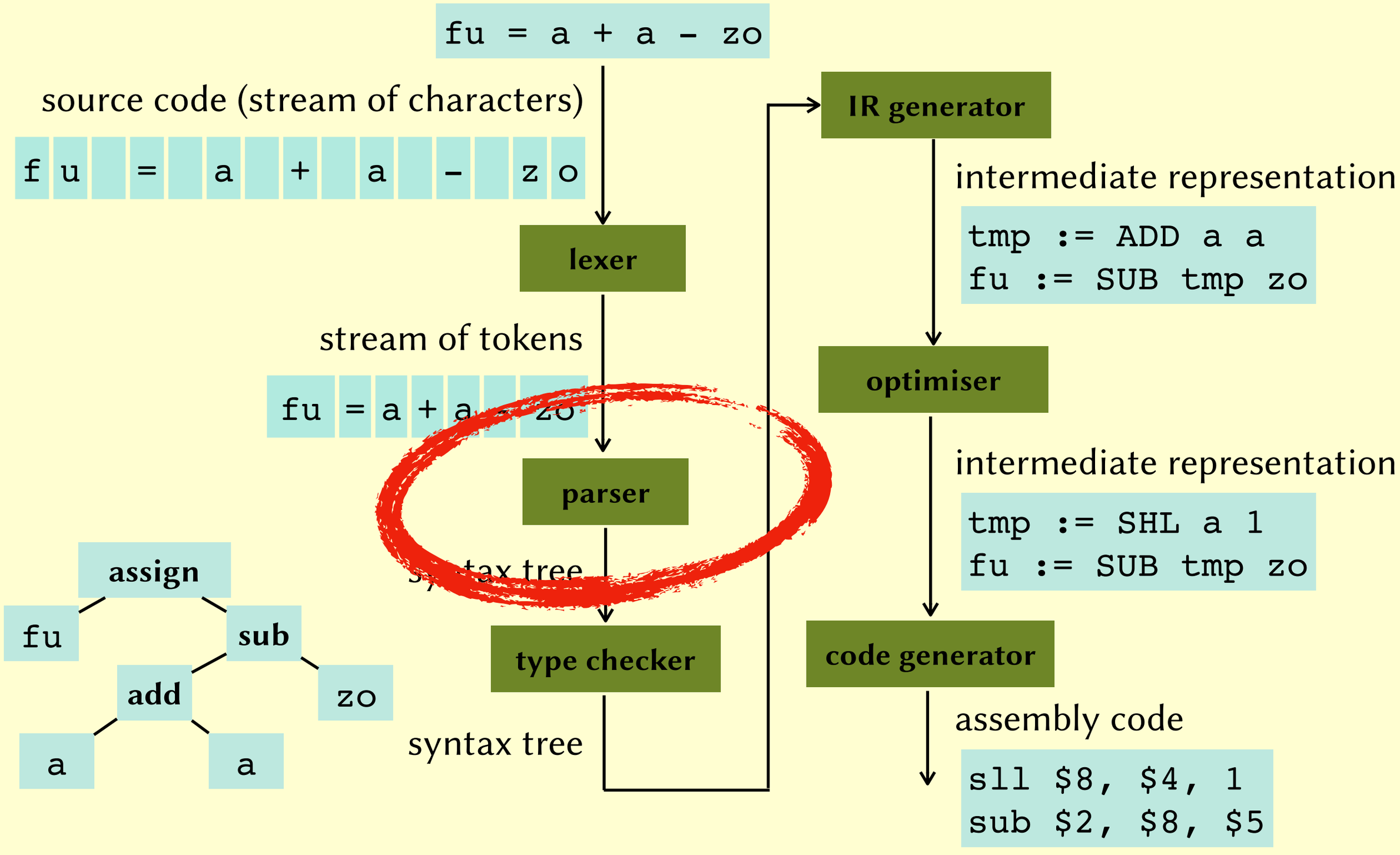
Lecture 4: Parsing

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 prepphrase

verbphrase → verb

verbphrase → verb nounphrase

verbphrase → verb prepphrase

Grammars

sentence ::= nounphrase verbphrase

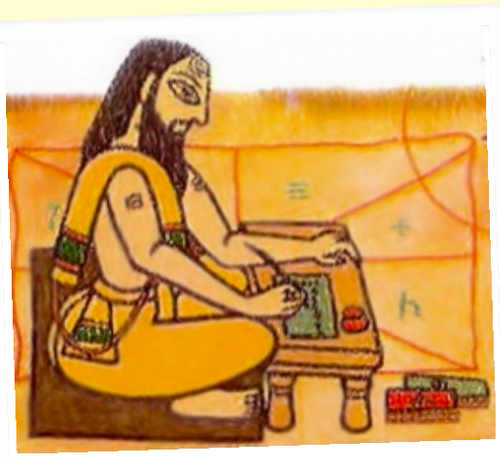
nounphrase ::= det noun | det noun prepphrase

verbphrase ::= verb | verb nounphrase | verb prepphrase

John Backus
1924-2007

Peter Naur
1928-2016

hence, Backus-Naur form



Pāṇini
ca. 500 BC

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`

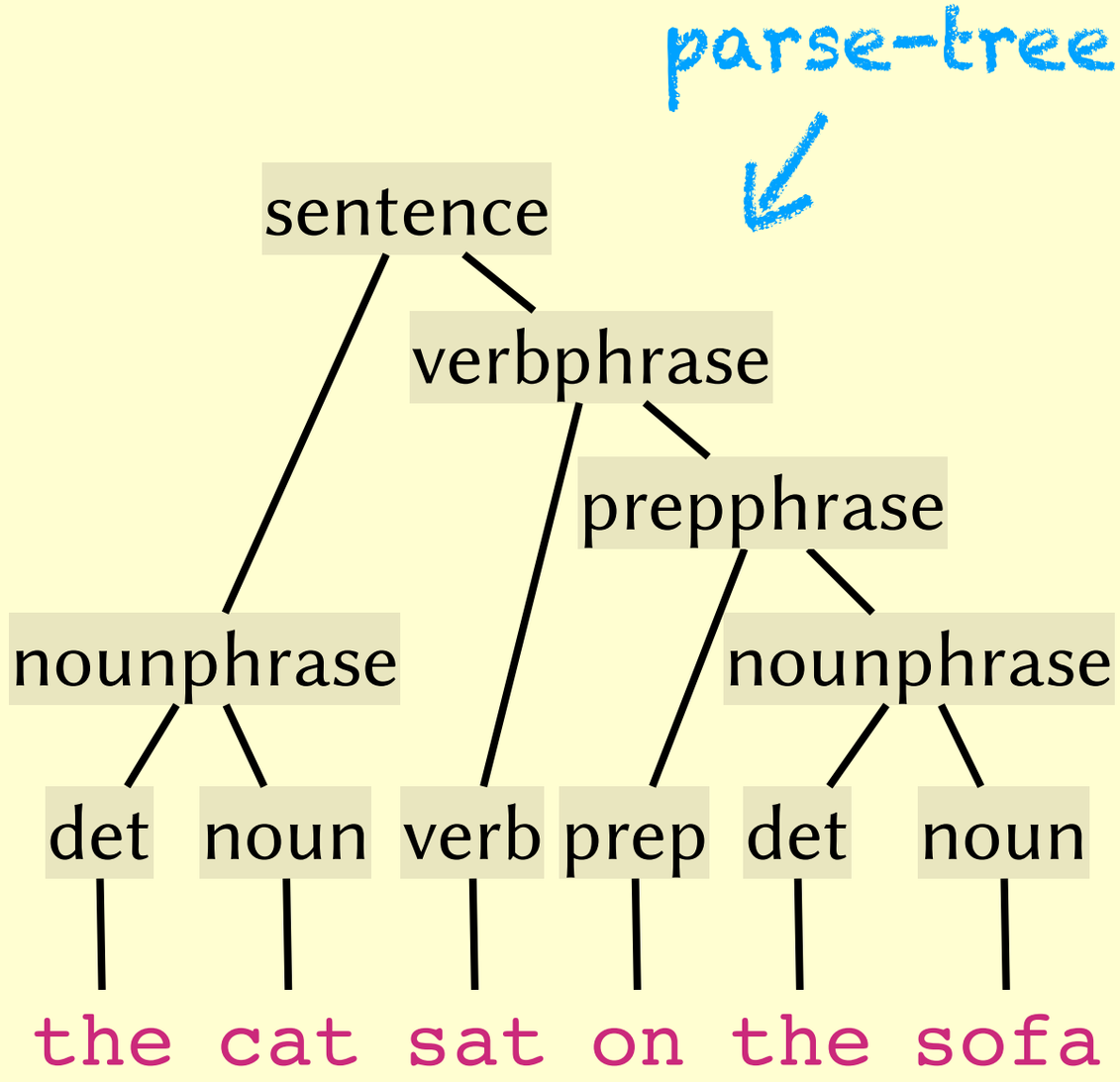
verb ::= `sat` | `ate`

prep ::= `on` | `with`

non-terminal

terminal

parse-tree



Grammars

sentence ::= nounphrase verbphrase

nounphr

verbphra

prepphra

noun

det

verb

prep

se

phrase

nounphrase

det noun

the cat sat on the sofa

Grammars

sentence ::= nounphrase verbphrase

nounphrase ::= det noun | det noun prepphrase

verbphrase ::= verb | verb nounphrase | verb prepphrase

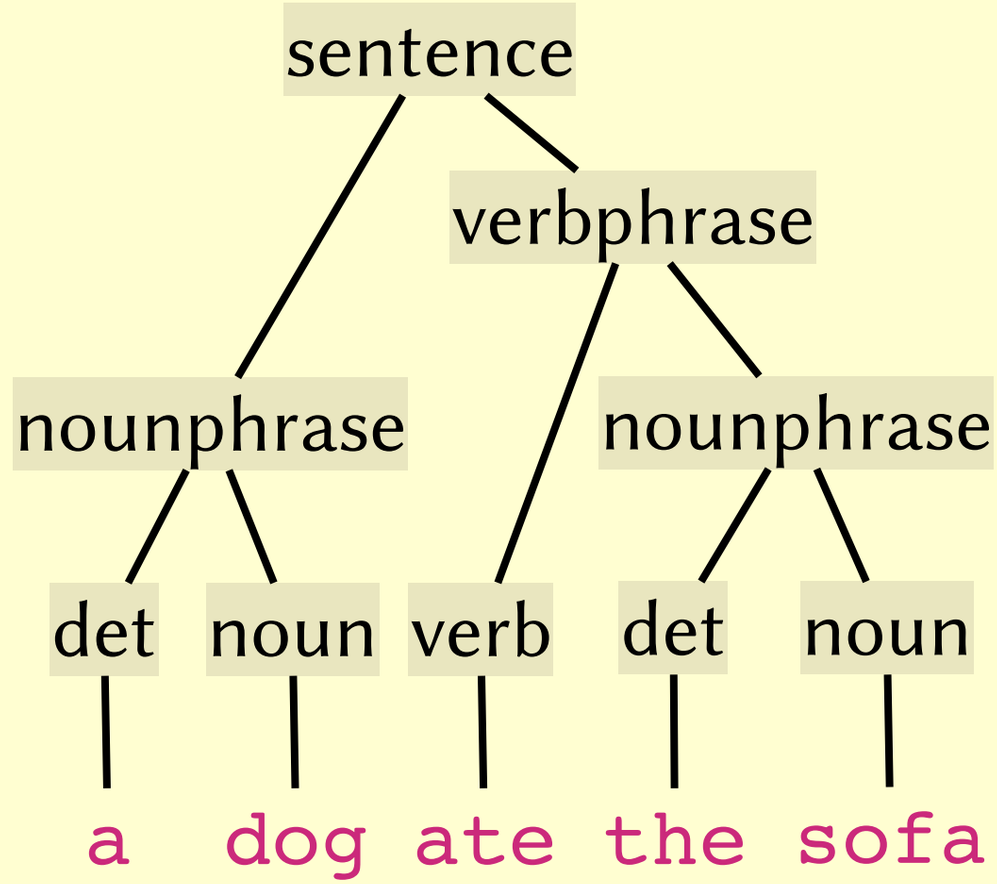
prepphrase ::= prep nounphrase

noun ::= `cat` | `dog` | `sofa`

det ::= `a` | `the`

verb ::= `sat` | `ate`

prep ::= `on` | `with`



Grammars

sentence → nounphrase verbphrase

nounphrase → det noun

verbphrase → verb nounphrase

prepphrase → prep nounphrase

noun → dog | sofa

det → a | the

verb → ate

prep → on

Grammars

sentence ::= nounphrase verbphrase

nounphrase ::= det noun | det noun prepphrase

verbphrase ::= verb | verb nounphrase | verb prepphrase

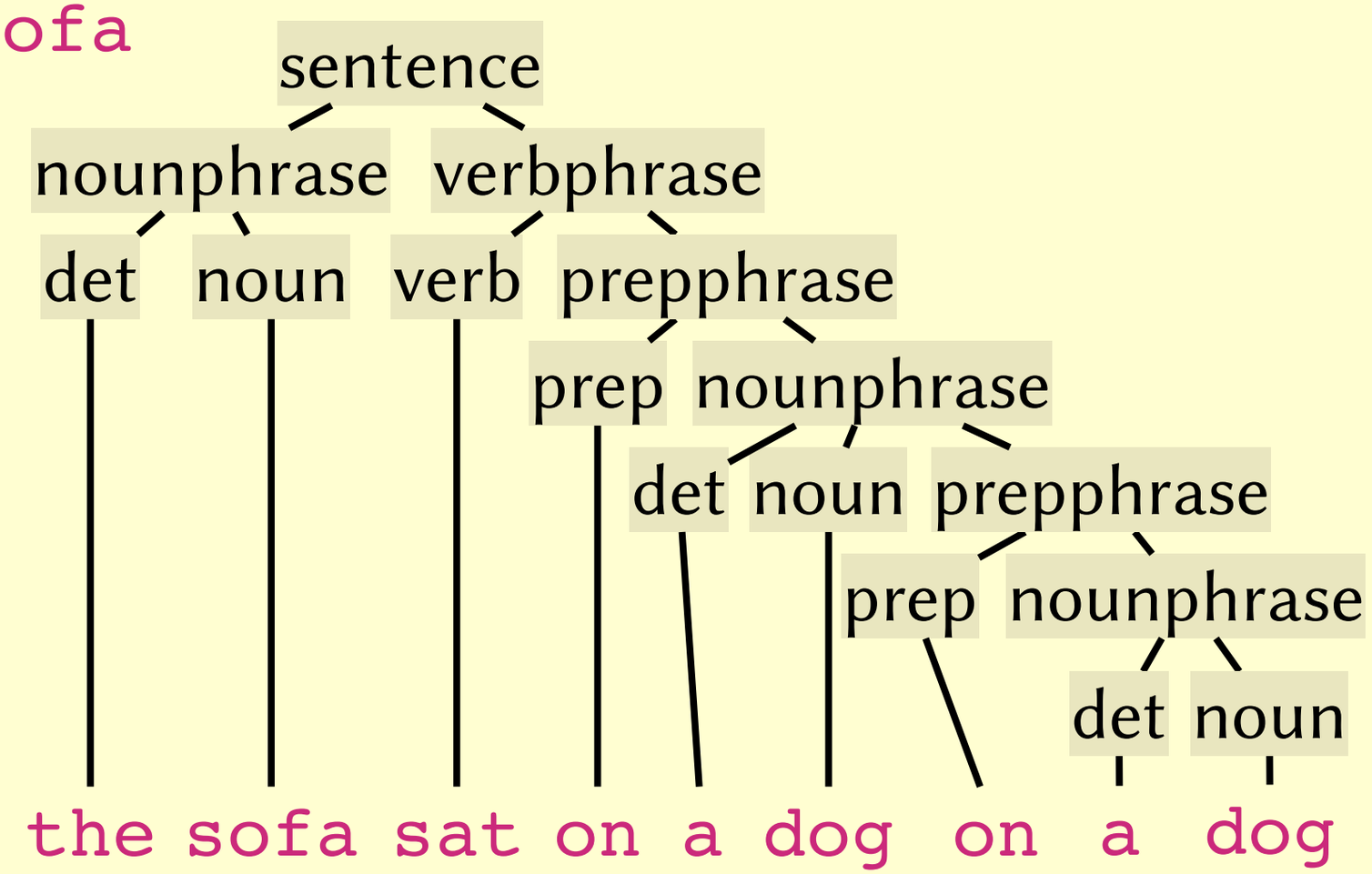
prepphrase ::= prep nounphrase

noun ::= `cat` | `dog` | `sofa`

det ::= `a` | `the`

verb ::= `sat` | `ate`

prep ::= `on` | `with`



...mars

sentence ::

nounphrase ::

verbphrase :

prepphrase

noun

det

verb

prep

...ase

...ase

...pphrase

nounphrase

det noun prepphrase

prep nounphrase

det noun

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

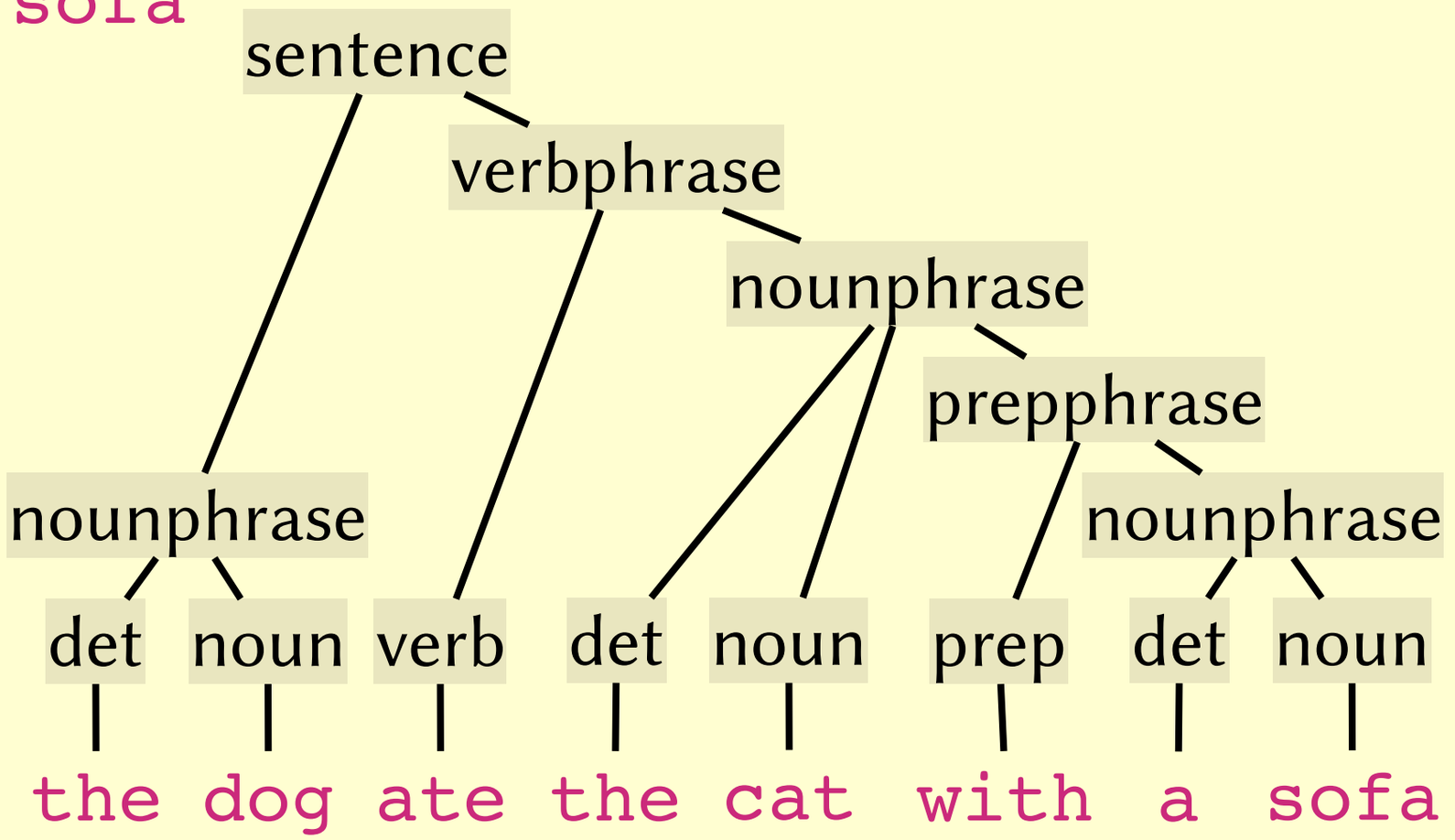
prepphrase ::= prep nounphrase

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

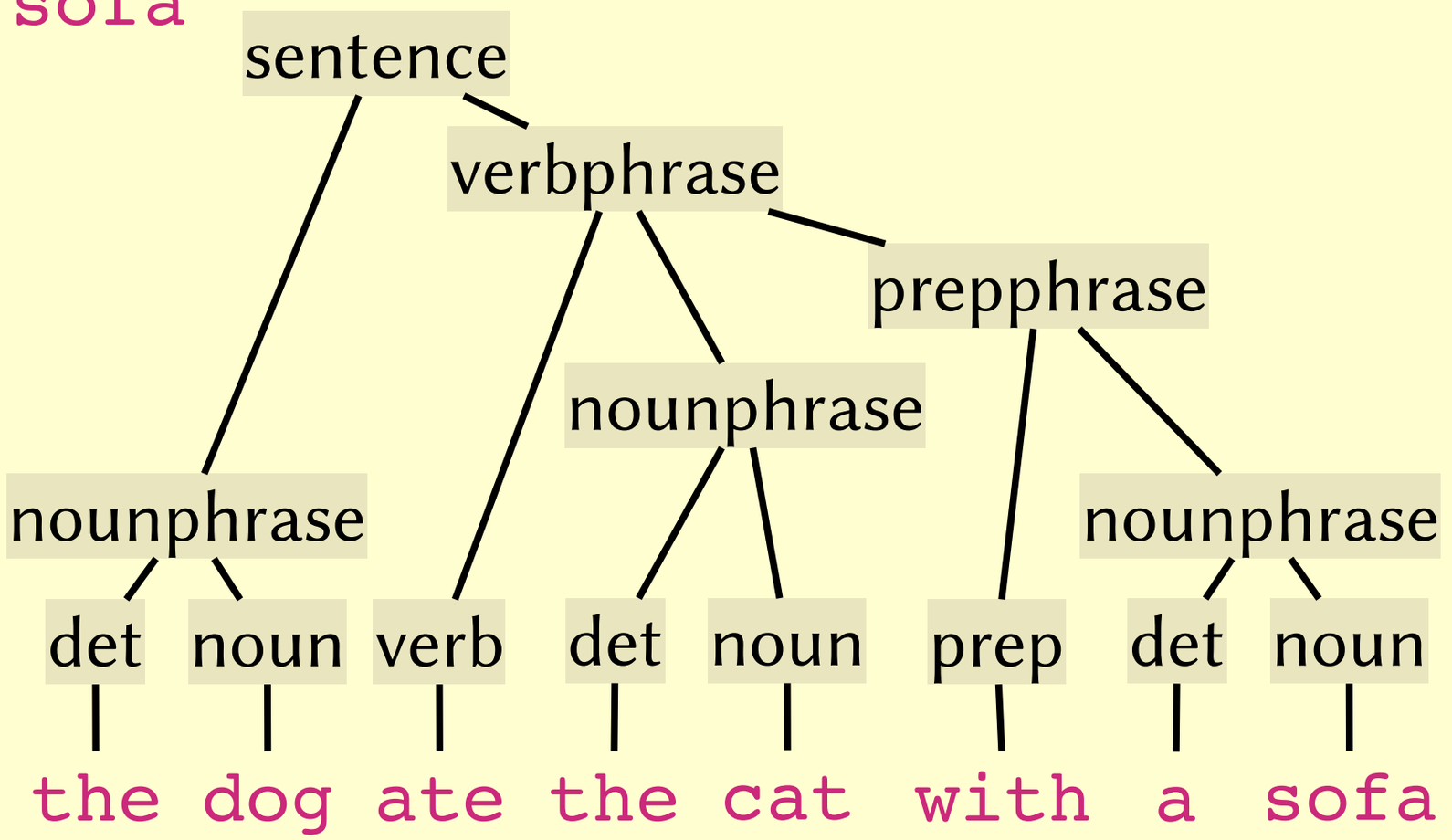
prepphrase ::= prep nounphrase

noun ::= `cat` | `dog` | `sofa`

det ::= `a` | `the`

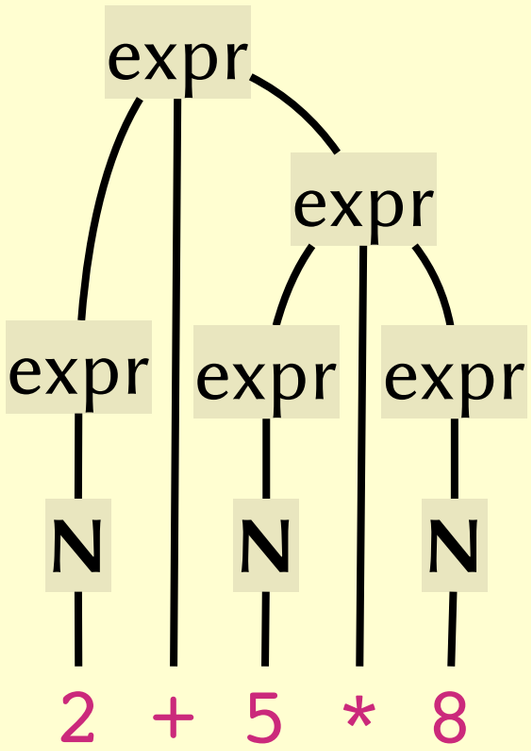
verb ::= `sat` | `ate`

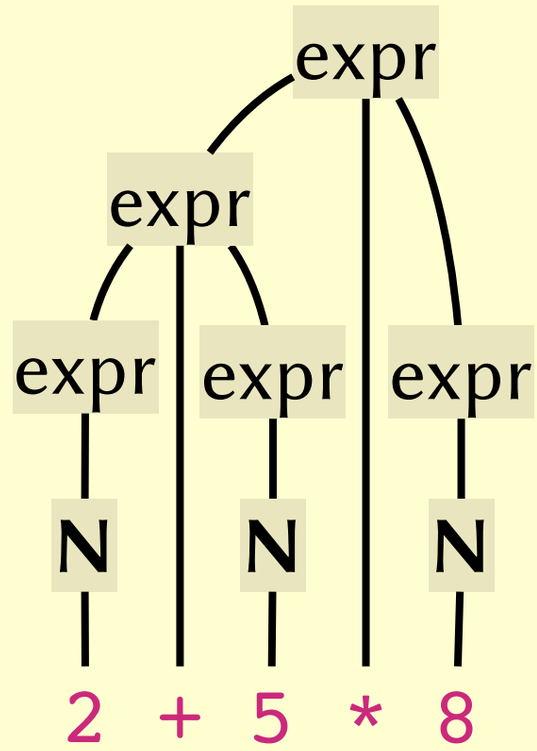
prep ::= `on` | `with`



Expressions

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



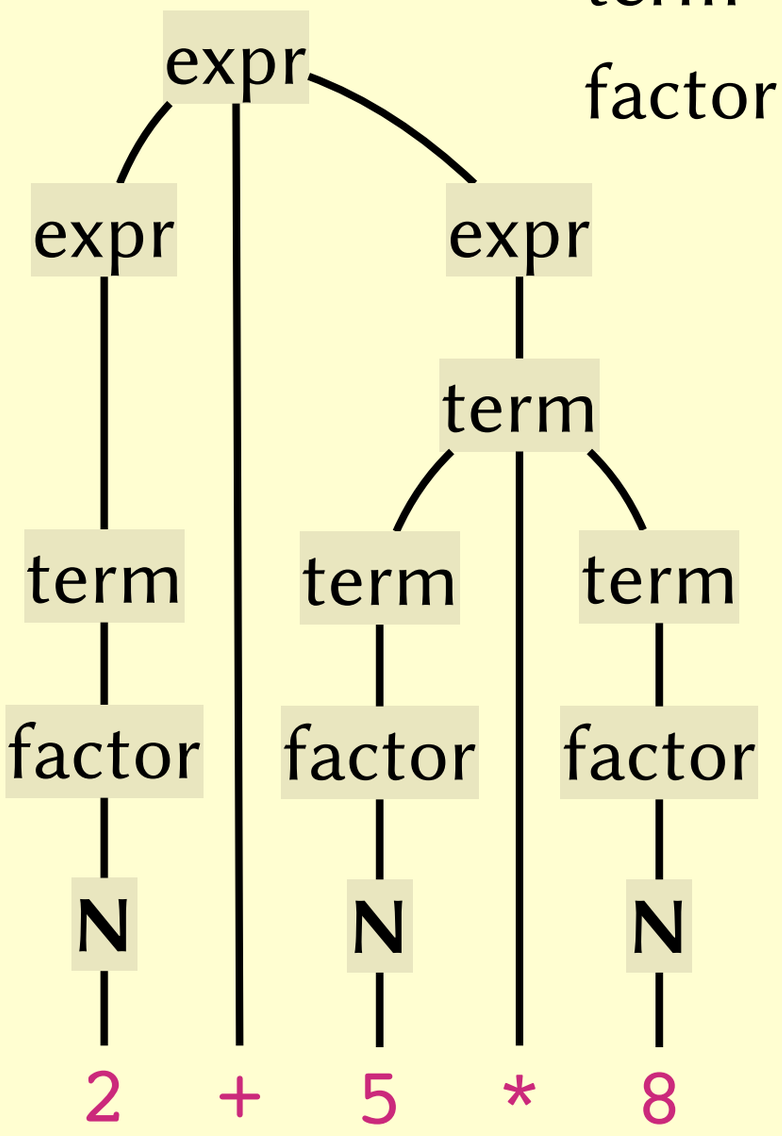


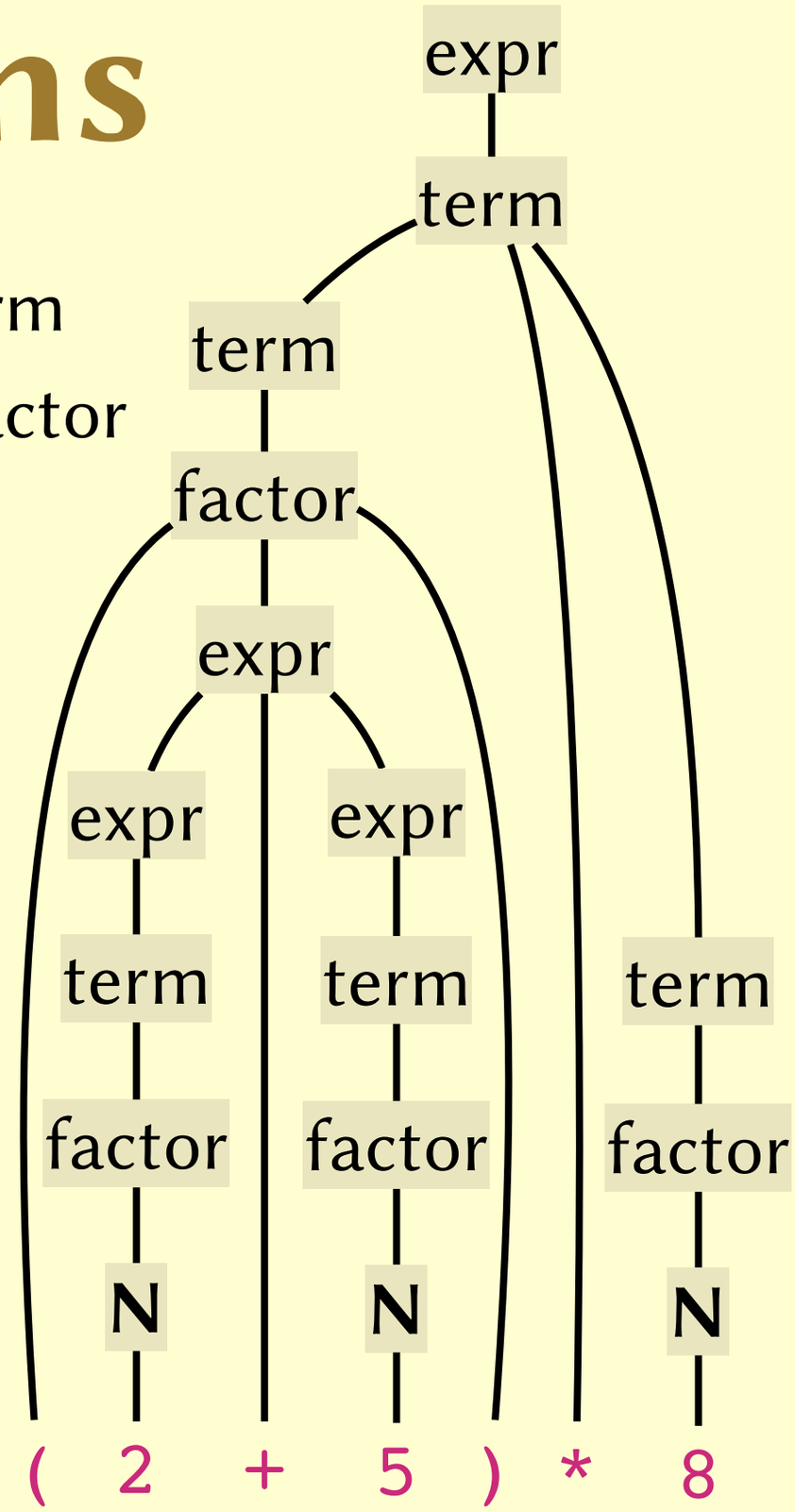
Expressions

expr ::= expr + expr | term

term ::= term * term | factor

factor ::= (expr) | **N**



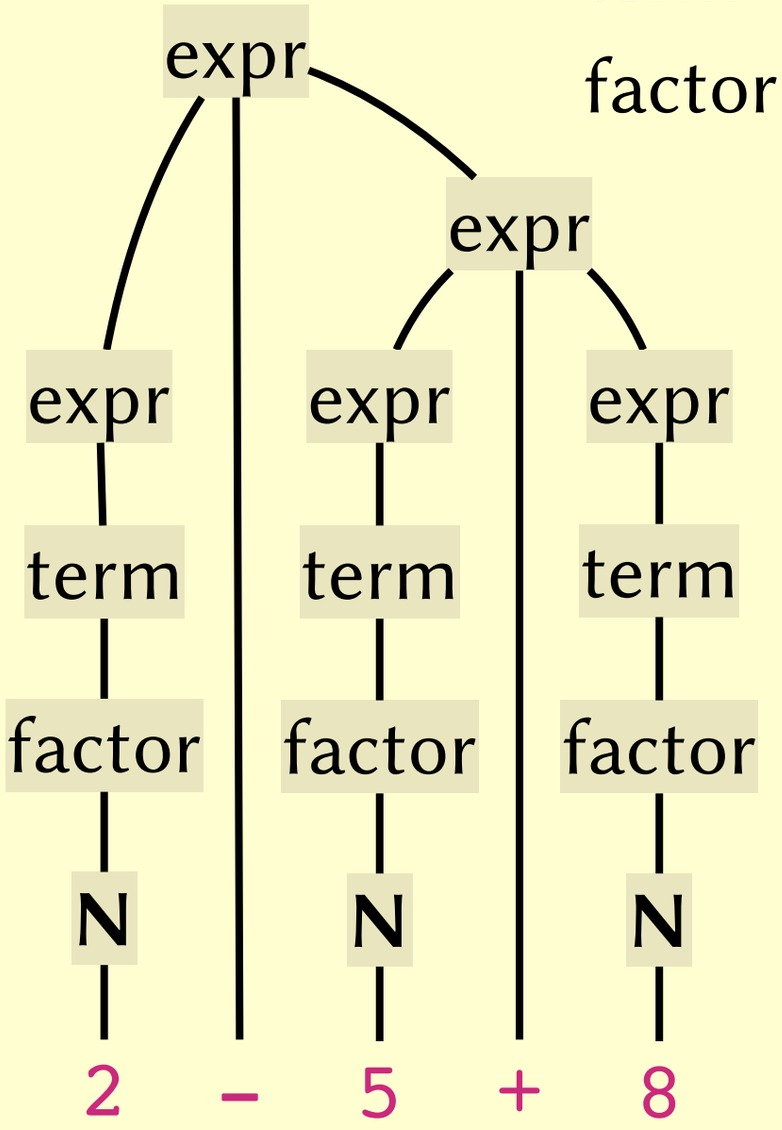


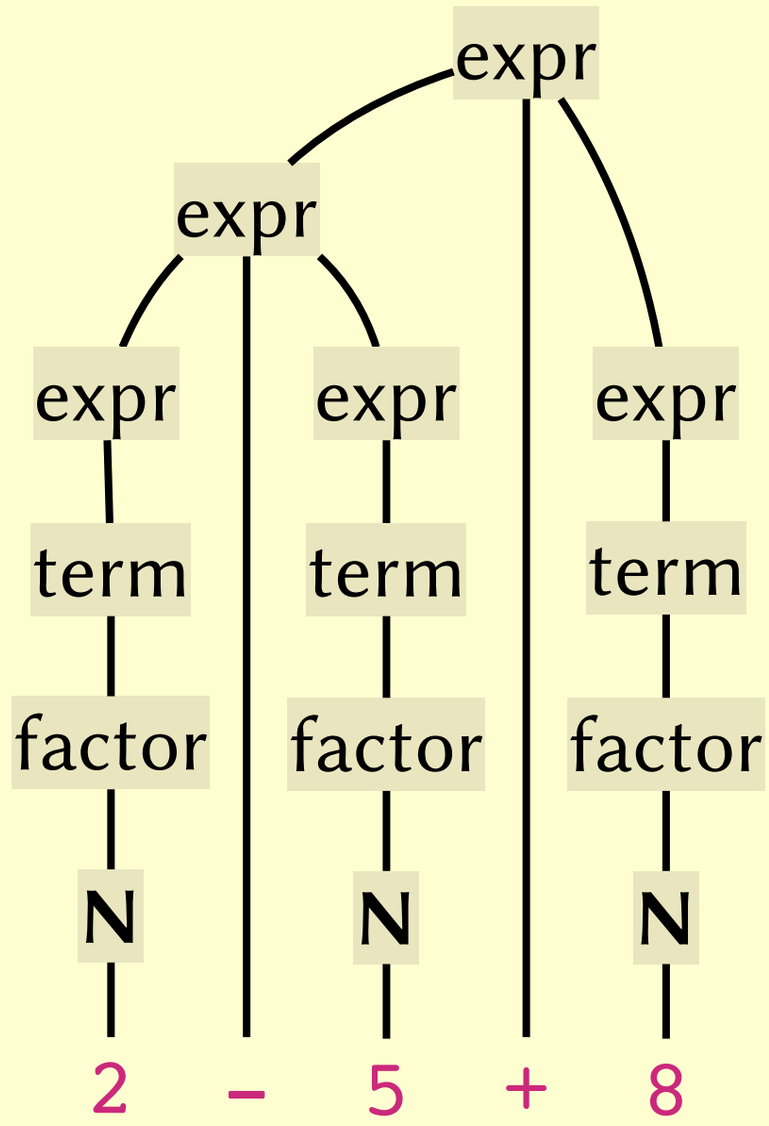
Expressions

expr ::= expr + expr | expr − expr | term

term ::= term * term | factor

factor ::= (expr) | **N**



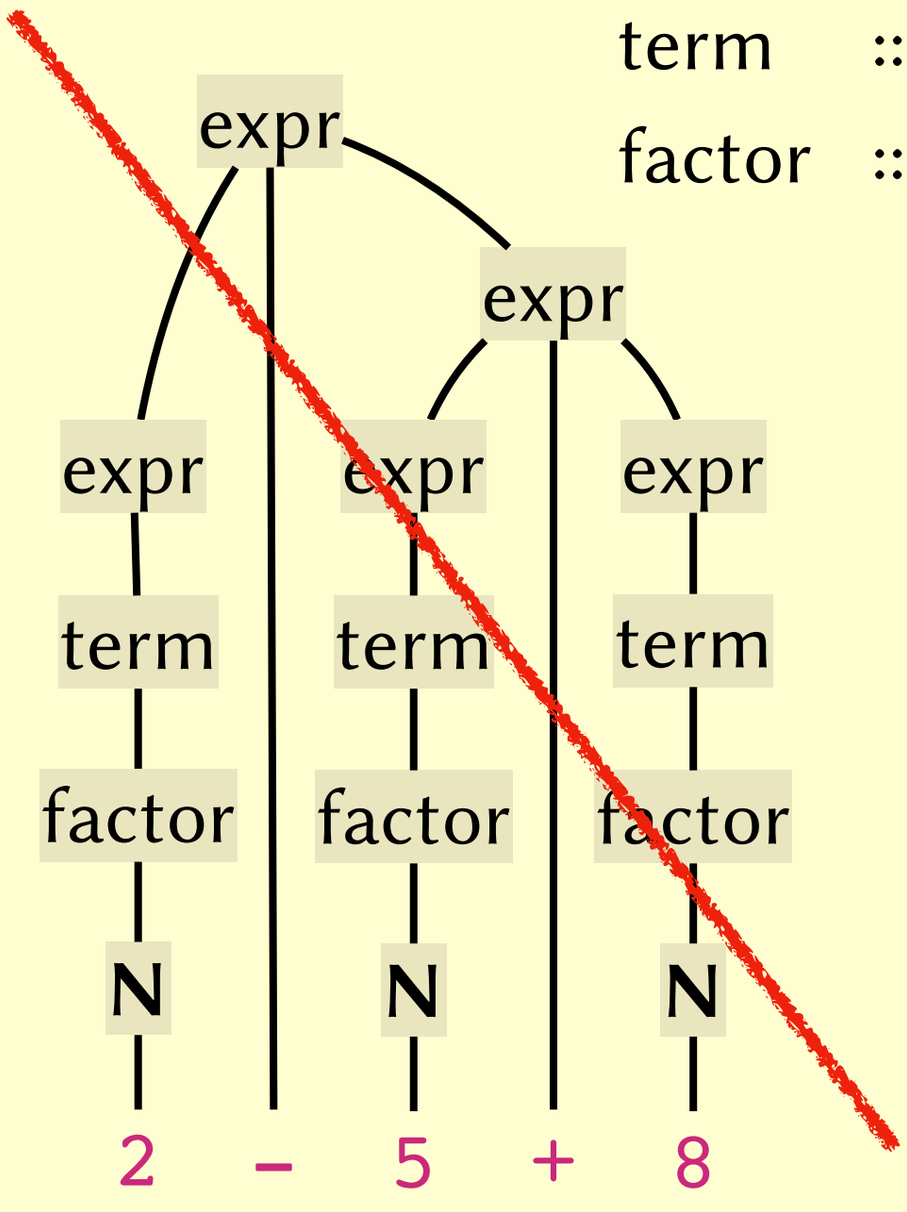


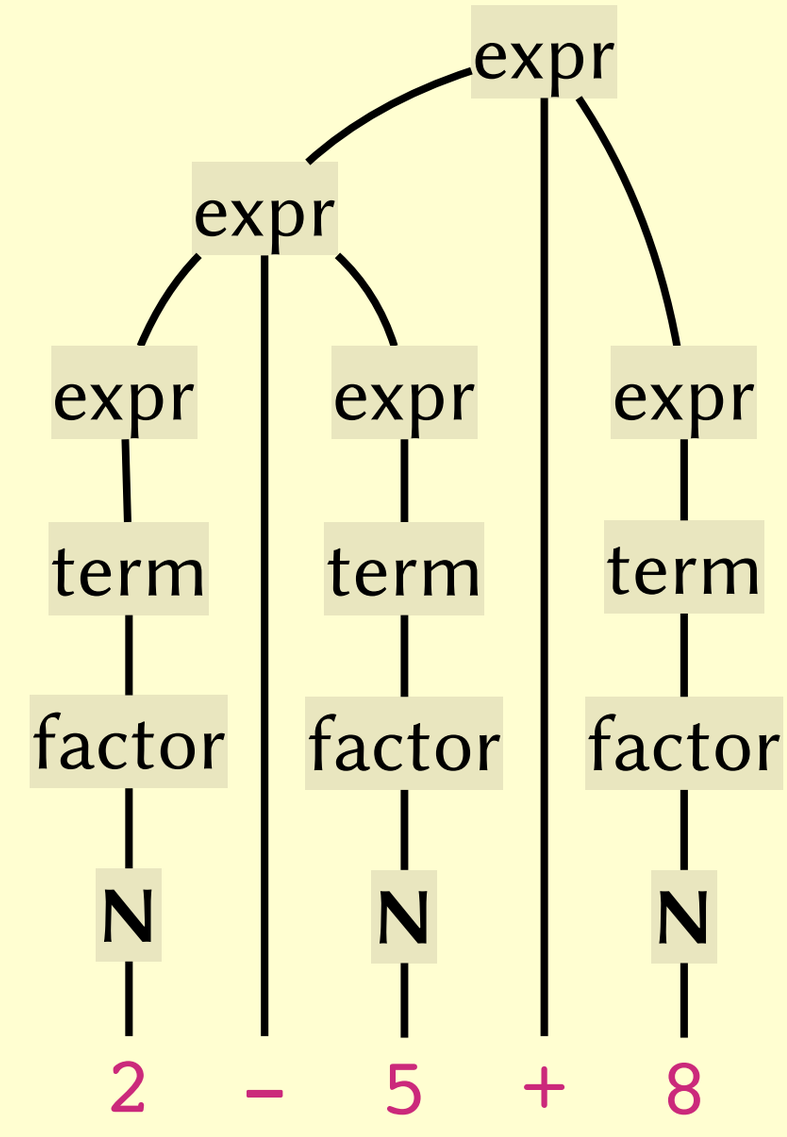
Expressions

expr ::= expr + term | expr – term | term

term ::= term * factor | factor

factor ::= (expr) | **N**



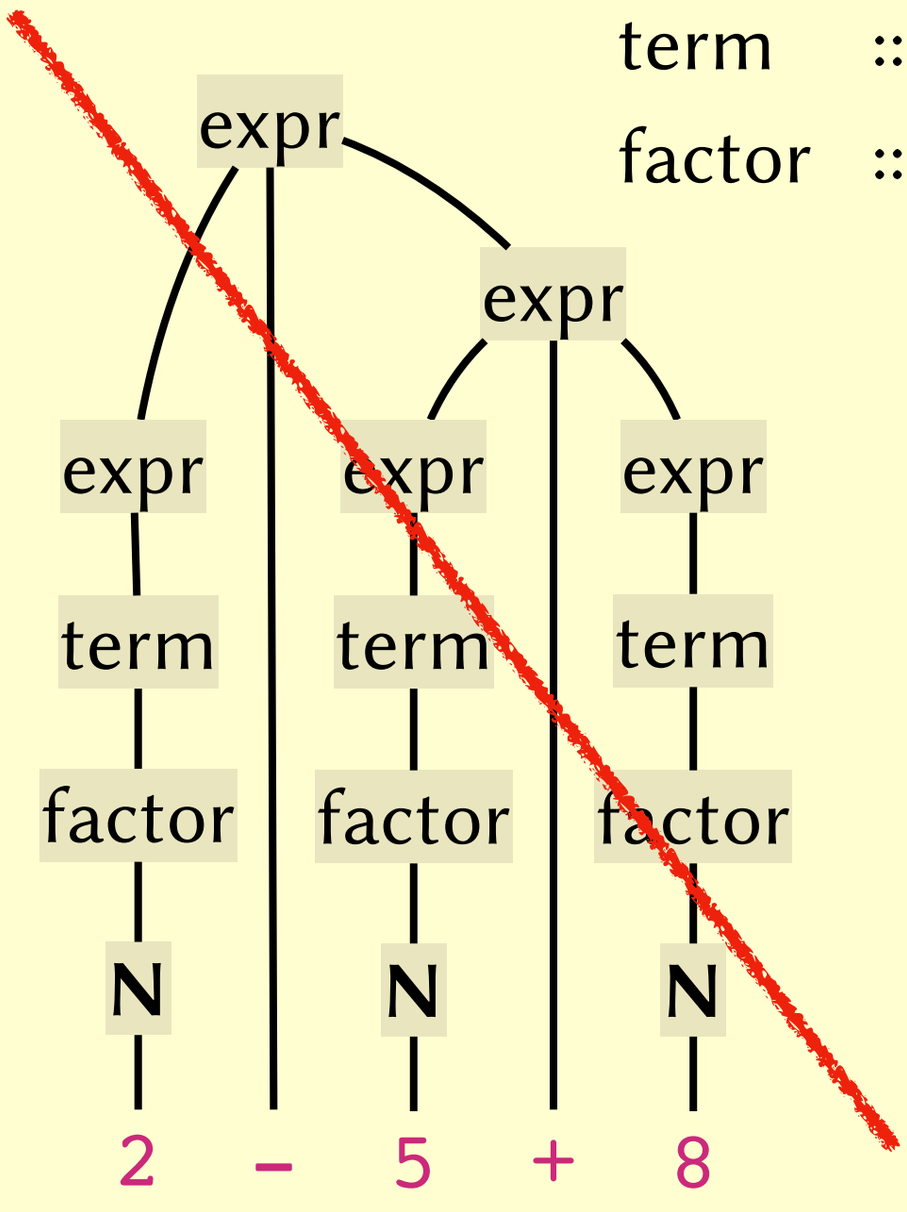


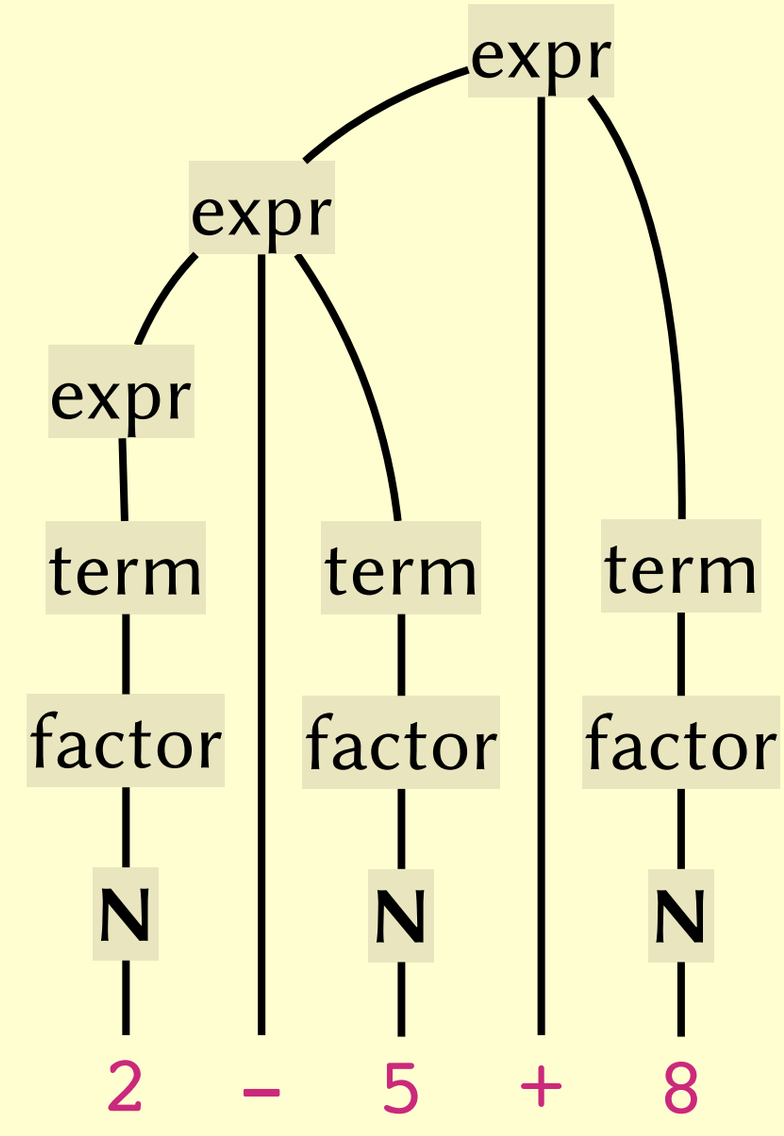
Expressions

expr ::= expr + term | expr – term | term

term ::= term * factor | factor

factor ::= (expr) | **N**





Grammars vs. regexes

L ::= aaL | a

Grammars vs. regexes

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

L ::= aaL | a

Grammars vs. regexes

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

L ::= aaL | a

a(aa)*

Grammars vs. regexes

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

Grammars vs. regexes

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

Grammars vs. regexes

Set of words in the language	Grammar	Regex
$\{ a^n \mid n \text{ is odd} \}$	L ::= aaL | a	a(aa)*
no consecutive bs	L ::= aL | abL | ε	

Grammars vs. regexes

Set of words in the language	Grammar	Regex
$\{ a^n \mid n \text{ is odd} \}$	L ::= aaL | a	a(aa)*
no consecutive bs	L ::= aL | abL | ε	(ab?)*

Grammars vs. regexes

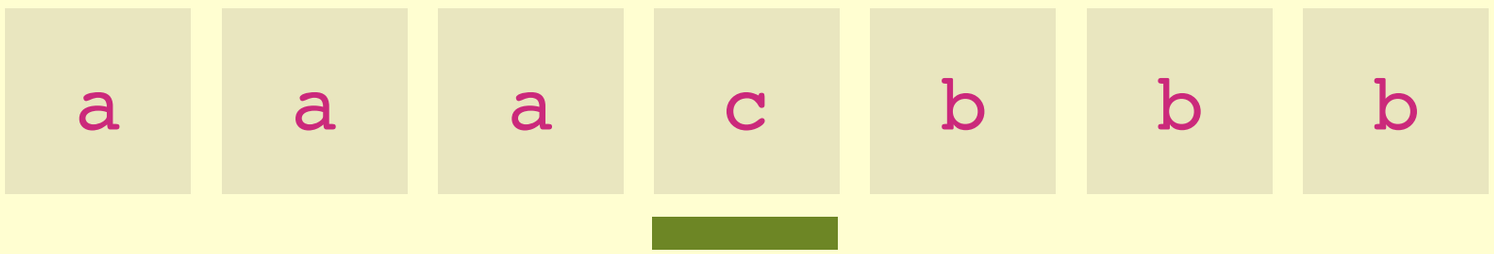
Set of words in the language	Grammar	Regex
$\{ a^n \mid n \text{ is odd} \}$	L ::= aaL | a	a(aa)*
no consecutive bs	L ::= aL | abL | ε	(ab?)*
$\{ a^n c b^n \mid n \geq 0 \}$		

Grammars vs. regexes

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

$$L \rightarrow aLb$$
$$\Rightarrow L \rightarrow c$$

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



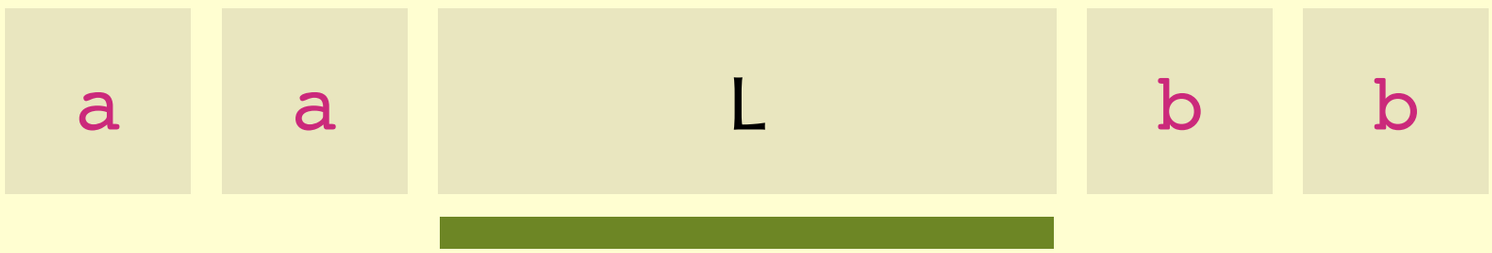
L → aLb

L → c

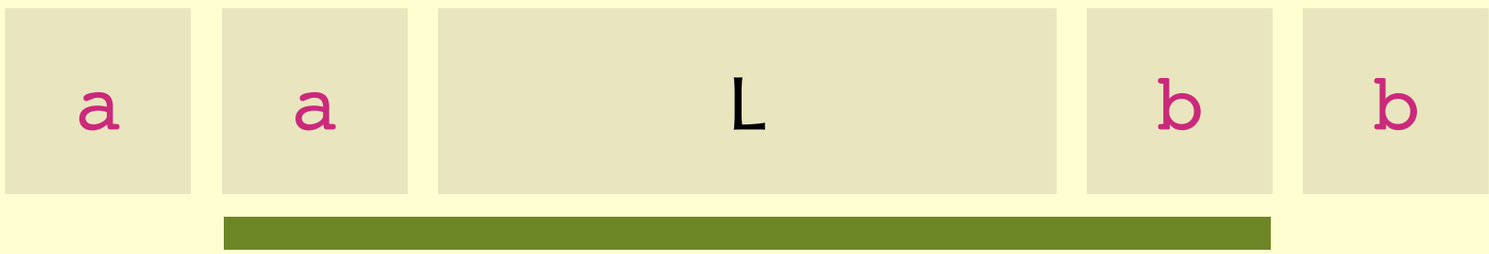
L → aLb

L → c

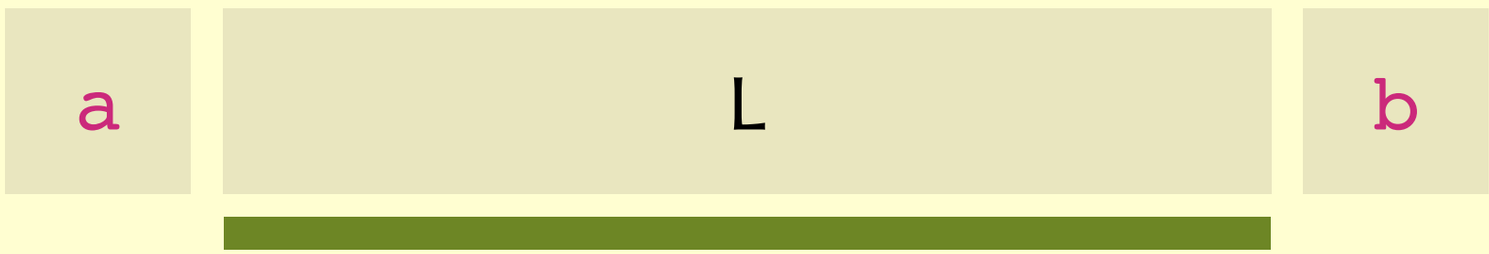
L → aLb
L → c



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

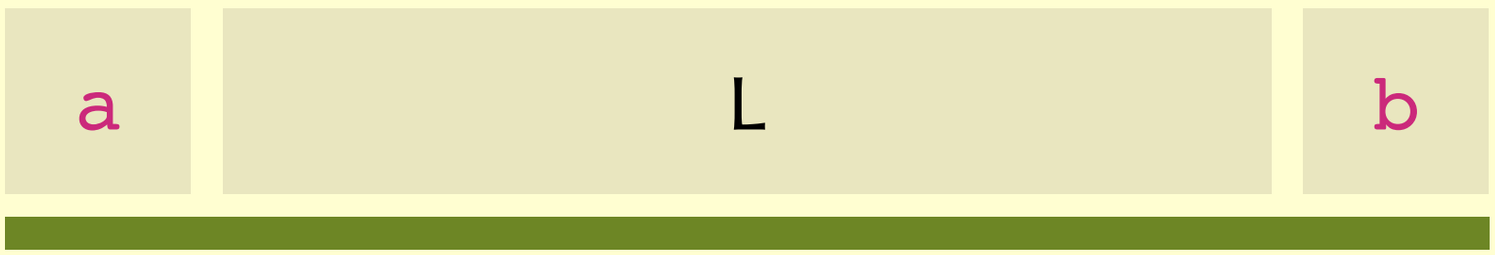


$$L \rightarrow aLb$$
$$L \rightarrow c$$



L → aLb

L → c

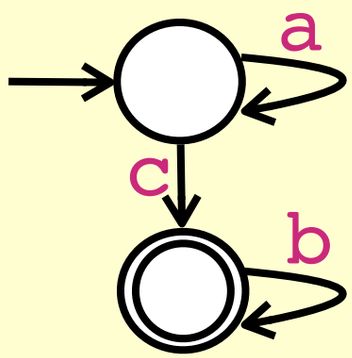


$$L \rightarrow aLb$$
$$L \rightarrow c$$

L

Grammars vs. regexes

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

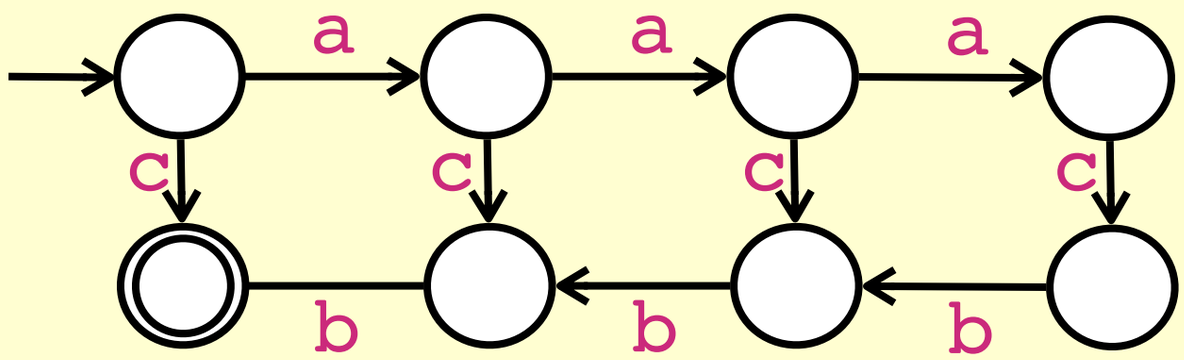


Grammars vs. regexes

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

not a "regular" language

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



$L ::= aLb \mid c$ ✗

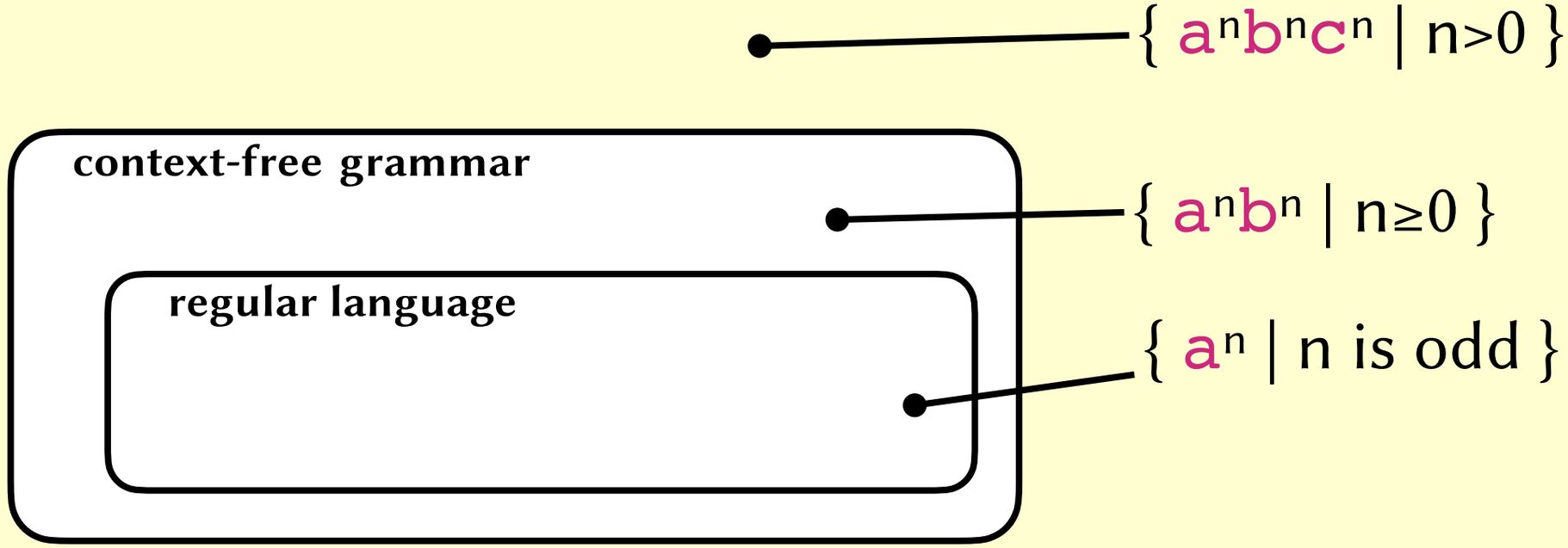
not a "regular" grammar

recall expr ::= ... | (expr)

Grammars vs. regexes

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

Hierarchy of languages



Context-sensitive grammars

Note: |LHS| ≤ |RHS|

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

$$
\begin{aligned}
L &\rightarrow aMLc \\
L &\rightarrow aMc \\
Ma &\rightarrow aM \\
Mc &\rightarrow bc \\
Mb &\rightarrow bb
\end{aligned}
$$

a	a	a	b	b	b	c	c	c

Context-sensitive grammars

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

$$
\begin{aligned}
L &\longrightarrow aMLc \\
L &\longrightarrow aMc \\
Ma &\longrightarrow aM \\
Mc &\longrightarrow bc \\
Mb &\longrightarrow bb
\end{aligned}
$$

Context-sensitive grammars

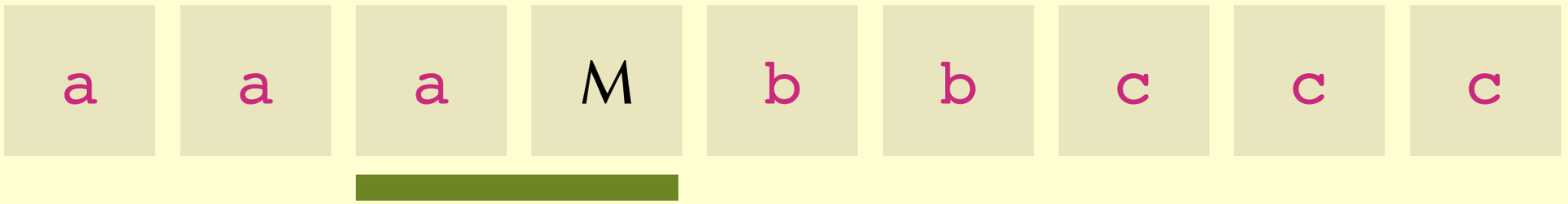
$\{ a^n b^n c^n \mid n>0 \}$

$$
\begin{aligned}
L &\longrightarrow aMLc \\
L &\longrightarrow aMc \\
Ma &\longrightarrow aM \\
Mc &\longrightarrow bc \\
Mb &\longrightarrow bb
\end{aligned}
$$

Context-sensitive grammars

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

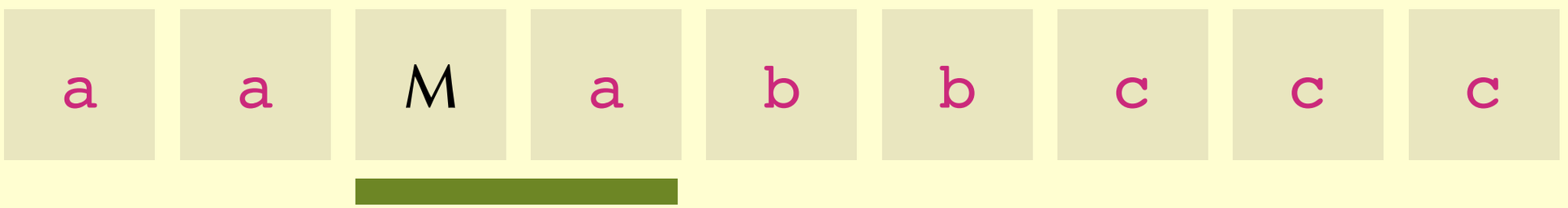
$$
\begin{aligned}
L &\to aMLc \\
L &\to aMc \\
Ma &\to aM \\
Mc &\to bc \\
Mb &\to bb
\end{aligned}
$$



Context-sensitive grammars

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

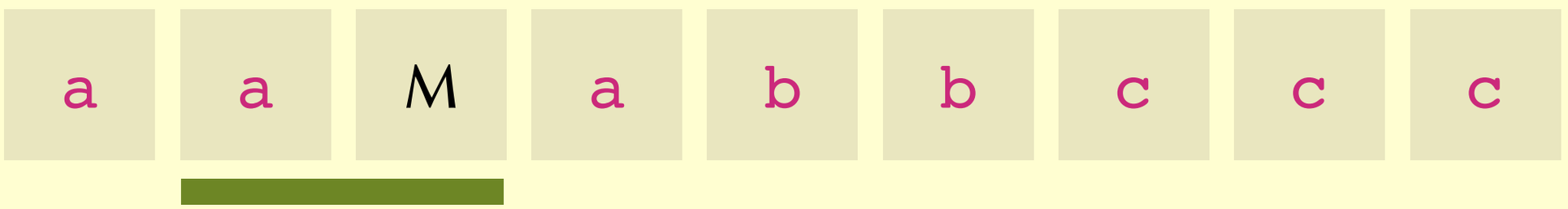
$$
\begin{aligned}
L &\to aMLc \\
L &\to aMc \\
Ma &\to aM \\
Mc &\to bc \\
Mb &\to bb
\end{aligned}
$$



Context-sensitive grammars

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

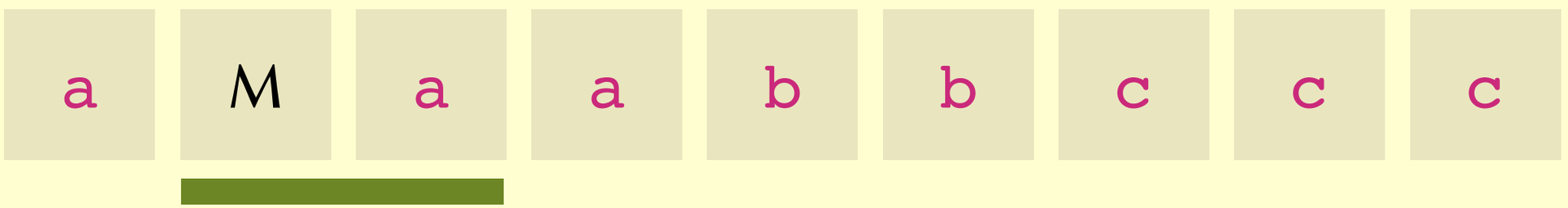
$$
\begin{aligned}
L &\rightarrow aMLc \\
L &\rightarrow aMc \\
Ma &\rightarrow aM \\
Mc &\rightarrow bc \\
Mb &\rightarrow bb
\end{aligned}
$$



Context-sensitive grammars

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

$$
\begin{aligned}
L &\longrightarrow aMLc \\
L &\longrightarrow aMc \\
Ma &\longrightarrow aM \\
Mc &\longrightarrow bc \\
Mb &\longrightarrow bb
\end{aligned}
$$



Context-sensitive grammars

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

$$
\begin{aligned}
L &\longrightarrow aMLc \\
L &\longrightarrow aMc \\
Ma &\longrightarrow aM \\
Mc &\longrightarrow bc \\
Mb &\longrightarrow bb
\end{aligned}
$$

Context-sensitive grammars

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

$$
\begin{aligned}
L &\longrightarrow aMLc \\
L &\longrightarrow aMc \\
Ma &\longrightarrow aM \\
Mc &\longrightarrow bc \\
Mb &\longrightarrow bb
\end{aligned}
$$

Context-sensitive grammars

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

$$
\begin{aligned}
L &\longrightarrow aMLc \\
L &\longrightarrow aMc \\
Ma &\longrightarrow aM \\
Mc &\longrightarrow bc \\
Mb &\longrightarrow bb
\end{aligned}
$$

Context-sensitive grammars

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

$$
\begin{aligned}
L &\longrightarrow aMLc \\
L &\longrightarrow aMc \\
Ma &\longrightarrow aM \\
Mc &\longrightarrow bc \\
Mb &\longrightarrow bb
\end{aligned}
$$

Context-sensitive grammars

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

$$
\begin{aligned}
L &\longrightarrow aMLc \\
L &\longrightarrow aMc \\
Ma &\longrightarrow aM \\
Mc &\longrightarrow bc \\
Mb &\longrightarrow bb
\end{aligned}
$$

Context-sensitive grammars

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

$$
\begin{aligned}
L &\longrightarrow aMLc \\
L &\longrightarrow aMc \\
Ma &\longrightarrow aM \\
Mc &\longrightarrow bc \\
Mb &\longrightarrow bb
\end{aligned}
$$

Context-sensitive grammars

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

$L \longrightarrow aMLc$

$L \longrightarrow aMc$

$Ma \longrightarrow aM$

$Mc \longrightarrow bc$

$Mb \longrightarrow bb$

Context-sensitive grammars

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

$L \longrightarrow aMLc$

$L \longrightarrow aMc$

$Ma \longrightarrow aM$

$Mc \longrightarrow bc$

$Mb \longrightarrow bb$

Context-sensitive grammars

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

$L \longrightarrow aMLc$

$L \longrightarrow aMc$

$Ma \longrightarrow aM$

$Mc \longrightarrow bc$

$Mb \longrightarrow bb$

Context-sensitive grammars

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

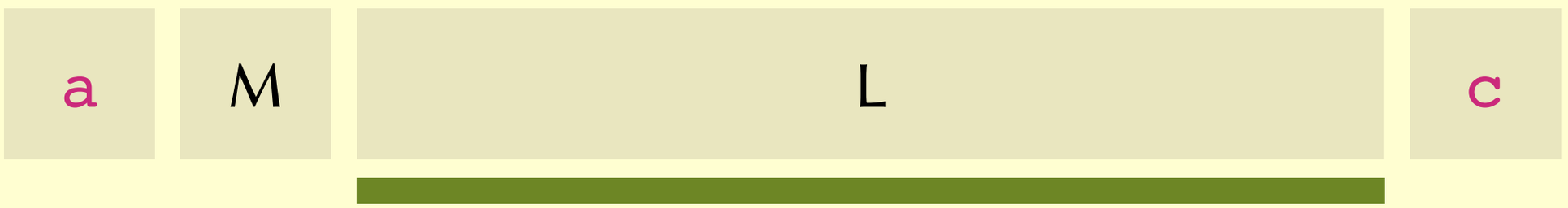
$L \longrightarrow aMLc$

$L \longrightarrow aMc$

$Ma \longrightarrow aM$

$Mc \longrightarrow bc$

$Mb \longrightarrow bb$



Context-sensitive grammars

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

$L \longrightarrow aMLc$

$L \longrightarrow aMc$

$Ma \longrightarrow aM$

$Mc \longrightarrow bc$

$Mb \longrightarrow bb$

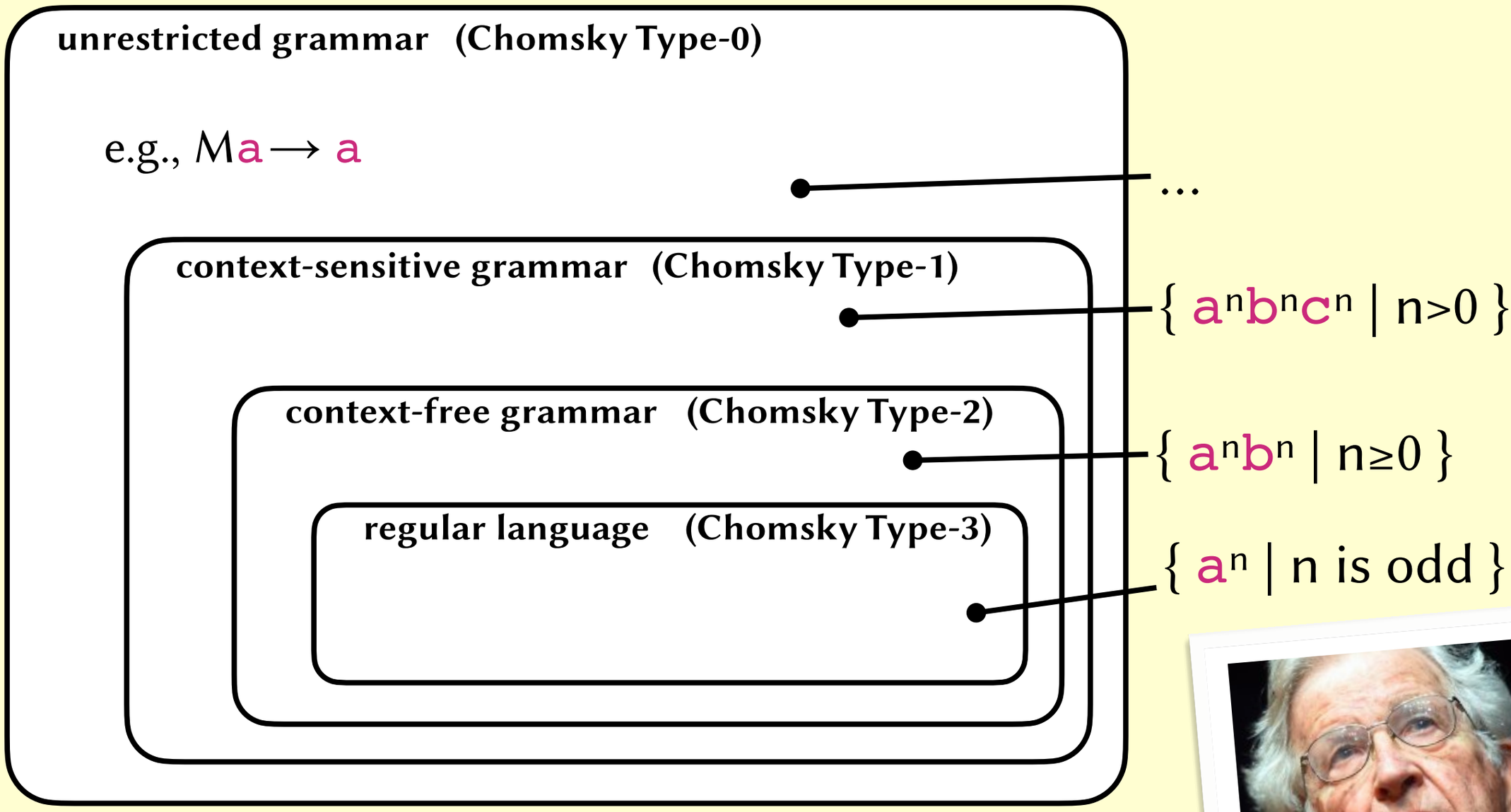
Context-sensitive grammars

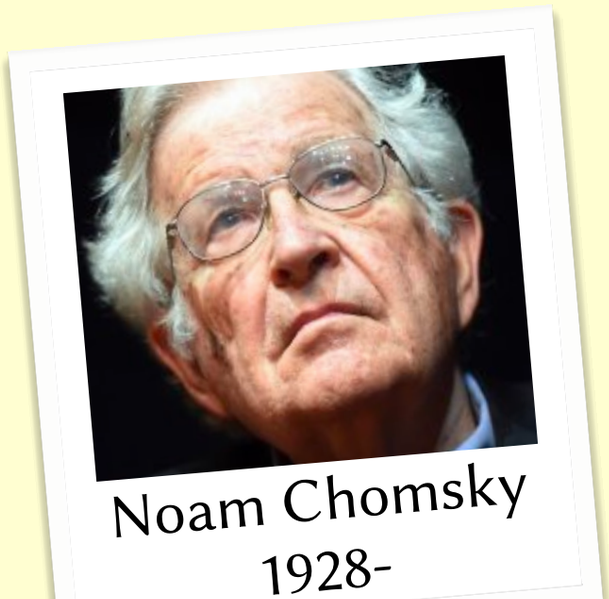
$\{ a^n b^n c^n \mid n>0 \}$

$$
\begin{aligned}
L &\longrightarrow aMLc \\
L &\longrightarrow aMc \\
Ma &\longrightarrow aM \\
Mc &\longrightarrow bc \\
Mb &\longrightarrow bb
\end{aligned}
$$

L

Hierarchy of languages





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.